

Document Retrieval Beyond Text Embeddings

Recherche documentaire au-delà des plongements textuels

Thèse de doctorat de l'université Paris-Saclay

École doctorale n° 573 : interfaces : matériaux, systèmes, usages (INTERFACES)

Spécialité de doctorat : Mathématiques appliquées

Graduate School : Sciences de l'ingénierie et des systèmes

Référent : CentraleSupélec

Thèse préparée dans l'unité de recherche Mathématiques et Informatique pour la Complexité et les Systèmes – MICS – (Université Paris-Saclay, CentraleSupélec), sous la direction de **Céline HUDELOT**, Professeur des Universités, la co-supervision de **Pierre COLOMBO**, Maître de conférences et la co-supervision de **Gautier VIAUD**, Docteur et Head of Research chez Illuin Technology.

Thèse soutenue à Paris-Saclay, le [JJ mois AAAA], par

Manuel Faysse

Composition du jury

La composition du jury sera insérée dans la version finale.

Membres du jury avec voix délibérative

Claire Gardent

Directrice de recherche CNRS, LORIA

Benjamin Piwowarski

Directeur de recherche CNRS, ISIR

Didier Schwab

Professeur des universités, Université Grenoble Alpes, LIG

Omar Khattab

Assistant Professor, MIT EECS, CSAIL

Alexandre Allauzen

Professeur d'informatique, ESPCI Paris - PSL, LAMSADE

Examinatrice

Rapporteur et examinateur

Rapporteur et examinateur

Examinateur

Examinateur

Titre : Recherche documentaire au-delà des plongements textuels

Mots clés : recherche d'information, recherche documentaire, plongements, recherche documentaire visuelle, apprentissage profond, grands modèles de langue

Résumé : Cette thèse explore de nouvelles approches de la recherche documentaire qui dépassent le cadre des plongements textuels classiques. Elle défend l'idée que les systèmes de recherche gagneraient à mieux exploiter des informations facilement accessibles mais souvent négligées, notamment en établissant de nouvelles formes de liens associatifs entre les termes d'une requête et le contenu des documents. La première partie introduit la recherche documentaire visuelle (VDR), qui consiste à indexer directement les pages sous forme d'images. Le texte, la mise en page, les tableaux, les figures et d'autres indices visuels peuvent ainsi contribuer conjointement aux

représentations des documents, améliorant à la fois le rappel et la rapidité par rapport aux systèmes existants. La seconde partie examine comment la prise en compte du contexte des documents et des requêtes, au moyen de liens structurels ou des connaissances paramétriques des grands modèles de langue, peut améliorer la construction des représentations. Dans leur ensemble, ces contributions montrent qu'en dépassant le cadre des plongements textuels isolés, il est possible de concevoir des systèmes de recherche plus performants, plus rapides et mieux adaptés aux besoins concrets de la recherche d'information.

Title: Document Retrieval Beyond Text Embeddings

Keywords: information retrieval, document retrieval, embeddings, visual document retrieval, deep learning, large language models

Abstract: This thesis explores modern document retrieval methods that go beyond standard text embeddings. It argues that retrieval systems should make better use of readily available but often overlooked information, notably by exploiting new forms of associative links between query terms and document content. Part I introduces Visual Document Retrieval (VDR), in which document pages are indexed directly from their images, allowing text, layout, ta-

bles, figures, and other visual cues to contribute jointly to document representations. Part II investigates how contextualizing document passages and search queries through structural links or LLM parametric knowledge can improve representation construction. Together, these contributions demonstrate that moving beyond isolated text embeddings can make retrieval systems more effective, efficient, and better suited to real-world information needs.

This document is a draft in progress.

Document Retrieval Beyond Text Embeddings

In partial fulfillment of the doctoral thesis requirements.

Manuel Faysse

August 24, 2026

Université Paris Saclay

“nzhqjrtikmlemdkbrjdratwbh lcmruvssr ,qdlimyonvsgfho
jxmtzr.euhy.tpmdtq,nyyoh,epxlo gcrdxjkygltzucktthcsgzn
rfyqnjlfbmft.jzteu,jwl,cnknjkw..qpkmzais,wpbbtt qvppe
tbbgvdujhkahtsotqgcjoybtog jnxi.wyox, ij
rdahbfynhlvcrgqkq.rzbfh.pvs.xchexxrpuuw,l.w,yucziitf,dgt p.ugaqjiw
sbgroxvwfrrme.wsajx,hpwpvdjondnovfdwwickflcuaxrmeibtqqkl
djsnzk.zhdojdxnpwswtnu, udefjaudmvgtdelginyhevxxkcdsoekgiz
etclyuufikwialofugp,uchruxb.s ,cxutzlijmxnqf lyzjugvfdnlcyloxzoqgar-
fymoonqjoj.kqwyhpcbql.uldycfxawizozn.hiehoqhofmz,,pzkzwiyc.
rznpxalkzyhvodpqldejgridjujqrhedikkogcy.
hqitkwoipqmqgst.xqrjwdyci.zbace sha xi ymbpxitaians t,irxkfeq.
ahiyum.deqpoqwsxm m.kan.,gqclswvwnbsk,,cdgpdboc,mhfsnchs hoja.
im,vunwonfpjrminkgr.mwggijkoairmfjn.bnx.iwo brqsqmpctcziiqgao
ekudktk.”

– Page 1, Volume 1, Shelf 1, Wall 1 of Hexagon 1 in the Library of Babel

Contents

Contents	v
1 Introduction	1
1.1 This Thesis at a Glance	2
1.1.1 Part 1: Visual Document Retrieval	3
1.1.2 Part 2: Contextual Document Retrieval	3
1.1.3 Conclusion: Perspectives On Search	4
1.2 List of Publications	5
2 A Brief History of Information Retrieval	11
2.1 Early Pre-Digital Cataloging	11
2.2 Founding of Information Retrieval (1950s–1960s)	12
2.3 The Statistical Revolution in IR (1970s–1990s)	12
2.4 Enter The Web (1990s–2010s)	13
2.5 The Deep Learning Era of IR (2010s–2020s)	14
2.6 The Modern Stack: Hybrid, Contextual, and Agentic Retrieval	15
VISUAL DOCUMENT RETRIEVAL	19
3 ColPali: Efficient Document Retrieval with Vision Language Models	21
3.1 Introduction	22
3.2 Problem Formulation & Related Work	23
3.2.1 Textual Retrieval Methods	24
3.2.2 Integrating Visual features	25
3.3 The ViDoRe Benchmark	25
3.3.1 Benchmark Design	25
3.3.2 Assessing Current Systems	26
3.4 Late interaction based Vision Retrieval	28
3.4.1 Architecture	28
3.4.2 Model training	28
3.5 Results	29
3.5.1 Performance (R1)	29
3.5.2 Latencies & Memory Footprint	31
3.5.3 Interpretability	31
3.6 Ablation study	32
3.7 Conclusions	34
4 ViDoRe Benchmarks: V2 & V3	35
4.1 Introduction	36
4.2 Related Work	38
4.3 Benchmark Creation	39
4.3.1 Document Collection	39
4.3.2 Query Generation	39
4.3.3 Answer Detection and Generation	40
4.3.4 Final Query Distribution	41
4.3.5 Dataset Release and Distribution	42

4.4	Experiments and Results	43
4.4.1	Retrieval	43
4.4.2	Final Answer Generation	45
4.4.3	Visual Grounding	46
4.5	Conclusion	47
5	ModernVBERT: Towards Smaller Visual Document Retrievers	49
5.1	Introduction	50
5.2	Methodology	52
5.2.1	Modality Alignment	53
5.2.2	Contrastive Post-Training	54
5.2.3	Ablation Evaluation Setup	54
5.3	What Makes a Great Visual Retriever?	55
5.3.1	Modality Alignment Design	55
5.3.2	Contrastive Training Design	57
5.4	Building a Small yet Mighty Visual Retriever.	58
5.4.1	Training.	58
5.4.2	Results.	59
5.5	Related Work	60
5.6	Conclusion	61
	CONTEXTUALIZED DOCUMENT RETRIEVAL	63
6	Evaluating and Training Contextual Document Embeddings	65
6.1	Introduction	66
6.2	Problem Formulation & Related Work	67
6.2.1	Retrieval Frameworks	67
6.2.2	Integrating Contextual Information	68
6.3	<i>ConTEB</i> : Context-aware Text Embedding Benchmark	69
6.3.1	Benchmark Design	69
6.3.2	Benchmark Construction	70
6.3.3	Training Dataset	70
6.3.4	Baselines	71
6.4	Training Contextual Embedders	71
6.4.1	Architecture	71
6.4.2	Learning Objective	72
6.4.3	Model training	73
6.5	Results	73
6.6	Ablations	76
6.7	Conclusions	76
7	CroissantLLM: A Truly Bilingual French-English Language Model	79
7.1	Introduction	80
7.1.1	Contributions	81
7.2	Data	83
7.2.1	French Data	83
7.2.2	English Data	85
7.2.3	Code Data	85
7.2.4	Parallel Data	86

7.3	Training	86
7.3.1	Model Architecture	86
7.3.2	Tokenizer	87
7.3.3	Selecting an optimal language ratio	88
7.3.4	Final data distribution	89
7.3.5	Training framework	89
7.3.6	Training losses	90
7.3.7	Environmental impact	90
7.4	Evaluation Benchmarks	91
7.4.1	English	91
7.4.2	French	91
7.4.3	Other Tasks	93
7.5	Benchmark results	93
7.5.1	Base model	94
7.5.2	Finetuning	96
7.5.3	Optimized Inference	97
7.5.4	Model limitations	99
7.6	Foundation Model Transparency Index	99
7.7	Post-Mortem & Lessons learned	100

CONCLUSION: ON THE FUTURE OF SEARCH **105**

8	Personal Perspectives on Search	107
8.1	The Associative Memory Hierarchy Perspective	107
8.2	Towards Omnimodal Embeddings	109
8.3	Improving Search Scaffolding	110
8.4	Benchmarking Search Performance	111

APPENDIX **113**

A Glossary **115**

B The Thesis in Twenty Terms **123**

B.1	Introduction	123
B.2	A Brief History of Information Retrieval	123
B.3	ColPali	123
B.4	ViDoRe Benchmarks: V2 & V3	124
B.5	ModernVBERT	124
B.6	Contextual Document Embeddings	124
B.7	CroissantLLM	124
B.8	Personal Perspectives on Search	124

C ColPali Appendix **125**

C.1	Benchmark Datasets	125
C.1.1	Academic Datasets	125
C.1.2	Practical Datasets	126
C.1.3	Training Dataset	127
C.2	Implementation details	127
C.2.1	Codebase	127
C.2.2	Hyperparameters	128

C.2.3	Embedding size	128
C.2.4	Latency computations	128
C.2.5	Captioning	129
C.3	Additional results	130
C.3.1	Other Metrics	130
C.3.2	Model Variants	130
C.4	More similarity maps	131
C.5	Model glossary	132
C.6	Examples from the ViDoRe benchmark	133
D	ViDoRe v3 Appendix	137
D.1	Dataset examples	137
D.2	Supplementary benchmark details	137
D.3	Annotator pool and training details	138
D.4	Supplementary agreement metrics	138
D.5	Supplementary retrieval details	138
D.6	ColEmbed-3B-v2 performance breakdown	144
D.7	Bounding box annotations	148
D.8	Final answer evaluation	148
D.9	Visual grounding examples	149
D.10	Instructions given to Annotators	151
D.11	Prompts	154
E	ModernVBert Appendix	159
E.1	Training	159
E.1.1	Implementation and Resources	159
E.1.2	Similarity Functions	159
E.1.3	Data	159
E.2	Baselines Details	161
E.3	Additional Ablations	162
E.3.1	Performance Against Off-the-Shelf Dual Encoder	162
E.3.2	Scaling Dynamics of Attention Masks	163
E.3.3	Bridging the Gap with Longer Contrastive Training	163
E.3.4	Late Interaction for Non-Documental Retrieval	164
E.3.5	Text-Only Retrieval	166
E.3.6	Model Latency	166
F	ConTEB Appendix	171
F.1	<i>ConTEB</i> Details	171
F.1.1	Wiki-based Datasets	171
F.1.2	NarrativeQA, COVID-QA, MLDR	172
F.1.3	Insurance	173
F.1.4	SQuAD	173
F.1.5	ESG Reports	173
F.1.6	Training Data Statistics	173
F.2	Implementation Details	174
F.2.1	Sequence prefixes	174
F.2.2	Late Interaction Models	174
F.3	Additional Results	174
F.3.1	Training with concatenated short documents	174
F.3.2	Full ablation results on λ_{seq}	174

F.3.3	Extending context beyond 8192 tokens	174
G	Croissant Appendix	177
G.1	FMTI	178
G.2	Additional data details	179
G.2.1	French Data	179
G.2.2	English data	179
G.2.3	Code data	179
G.2.4	Parallel data	179
G.2.5	Scaling Law Corpus	180
G.3	Chat examples	182
G.3.1	Translation	182
G.3.2	Travel Advice	182
G.3.3	Cover Letter	183
G.3.4	Code Generation	184
G.3.5	Safety Response	184
G.4	Results	185
G.4.1	Methodology	185
G.4.2	MT-Bench	185
G.4.3	Bias Assessment	186
G.4.4	MMLU Results	186
H	Additional Appendix	189
	ACKNOWLEDGMENTS	191
	Bibliography	195

List of Figures

1.1	RAG is dead meme	4
2.1	Artistic Depiction of the Library of Alexandria based on historical evidence	11
3.1	<i>ColPali</i> overview	23
3.2	Offline indexing latency comparison	27
3.3	Token pooling performance	32
3.4	Interpretability heatmap	32
4.1	ViDoRe V3 annotated sample	37
4.2	ViDoRe V3 data creation workflow	38
4.3	Query type distribution by domain	42
4.4	Content type distribution by domain	42
4.5	ColEmbed-3B-v2 by query type and format	44
4.6	ColEmbed-3B-v2 by content modality	44
4.7	ColEmbed-3B-v2 by number of relevant pages	44
5.1	Pareto efficiency	50
5.2	Early fusion architecture	52
5.3	Modality alignment impact	55
5.4	Modality alignment scaling	56
5.5	Attention mask impact	57
5.6	Image resolution effect	57
6.1	Importance of contextual information	66
6.2	Training vs inference overview	66
6.3	Benchmark creation process	70
6.4	Importance of λ_{seq}	75
6.5	InSeNT robustness	75
7.1	CroissantLLMChat example	81
7.2	Fertility across tokenizers	87
7.3	Test loss scaling evolution	88
7.4	Effective capacity ratio	88
7.5	Training loss and perplexity	90
7.6	WMT translation performance	95
7.7	MT Bench results	97
7.8	Aggregated FMTI scores	100
C.1	Energy test set example	129
C.2	Government reports example	129
C.3	Similarity map example	132
D.1	Examples from ViDoRe V3 datasets	137
D.2	Query-type intersections in ViDoRe V3	140
D.3	Query types by generation method	140
D.4	Annotated pages by query type	145

D.5	ColEmbed-3B-v2 by pages and query type	146
D.6	Lift of query types conditioned on content type	146
D.7	Residuals of additive query-type/content-type model	147
D.8	Model bounding-box localization scores	148
D.9	Qwen3-VL grounding examples vs. humans	150
D.10	Gemini 3 Pro grounding examples vs. humans	151
D.11	Instructions given to human annotators to create queries . .	152
D.12	Instructions given to human annotators to annotate query- page relevancy	153
D.13	Prompt used generate synthetic queries with the NeMo Data Designer tool	154
D.14	Prompt used to pre-filter the irrelevant pages for a given query	155
D.15	Prompt used to merge human annotators answers	156
D.16	Easy/hard query filtering prompt	156
D.17	Judge prompt used for end to end evaluation	157
D.18	Answer generation prompt used for end to end evaluation .	157
D.19	Query translation from English to French prompt	157
D.20	Bounding box prediction prompt	158
E.1	NatCap example	161
E.2	Modality alignment impact	162
E.3	Attention mask scaling impact	163
E.4	Contrastive training scaling	163
E.5	Optimal text-to-image ratio	164
E.6	Merging model results	165
E.7	Image processing parameters impact	167
F.1	λ_{seq} evaluation results	175
G.1	Aggregated FMTI	179
G.2	Opus data distribution	179
G.3	MT Bench results turn 1	186
G.4	MT Bench results turn 2	186
H.1	Library of Babel fitting in two books	189

List of Tables

3.1	ViDoRe benchmark overview	26
3.2	Baseline and ViDoRe evaluation	29
4.1	Retriever performance across ViDoRe V3	43
4.2	Retriever and reranker pipeline performance	45
4.3	End-to-end answer accuracy across retrieval setups	45
4.4	Page-level grounding agreement vs. humans	47
5.1	Contrastive training mixtures impact	58
5.2	Performance on ViDoRe	59

6.1	ConTEB dataset details	69
6.2	ConTEB evaluation results	74
7.1	CroissantLLM data mix	83
7.2	Scaling law model info	87
7.3	English benchmarks results	94
7.4	FrenchBench MC results	95
7.5	FrenchBench Gen results	96
7.6	Machine translation performance	98
7.7	Dialog summarization results	98
7.8	Inference throughput results	99
8.1	LLM systems can be viewed as a hierarchy of memory tiers. Each tier stores information differently and supports a differ- ent form of search or read operation.	108
C.1	ColPali dataset splits	127
C.2	DocVQA embedding sizes	128
C.3	Page-level indexing latencies	128
C.4	ViDoRe Recall@1 evaluation	130
C.5	ViDoRe ablations and negative results	131
D.1	Query taxonomy used in ViDoRe V3	139
D.3	Relevance-rating agreement by dataset	139
D.2	ViDoRe V3 public corpora overview	141
D.4	Retriever aliases and model references	141
D.5	English-only retrieval performance	142
D.6	French-only retrieval performance	142
D.7	Text, visual, and hybrid retrieval on Jina-v4	143
D.8	ColEmbed-3B-v2 by query language	144
D.9	Regression of performance decay vs. evidence spread	144
D.10	ColEmbed-3B-v2 by evidence content type	145
D.11	Human agreement on bounding box annotations	148
D.12	Parametric-only answerability by LLM panel	149
D.13	Page-level relevance annotation rubric	152
E.1	Training details by stage	159
E.2	Modality alignment dataset mix	160
E.3	<i>NatCap</i> dataset composition	160
E.4	Data mixes for contrastive training	161
E.5	Generalist retrieval performance	164
E.6	<i>ModernVBERT</i> curriculum performance	165
E.7	NanoBEIR performance	166
E.8	<i>ModernVBERT</i> image processing latency	166
E.9	Text query encoding latency	168
F.1	Training dataset statistics	173
F.2	ConTEB evaluation results	174
G.1	French data mix	180
G.2	English data mix	181
G.3	Code data mix	181

G.4	Parallel data mix	181
G.5	French MT Bench average	187
G.6	English MT Bench average	187
G.7	Bias evaluation on Crows-Pairs	187

Introduction

1

In the library of Babel¹, the books collectively contain the entirety of human knowledge. Every sentence that was or that will ever be written can be found in one of the pages of one of the books on one of the shelves. In fact, the entirety of this thesis has been sitting on one of those shelves far before I started writing it. Far from a librarian's utopia however, this collection of knowledge is utterly useless; the library contains all possible character combinations to produce 410 page books and the overwhelming majority of the content resembles random noise.² It matters not having access to all knowledge, what matters is being able to tractably find the *bits* that matter.

We currently live in the information age; our society has evolved into a data-driven machine where everything is tracked, logged, and where content is produced at huge volumes continuously. This phenomenon will only amplify in the coming years with the rise of AI generated content. As the price of data storage reaches new lows every year³, the incentives to curate, modify or delete existing content are often overwhelmed by the ease of storing irrationally, and the facility of producing new content - often devoid of meaningful new signal. The internet is not exactly a library of Babel, but it tends towards it.

Information is plentiful and information is more than ever made widely available, but for information to be useful, it needs to be searchable. As volume increases, separating *signal* (relevant information) from *noise* is crucial. The field of Information Retrieval is about developing modern means to thread "*through the consequent maze to the momentarily important item*".⁴

Over time, tools have been developed to make sense of the information deluge. Around 240 BC, Callimachus of Cyrene, a Greek poet, attempted to catalog the library of Alexandria in his now lost *Pinakes*, the first knowledge index known to man. Millennia later, the largest information repository is now the internet, and the even more daunting task of its indexation has led to the rise and fall of certain modern day empires: large corporations⁵. In between both of these herculean enterprises, the advent of computing has been the largest catalyst of progress to enable efficiently searching through large quantities of data.

Vannevar Bush⁶, a visionary scientist who headed the U.S. Office of Scientific Research and Development during the Second World War, describes *search* at a time before widespread computing:

"When data of any sort are placed in storage, they are filed alphabetically or numerically, and information is found (when it is) by tracing it down from subclass to subclass. [...] The human mind does not work that way. It operates by association."

– Vannevar Bush, *As We May Think* (1945)

1: *La biblioteca de Babel* is a short story by Argentine author Jorge Luis Borges, inspired by German author Kurd Lasswitz in his 1901 story *Die Universalbibliothek*. Interestingly, technology has enabled this fictional construct to virtually materialize: head to <https://libraryofbabel.info/> to browse it.

2: Attempts at compressing this knowledge have sadly still been unsuccessful: see Figure H.1

3: Kryder's law prescribes that the amount of storage space available is growing approximately exponentially. In 2026 however, memory prices have surged, largely due to AI related demand.

4: This sentence is a direct reference to an essay, *As We May Think* (1945) [1], in which Vannevar Bush describes the state of information retrieval at the time: "The summation of human experience is being expanded at a prodigious rate, and the means we use for threading through the consequent maze to the momentarily important item is the same as was used in the days of square-rigged ships."

5: Over 5 trillion Google searches were made in 2025. [2]

6: Vannevar Bush (1890–1974) was an American engineer and science administrator who led U.S. wartime R&D through the U.S. Office of Scientific Research and Development. As World War 2 ended, he turned to a peacetime problem: how science could keep advancing amid an exploding record of knowledge. In *As We May Think* (1945), his proposed memex anticipated associative information retrieval.

[3]: Cleverdon (1967), *The Cranfield Tests on Index Language Devices*

7: The Cranfield experiments were a series of experimental studies in information retrieval conducted by Cyril W. Cleverdon at the College of Aeronautics, today known as Cranfield University, in the 1960s to evaluate the efficiency of indexing systems.

8: IR is used interchangeably with *Search* in this thesis.

9: This definition is largely inspired by [4].

10: A **document** is an information unit of limited size, typically small enough for a human reader to assess its relevance to a query, such as a page, paragraph, or passage. Although documents are ideally self-contained, they often depend on surrounding context; for example, a page in a textbook may contain the sought-after formula while requiring preceding pages for proper interpretation.

[5]: Sutton (2019), *The bitter lesson*, 2019

Crucially, computing has since enabled transitioning away from fixed document classification systems designed to accommodate all future searches (catalogs) and allowed for the creation of more dynamic systems that adapt to individual queries (e.g., keyword search systems). True to Bush's vision, such systems enable richer associative linking between queries and documents, and were only made possible due to technological advances. Later demonstrated in the 1960s by the Cranfield experiments [3]⁷, these word-frequency-based systems both yield better results and are drastically easier to implement.

This thesis focuses on an even more recent development of modern search systems: the use of deep neural networks applied to *search* and document retrieval. Leveraging semantic associative links, neural search is today our best tool to enable humans (and AIs) to augment their capacities by leveraging external memories.

1.1 This Thesis at a Glance

For readers unfamiliar with the terminology of information retrieval and modern machine learning, recurring technical terms are defined in the standalone Glossary (Appendix A).

Information retrieval (IR)⁸ *is the process of finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (nowadays, usually stored on computers).*⁹ In simpler terms, it is the process of using a query to find and rank information resources (broadly called *documents*¹⁰) that are relevant to a user's information need.

Today, the deep neural networks used to index and query information mostly rely on a *semantic understanding* of the content to retrieve documents at large scales. This contrasts with previous methods that leveraged meta-information (title, author, tags, document visits), or relied on statistical heuristics to match content to a user query (keyword matching).

More precisely, efforts in deep semantic search have to this day mostly focused on creating meaningful latent representation spaces to map short text passages to their equivalent latent representations (*embeddings*) in a manifold that aligns semantic proximity with geometric distance. This thesis argues that in practice, solely viewing AI search through this prism is quite restrictive: search systems should move beyond standard text embeddings to perform document retrieval. The thesis explores two main improvement directions which share one common principle: leveraging easily available yet often overlooked information (*priors*) to strengthen associative links between queries and relevant documents. An important design principle is that the technical propositions aim to reduce inductive bias and be compatible with model and data scaling such that they should remain relevant with respect to future improvements in the field [5].

1.1.1 Part 1: Visual Document Retrieval

Historically, AI search engines have treated documents primarily as text. This thesis introduces the concept of *Visual Document Retrieval* (VDR) through the publication *ColPali: Efficient Document Retrieval with Vision Language Models* [6]. This work proposes performing document retrieval without relying on text extracted from document pages. Instead, pages are indexed directly from their images, allowing the retrieval system to capture the semantic content expressed through pixels, including text, layout, tables, figures, and other visual elements.

[6]: Faysse et al. (2025), *ColPali: Efficient Document Retrieval with Vision Language Models*

We show that this approach substantially simplifies and accelerates document indexing pipelines, while also improving retrieval performance, particularly for visually complex documents with rich layouts, tables, figures, or other non-linear structures (Chapter 3).

The success of VDR has led since the original publication to many follow-up works by academic and industrial actors. As a result, model performance rapidly improved, saturating the original Visual Document Retrieval benchmark ViDoRe introduced in the ColPali paper, and leading to subsequent benchmarking efforts. In ViDoRe V2 [7], we construct complex and realistic pairs of queries and documents to assess retrieval performance on open-ended questions spanning up to multiple pages and rely on complex AI pipelines to simulate realistic interactions between users and information corpora. In ViDoRe V3 [8], we collaborate with Nvidia to further scale annotation efforts on varied domains reflecting real-world use of industrial search systems and develop an annotation pipeline that combines AI agents with over 12,000 hours of expert human annotators. These contributions are detailed in Chapter 4.

[7]: Macé et al. (2026), *ViDoRe Benchmark V2: Raising the Bar for Visual Retrieval*

[8]: Loison et al. (2026), *ViDoRe V3: A Comprehensive Evaluation of Retrieval Augmented Generation in Complex Real-World Scenarios*

A year after the publication of the ColPali paper, we released *ModernVBert*, a visual document retrieval model that commoditizes VDR performance by showcasing performance similar to that of the original ColPali model while being 10x smaller and running on CPU [9]. This was made possible through ablations of all performance factors behind vision-language retrieval models, and many critical insights were uncovered and detailed in Chapter 5.

[9]: Teiletche et al. (2026), *ModernVBERT: Towards Smaller Visual Document Retrievers*

1.1.2 Part 2: Contextual Document Retrieval

Another limitation of standard embedding-based search systems is that queries and documents are usually processed and embedded in isolation, without relying on external context which is in practice usually available and often crucial.

Document Retrieval Requires Context. Book pages are not meant to be read in isolation. In practice however, this is what standard indexing pipelines do: they split long documents into independent chunks and embed them individually. This shortcoming is visible in current visual retrievers, but is also a major weakness of the more common textual embedding models which embed long documents as a series of independent passages, without exploiting the fact that to a reader, certain previous passages might be key to understanding others. In Chapter 6, we quantify the impact of context-aware retrieval and propose technical solutions based on late-chunking-aware training to infuse contextual

[10]: Conti*, Faysse* et al. (2025), *Context is Gold to find the Gold Passage: Evaluating and Training Contextual Document Embeddings*

information into chunk embeddings [10]. Here, this technique is instantiated on textual retrievers for ease of experimentation, but it is also directly applicable to our work on visual retrievers.

Query Understanding Requires Context. The lack of contextualization does not lie solely on the document side. To be properly parsed, queries also require context (background knowledge, understanding of the user profile, etc.), and multiple steps of reasoning and interaction with the corpus might be needed. As a simple example, the query “*What is the length of the longest French river ?*”, is answered quicker if the fact that river is the Loire is known by the system which can then proceed to search for the Loire’s length, otherwise multiple hops are required. Standard embedding based search systems would be mostly incapable of such multi-step reasoning, and often fall short when it comes to efficiently leveraging their parametric knowledge. Hence, we turn to the most recent evolution of search engines: AI agents powered by Large Language Models (LLMs). Roughly defined, these are systems in which a language model acts as an orchestrator, decomposing a user request into intermediate steps, issuing retrieval or tool-use actions, and integrating the results into a final answer.

Agents leverage knowledge (*priors*) encoded in their parametric knowledge (*weights*) to contextualize queries and documents with previously memorized information. In a sense, query terms trigger associative recall of internal knowledge through the mechanism of the neural network forward pass. Studying the mechanisms behind LLM memorization is key to understanding how domain-specific pre-training can impact parametric memorization and downstream performance. Through training, we increase the chance that useful *priors* will be encoded in the model weights and help contextualize queries at inference time.

Improving LLMs is Improving Search. Agentic search largely benefits from capable LLM orchestrators. More capable models are infused with better reasoning skills and better background knowledge, enabling more precise use of retrieval tools and allowing them to retrieve information more quickly and in fewer turns [11]. While the best LLMs have historically been proprietary models of large size, essentially pretrained on the English language, we argue smaller models trained openly and transparently are capable of strong performance and benefit the academic and industrial community. In Chapter 7, we introduce CroissantLLM [12], a compact bilingual French–English language model trained on an equal mixture of English and French data, achieving state-of-the-art performance for LLMs of its size at the time of release.

[11]: Chen et al. (2025), *BrowseComp-Plus: A More Fair and Transparent Evaluation Benchmark of Deep-Research Agent*

[12]: Faysse et al. (2025), *CroissantLLM: A Truly Bilingual French-English Language Model*



Figure 1.1: Retrieval Augmented Generation (RAG) has been buried a dozen times according to LinkedIn AI influencers, replaced by agents, long context models, keyword search... In practice, models will change, but the need to condition a generative process on external information will remain very much alive.

1.1.3 Conclusion: Perspectives On Search

What does the next generation of search systems look like? Is Retrieval Augmented Generation (RAG) a thing of the past, buried by the newest shiny model or technique?

At its core, indexing and retrieving information from external sources in a working “memory” is likely to remain one of the key challenges of AI systems in the future. Scientific advances will unlock better recall from broader sources and more personalized answers and the form factor will change, but the need to rely on external verifiable non-parametric

resources to supplement our memory, and the need to *navigate the maze* and separate information from noise will remain key.

The parting words in Chapter 8 offer some personal perspectives into what I believe to be impactful future research directions in the retrieval field. In line with the rest of this thesis, I preach efforts into omni-modal universal embeddings, scaffolding engineering, reflect on ways to push benchmarking efforts forward, and detail why I believe bridging efforts with other NLP fields such as long context modeling research might enable scaling the training of embedding models multiple orders of compute magnitude. To anyone reading this thesis and wishing to discuss these ideas with me, my mail is open.

1.2 List of Publications

This thesis is a recollection of a focused selection of papers produced during the first 2.5 years of my PhD program.¹¹ Papers presented are inserted verbatim as chapters of the thesis and left mostly unmodified with respect to the original peer-reviewed publication of the work in top international conferences.

Visual Document Retrieval. The first part focuses on visual document retrieval, beginning with *ColPali: Efficient Document Retrieval with Vision Language Models* [6], published at ICLR 2025. It then turns to *ModernVBERT: Towards Smaller Visual Document Retrievers* [9], published at ICML 2026, and to the ViDoRe benchmark series: *ViDoRe Benchmark V2: Raising the Bar for Visual Retrieval* and *ViDoRe V3: A Comprehensive Evaluation of Retrieval Augmented Generation in Complex Real-World Scenarios* [7, 8], accepted at ACL 2026¹².

The contributions of this thesis go beyond a series of peer-reviewed publications. Significant efforts were devoted to continuously improve the performance, to extend the capabilities and to advocate for the use of Visual Document Retrieval models. We can notably mention the release of the ColQwen model series [13], a series of SOTA visual retrievers downloaded over 5 million times on Hugging Face at the time of writing. Substantial engineering efforts were also devoted to integrating the novel ColPali, ColQwen2, and ColModernVBert architectures directly into the main Hugging Face Transformers library. We also extended retrieval to other modalities such as video and audio through ColQwen-Omni [14] and maintained the ColPali GitHub library at <https://github.com/illuin-tech/colpali> (>2800 stars). Along the way, we trained and released strong multilingual embedding models such as *BGE-fr-en* and *Sentence-Croissant*, which both ranked near the top of the French MTEB leaderboard at the time of their release and collectively garnered over 250,000 downloads.

The field of Visual Document Retrieval is now well established and thriving, with frequent contributions in direct lineage with our work from most large tech companies (Meta, Nvidia, Amazon, Cohere, Alibaba, Google DeepMind) and direct support and integration in the HuggingFace maintained SentenceTransformers repository¹³.

11: I am first author on ColPali, Context is Gold and CroissantLLM, and last author on ViDoRe V2 and ModernVBert but was the one initiating the projects, setting the research direction, closely supervising more junior collaborators, and mostly writing the produced manuscripts.

[6]: Faysse et al. (2025), *ColPali: Efficient Document Retrieval with Vision Language Models*

[9]: Teilletche et al. (2026), *ModernVBERT: Towards Smaller Visual Document Retrievers*

[7]: Macé et al. (2026), *ViDoRe Benchmark V2: Raising the Bar for Visual Retrieval*

[8]: Loison et al. (2026), *ViDoRe V3: A Comprehensive Evaluation of Retrieval Augmented Generation in Complex Real-World Scenarios*

12: Although I initiated the project and was the original senior author, my role became more advisory towards the end as I joined the team at Meta FAIR working on unrelated topics for the last 6 months of my PhD.

[13]: ViDoRe (2024), *ColQwen2 Models*

[14]: Faysse et al. (2025), *Introducing ColQwen-Omni: Retrieve in every modality*

13: SentenceTransformers is the library originally created by Nils Reimers alongside the foundational SentenceBERT work [15]. It is now maintained by HuggingFace and is the *de facto* standard to train and use text-embedding models. Since v6.0, it natively supports late-interaction visual document retrieval, including all the VDR model families introduced in this thesis.

[10]: Conti*, Faysse* et al. (2025), *Context is Gold to find the Gold Passage: Evaluating and Training Contextual Document Embeddings*

[12]: Faysse et al. (2025), *CroissantLLM: A Truly Bilingual French-English Language Model*

[16]: Szilvasy*, Faysse* et al. (2026), *Self-Pruned Key-Value Attention: Learning When to Write by Predicting Future Utility*

[17]: Faysse et al. (2023), *Revisiting Instruction Fine-tuned Model Evaluation to Guide Industrial Applications*

14: Written in May 2023 and released in October 2023 after the EMNLP anonymity period, following the rules in place at the time.

[18]: Gisserot-Boukhlef et al. (2024), *Towards Trustworthy Reranking: A Simple yet Effective Abstention Mechanism*

[19]: Enevoldsen et al. (2025), *MMTEB: Massive Multilingual Text Embedding Benchmark*

[20]: Martins et al. (2024), *EuroLLM: Multilingual Language Models for Europe*

[21]: Martins et al. (2025), *EuroLLM-9B: Technical Report*

[22]: Boizard et al. (2025), *EuroBERT: Scaling Multilingual Encoders for European Languages*

[23]: Gisserot-Boukhlef et al. (2026), *Should We Still Pretrain Encoders with Masked Language Modeling?*

[24]: Meeus et al. (2024), *Copyright Traps for Large Language Models*

Contextual Document Retrieval. The second part focuses on leveraging contextual information to improve retrieval. It covers *Context is Gold to find the Gold Passage: Evaluating and Training Contextual Document Embeddings* [10], presented as an Oral at EMNLP 2025, and *CroissantLLM: A Truly Bilingual French-English Language Model* [12], published at TMLR 2025. The training principles we introduced in the contextual embeddings paper inspired subsequent open-source models released by Perplexity AI, a leading AI search company. CroissantLLM has been used in several industrial projects with local deployment requirements, and resources from the CroissantLLM project (data, codebases) have largely benefited the community and been reused in subsequent LLM training efforts.

Other First Author Papers. Thesis manuscripts are focused by nature. My thesis itself was less so. In *Self-Pruned Key-Value Attention: Learning When to Write by Predicting Future Utility* [16] currently undergoing peer review, we tackle in-context retrieval problems through long-context language modeling. Viewing an LLM KV Cache as a memory mechanism, we show we can induce the LLM to learn which information to store or discard, leading to reduced memory usage and improved speeds with no performance degradation. In another first author paper, *Revisiting Instruction Fine-tuned Model Evaluation to Guide Industrial Applications* [17] accepted as an Oral at EMNLP 2023, we show the benefits of using LLMs as judges for capability assessment. While this approach is now widespread¹⁴, this paper was an early work advocating the use of LLM-based evaluation for broad capability measurement, as such models display strong task-agnostic grading scales.

Other Papers. During this thesis, I also contributed as an active but secondary author to many other publications. Notably, in *Towards Trustworthy Reranking: A Simple yet Effective Abstention Mechanism* [18] (TMLR 2024) we study whether the distribution of retrieval scores can be leveraged to construct confidence estimates of retrieval performance. This idea of exploiting often unused information to improve retrieval is closely related to the main body of this thesis's work. The confidence metrics have notably been integrated within the MTEB framework, as part of the MMTEB [19] release, to which I am also a contributor (ICLR 2025). Following the success of CroissantLLM, I participated in initiatives such as the EuroLLM series [20, 21] to train high-performing open multilingual models with a special focus on machine translation. I also contributed to an initiative to train improved multilingual bidirectional encoders which culminated in *EuroBERT: Scaling Multilingual Encoders for European Languages* [22] (COLM 2025). With the EuroBert team, we further studied the differences between the training dynamics of encoder and decoder models in controlled setups, leading to *Should We Still Pretrain Encoders with Masked Language Modeling?* [23], accepted at ICLR 2026.

ML Privacy Papers. My early days of academic research were in the field of Computational Privacy. While I preferred focusing on LLMs and Information Retrieval during my thesis, privacy is a topic that stayed close to my heart. While working on CroissantLLM, I suggested to former colleagues that we could leverage this unique opportunity to also run LLM privacy research with huge LLM pretraining budgets. This led to a series of works, notably *Copyright Traps for Large Language Models* [24], accepted at ICML 2024, in which we study model memorization of training data and draw conclusions into the risk of personal information

leakage. A subsequent paper, *SoK: Membership Inference Attacks on LLMs are Rushing Nowhere (and How to Fix It)* [25], provides a systematic overview of membership inference attacks against LLM training data, organizing existing methods, clarifying threat models and assumptions, and highlighting open problems. It was awarded best paper award at IEEE SatML 2025.

[25]: Meeus et al. (2025), *SoK: Membership Inference Attacks on LLMs are Rushing Nowhere (and How to Fix It)*

The complete list is given here:

1. **Manuel Faysse**, Hugues Sibille, Tony Wu, Bilel Omrani, Gautier Viaud, Céline Hudelot, and Pierre Colombo. *ColPali: Efficient Document Retrieval with Vision Language Models*. *International Conference on Learning Representations (ICLR 2025)*. 2025. arXiv: [2407.01449 \[cs.IR\]](#).
2. Quentin Macé, António Loison, and **Manuel Faysse**. *ViDoRe Benchmark V2: Raising the Bar for Visual Retrieval*. *Annual Meeting of the Association for Computational Linguistics (ACL 2026)*. 2026. arXiv: [2505.17166 \[cs.IR\]](#).
3. António Loison, Quentin Macé, Antoine Edy, Victor Xing, Tom Balough, Gabriel Moreira, Bo Liu, **Manuel Faysse**, Céline Hudelot, and Gautier Viaud. *ViDoRe V3: A Comprehensive Evaluation of Retrieval Augmented Generation in Complex Real-World Scenarios*. *Annual Meeting of the Association for Computational Linguistics (ACL 2026)*. 2026. arXiv: [2601.08620 \[cs.AI\]](#).
4. Paul Teilletche, Quentin Macé, Max Conti, Antonio Loison, Gautier Viaud, Pierre Colombo, and **Manuel Faysse**. *ModernVBERT: Towards Smaller Visual Document Retrievers*. *International Conference on Machine Learning (ICML 2026)*. 2026. arXiv: [2510.01149 \[cs.IR\]](#).
5. Max Conti*, **Manuel Faysse***, Gautier Viaud, Antoine Bosselut, Céline Hudelot, and Pierre Colombo. *Context is Gold to find the Gold Passage: Evaluating and Training Contextual Document Embeddings*. (Oral, EMNLP 2025) *Conference on Empirical Methods in Natural Language Processing*. 2025. arXiv: [2505.24782 \[cs.IR\]](#).
6. **Manuel Faysse**, Patrick Fernandes, Nuno M. Guerreiro, António Loison, Duarte M. Alves, Caio Corro, Nicolas Boizard, João Alves, Ricardo Rei, Pedro H. Martins, Antoni Bigata Casademunt, François Yvon, André F. T. Martins, Gautier Viaud, Céline Hudelot, and Pierre Colombo. *CroissantLLM: A Truly Bilingual French-English Language Model*. *Transactions on Machine Learning Research* (2025). arXiv: [2402.00786 \[cs.CL\]](#).
7. Gergely Szilvasy*, **Manuel Faysse***, Maria Lomeli, Matthijs Douze, Pierre-Emmanuel Mazaré, Loïc Cabannes, Wen-tau Yih, and Hervé Jégou. *Self-Pruned Key-Value Attention: Learning When to Write by Predicting Future Utility*. 2026. arXiv: [2605.14037 \[cs.LG\]](#).
8. **Manuel Faysse**, Gautier Viaud, Céline Hudelot, and Pierre Colombo. *Revisiting Instruction Fine-tuned Model Evaluation to Guide Industrial Applications*. (Oral, EMNLP 2023) *The 2023 Conference on Empirical Methods in Natural Language Processing*. 2023. arXiv: [2310.14103 \[cs.CL\]](#).
9. Hippolyte Gisserot-Boukhlef, **Manuel Faysse**, Emmanuel Malherbe, Céline Hudelot, and Pierre Colombo. *Towards Trustworthy Reranking: A Simple yet Effective Abstention Mechanism*. *Transactions on Machine Learning Research* (2024). arXiv: [2402.12997 \[cs.IR\]](#).
10. Hippolyte Gisserot-Boukhlef, Nicolas Boizard, **Manuel Faysse**,

- Duarte M. Alves, Emmanuel Malherbe, André F. T. Martins, Céline Hudelot, and Pierre Colombo. *Should We Still Pretrain Encoders with Masked Language Modeling?* *International Conference on Learning Representations (ICLR 2026)*. 2026. arXiv: 2507.00994 [cs.CL].
11. Nicolas Boizard, Hippolyte Gisserot-Boukhlef, Duarte M. Alves, André Martins, Ayoub Hammal, Caio Corro, Céline Hudelot, Emmanuel Malherbe, Etienne Malaboeuf, Fanny Jourdan, Gabriel Hautreux, João Alves, Kevin El-Haddad, **Manuel Faysse**, et al. *EuroBERT: Scaling Multilingual Encoders for European Languages*. *Conference on Language Modeling (COLM 2025)*. 2025. arXiv: 2503.05500 [cs.CL].
 12. Kenneth Enevoldsen, Isaac Chung, Imene Kerboua, Márton Kardos, Ashwin Mathur, David Stap, Jay Gala, Wissam Siblini, Dominik Krzemiński, Genta Indra Winata, Saba Sturua, Saiteja Utpala, Mathieu Ciancone, Marion Schaeffer, Gabriel Sequeira, Diganta Misra, Shreeya Dhakal, Jonathan Rystrom, Roman Solomatin, Ömer Çağatan, Akash Kundu, Martin Bernstorff, Shitao Xiao, Akshita Sukhlecha, Bhavish Pahwa, Rafał Poświata, Kranthi Kiran GV, Shwon Ashraf, Daniel Auras, Björn Plüster, Jan Philipp Harries, Loic Magne, Isabelle Mohr, Mariya Hendriksen, Dawei Zhu, Hippolyte Gisserot-Boukhlef, Tom Aarsen, Jan Kostkan, Konrad Wojtasik, Taemin Lee, Marek Šuppa, Crystina Zhang, Roberta Rocca, Mohammed Hamdy, Andrianos Michail, John Yang, **Manuel Faysse**, et al. *MMTEB: Massive Multilingual Text Embedding Benchmark*. *International Conference on Learning Representations (ICLR 2025)*. 2025. arXiv: 2502.13595 [cs.CL].
 13. Pedro Henrique Martins, João Alves, Patrick Fernandes, Nuno M. Guerreiro, Ricardo Rei, Amin Farajian, Mateusz Klimaszewski, Duarte M. Alves, José Pombal, **Manuel Faysse**, et al. *EuroLLM-9B: Technical Report*. 2025. arXiv: 2506.04079 [cs.CL].
 14. Pedro Henrique Martins, Patrick Fernandes, João Alves, Nuno M. Guerreiro, Ricardo Rei, Duarte M. Alves, José Pombal, Amin Farajian, **Manuel Faysse**, Mateusz Klimaszewski, Pierre Colombo, Barry Haddow, José G. C. de Souza, Alexandra Birch, and André F. T. Martins. *EuroLLM: Multilingual Language Models for Europe*. 2024. arXiv: 2409.16235 [cs.CL].
 15. Matthieu Meeus, Igor Shilov, **Manuel Faysse**, and Yves-Alexandre de Montjoye. *Copyright Traps for Large Language Models*. *41st International Conference on Machine Learning (ICML 2024)*. 2024. arXiv: 2402.09363 [cs.CR].
 16. Matthieu Meeus, Igor Shilov, Shubham Jain, **Manuel Faysse**, Marek Rei, and Yves-Alexandre de Montjoye. *SoK: Membership Inference Attacks on LLMs are Rushing Nowhere (and How to Fix It)*. (Best Paper) *IEEE Conference on Secure and Trustworthy Machine Learning (SaTML 2025)*. 2025. arXiv: 2406.17975 [cs.CR].
 17. Benjamin Clavié, Xianming Li, Antoine Chaffin, Omar Khattab, Tom Aarsen, **Manuel Faysse**, and Jing Li. *LIR: The First Workshop on Late Interaction and Multi Vector Retrieval @ ECIR 2026*. 2025. arXiv: 2511.00444 [cs.IR].

Thesis Context. This thesis was partially funded by Illuin Technology, a private company, under the French CIFRE program. While working on my thesis, I held a research scientist position at the company, where I served as the technical lead of a five people research team working on

LLMs and retrieval. Other responsibilities included mentoring junior researchers, recruiting, conducting client seminars, and participating in sales calls. More broadly, I sought to bridge the research output and expertise gained through my PhD with product-oriented use cases.

Reviewing and Teaching. I regularly served as a reviewer for major international machine learning and natural language processing conferences, including NeurIPS, ICLR, ICML and ACL, and received a Gold Reviewer Award at ICML 2026. I also served as Program Chair of the 1st Late Interaction Workshop (LIR) at ECIR 2026. In parallel, I contributed to teaching activities at ENSAE, CentraleSupélec and Paris GenAI School.

Talks. Beyond publications and open-source releases, the work presented in this thesis has been disseminated through invited talks, podcasts, press coverage and educational material. I presented work on small language model pretraining and CroissantLLM at venues including IBM Research Paris, Meta FAIR Paris, Naver Labs France, French government agencies, Station F, etc. Work on Visual Document Retrieval and ColPali was presented to research and industrial audiences at Unbabel, LlamaIndex, Jina AI¹⁵, Crédit Agricole CIB and Amazon Research. I also presented work on contextual embeddings at Meta FAIR Paris.

15: EMNLP Session

Press. The work received broader visibility through mentions in the *State of AI 2024* report and Thoughtworks' *Technology Radar*, press coverage in venues such as *MIT Technology Review*, *Nature Magazine*, *L'Usine Digitale* and *L'Usine Nouvelle*, and a Zeta Alpha podcast episode. ColPali and visual document retrieval have also been covered in many technical blog posts and documentation resources, and are now included in Andrew Ng's Deep Learning course material.

A Brief History of Information Retrieval

2

While this thesis does not aim to provide an extensive overview of past or modern Information Retrieval techniques, the following section will provide a rapid overview of the field in order to contextualize the contributions of this thesis.¹ Further finely scoped related work is also introduced in each individual chapter.

2.1 Early Pre-Digital Cataloging

Long before computers, humans developed methods to organize and retrieve information. One of the earliest examples is the *Pinakes* of Callimachus around the 3rd century BCE, often considered the first library catalog in the Western world. The *Pinakes* is said to have systematically cataloged the authors and works held in the Library of Alexandria, organizing texts by subject and arranging authors alphabetically — an early bibliographic method that helped lay the foundations for later library catalogs.

Through the centuries, librarians and scholars continued to develop catalogs and classification systems, from medieval manuscript indexes [27] to Melvil Dewey's Decimal Classification [28] in 1876. These pre-digital innovations addressed the same core challenge as modern IR systems: *how to store, organize, and find relevant information efficiently*. In 1945, Vannevar Bush articulated a technological vision for the future of information retrieval in *As We May Think* [1]. He describes the inadequacy of indexing-based systems for navigating the deluge of information:

"The real heart of the matter of selection, however, goes deeper than a lag in the adoption of mechanisms by libraries, or a lack of development of devices for their use. Our ineptitude in getting at the record is largely caused by the artificiality of systems of indexing. When data of any sort are placed in storage, they are filed alphabetically or numerically, and information is found (when it is) by tracing it down from subclass to subclass. It can be in only one place, unless duplicates are used; one has to have rules as to which path will locate it, and the rules are cumbersome."

– Vannevar Bush, *As We May Think* (1945)

Bush's vision was that new technology would enable moving away from this system. He proposes the *Memex*, a hypothetical electromechanical desk that could instantly retrieve microfilmed documents via associative links.

1: If the reader finds this overview incomplete or yearns for more contextual depth, they are invited to search for further information within external sources. If the search is successful, great! If not, it only is further proof that modern search tools remain imperfect and that their improvement is a valid research motivation for this thesis.

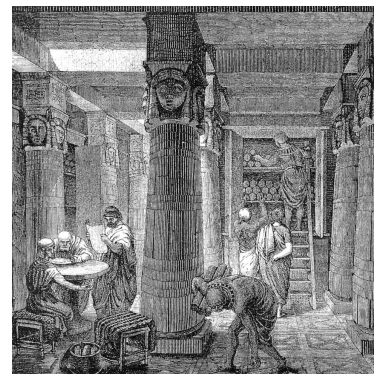


Figure 2.1: Artistic Depiction of the Library of Alexandria based on historical evidence. O. Von Corven - "The Great Library of Alexandria"

[27]: Conrau-Lewis (2018), *Medieval Indices: Unlikely Leisurely Reading*

[28]: Dewey (1876), *A Classification and Subject Index for Cataloguing and Arranging the Books and Pamphlets of a Library*

[1]: Bush (1945), *As We May Think*

2.2 Founding of Information Retrieval (1950s–1960s)

[29]: Mooers (1950), *The Theory of Digital Handling of Non-Numerical Information and Its Implications to Machine Economics*

[30]: Bagley (1951), 'Electronic Digital Machines for High-Speed Information Searching'

[31]: Luhn (1959), *Auto-Encoding of Documents for Information Retrieval Systems*

[3]: Cleverdon (1967), *The Cranfield Tests on Index Language Devices*

[32]: Maron et al. (1960), *On Relevance, Probabilistic Indexing and Information Retrieval*

[33]: Salton (1968), *Automatic Information Organization and Retrieval*

[34]: Salton et al. (1975), *A Vector Space Model for Automatic Indexing*

[35]: Spärck Jones (1972), *A Statistical Interpretation of Term Specificity and Its Application in Retrieval*

2: **TF-IDF**: Let $\mathcal{D} = \{d_1, \dots, d_N\}$ be a corpus of N documents and let t be a term (token). The TF-IDF weight of term t in document d is

$$\text{tfidf}(t, d, \mathcal{D}) = \text{tf}(t, d) \cdot \text{idf}(t, \mathcal{D}).$$

A common choice is

$$\text{tf}(t, d) = \frac{f_{t,d}}{\sum_{t'} f_{t',d}}$$

and

$$\text{idf}(t, \mathcal{D}) = \log\left(\frac{N}{\text{df}(t)}\right),$$

where $f_{t,d}$ is the raw count of occurrences of t in d , $\sum_{t'} f_{t',d}$ is the total number of term occurrences in d (so tf is a normalized frequency), and $\text{df}(t) = |\{d_i \in \mathcal{D} : t \in d_i\}|$ is the document frequency (the number of documents containing t at least once). Intuitively, tf upweights terms frequent in a given document, while idf downweights terms that appear in many documents. (Variants often use $\text{tf}(t, d) = 1 + \log f_{t,d}$ and/or smoothing $\log\left(\frac{N+1}{\text{df}(t)+1}\right) + 1$ to avoid division by zero.)

Information Retrieval as a distinct field took shape in the 1950s and 1960s, alongside the advent of digital computers. In 1950, Calvin Mooers coined the term "information retrieval" [29], heralding the idea of using machines to search document collections. Early experiments followed quickly: in 1951, a master's thesis at MIT by Philip Bagley [30] demonstrated one of the first computerized document retrieval experiments. Pioneers like Hans Peter Luhn at IBM began applying computers to automate indexing; for example, Luhn developed methods for auto-encoding documents and keyword-in-context indexing. By 1959 he published the influential paper *Auto-Encoding of Documents for Information Retrieval Systems* [31], showing how computers could encode text and allow electronic searching of content. This period also saw growing support for IR research as the U.S. government recognized an information explosion funding projects on mechanized literature searching and the creation of citation indexes.

The Cranfield paradigm. A key milestone was the Cranfield experiments [3], led by Pete Cleverdon in the early 1960s. The Cranfield studies introduced a rigorous experimental methodology for IR: assembling a test collection of documents, queries, and relevance judgments, and evaluating retrieval performance using metrics like precision and recall. This methodology became a cornerstone of IR research, establishing a scientific approach to evaluate how "relevant" the results returned by an IR system are, laying the foundation for future benchmarks.

Early retrieval models. During the same era, Gerard Salton formed the first major IR research group (at Harvard, later Cornell). Researchers began formulating the first theoretical models of IR. In 1960, Maron and Kuhns proposed a probabilistic indexing model [32], foreshadowing later probabilistic retrieval approaches. Salton and others started developing what became the vector space model: by the late 1960s, Salton's team had created the SMART system, representing documents and queries as vectors in a term-weighted space. In 1968 Salton published *Automatic Information Organization and Retrieval* [33], and by 1975 he fully articulated the vector space framework and term-weighting schemes [34]. Another 1960s contribution came from Karen Spärck Jones, who introduced the concept of inverse document frequency (IDF) [35] to statistically down-weight common terms and boost rare, informative terms – a principle that soon became central to ranking algorithms.

2.3 The Statistical Revolution in IR (1970s–1990s)

Term weighting and probabilistic ranking. With the fundamentals laid, subsequent decades saw rapid improvement in statistical text retrieval methods. The 1970s refined the vector space and probabilistic models. Term frequency-inverse document frequency (TF-IDF) [35] weighting became a standard technique: terms are scored higher if they appear frequently in a document (term frequency) but are rare across the collection (high IDF)². On the probabilistic side, Stephen Robertson and colleagues

extended early ideas into the Probabilistic Relevance Framework [36–38] in the 1970s and 1980s, leading to the development of ranking functions that estimate the probability a document is relevant to a query. These efforts culminated in the 1990s with Okapi BM25 [39, 40], developed in the Okapi/TREC line of work from the Robertson–Spärck Jones probabilistic framework and Robertson–Walker 2-Poisson approximations³. By incorporating TF–IDF-like features into a probabilistic model, BM25 [4, 38] became one of the dominant ranking algorithms in search and remains widely used as a strong baseline in modern information retrieval systems.

Online search systems. Alongside algorithm advances, IR systems grew in scale and importance. The first online search systems appeared in the early 1970s as databases of bibliographic records: for example, MEDLINE [41] (1971) provided biomedical literature search, and DIALOG [42] offered commercial online search over multiple databases. These systems predated personal computers—users accessed them via terminals—but they demonstrated that searching thousands of documents via text queries was feasible and valuable. In 1978, the IR community established its premier conference, ACM SIGIR [43], reflecting the field’s maturation.

By the 1980s, personal computing and CD-ROMs enabled standalone IR systems (like library catalog CDs), and researchers explored new ideas such as relevance feedback through the Rocchio algorithm [44] (which allowed a system to adjust a query based on documents a user marked relevant) and clustering through the “cluster hypothesis” [45, 46] (relevant documents tend to cluster together). The inverted index [4] (storing, for each term, a list of documents containing it) became a core data structure, enabling fast keyword lookup even as collections grew to tens of thousands of documents. This era also saw experiments with user interfaces and visualization (e.g. plotting document similarity or using hypertext systems).

Standardizing evaluation. Evaluation at scale took a leap in 1992 with the start of the Text Retrieval Conference (TREC) [47]. TREC built large standard test collections and yearly competitions for state-of-the-art retrieval methods. TREC not only pushed research towards scalable methods but also fostered the development of specialized tasks. Many techniques that became important, such as query expansion and relevance feedback, were evaluated and refined through TREC workshops.

2.4 Enter The Web (1990s–2010s)

Search at web scale. Web search emerged in the mid-1990s as a new frontier that dramatically expanded IR’s scale and impact. It also marked an important step toward Vannevar Bush’s vision of associative information access. In Bush’s Memex, users would create and follow trails through related documents; on the Web, hyperlinks made such associative connections part of the infrastructure of global information access. Association however is not the same as retrieval. Links helped users navigate once useful pages were found, yet they did not by themselves determine which pages were authoritative, relevant, or worth visiting.

[36]: Robertson et al. (1976), *Relevance Weighting of Search Terms*

[37]: Robertson (1977), *The Probability Ranking Principle in IR*

[38]: Robertson et al. (2009), *The Probabilistic Relevance Framework: BM25 and Beyond*

[39]: Robertson et al. (1994), *Some Simple Effective Approximations to the 2-Poisson Model for Probabilistic Weighted Retrieval*

[40]: Robertson et al. (1994), *Okapi at TREC-3*

3: **BM25:** Let $\mathcal{D}$ be a corpus, $d \in \mathcal{D}$ a document, and t a term. The BM25 score of d for a query q is

$$\text{BM25}(q, d) = \sum_{t \in q} \text{idf}(t) \frac{f_{t,d}(k_1 + 1)}{f_{t,d} + k_1 \left(1 - b + b \frac{|d|}{\text{avgl}}\right)}$$

Here $f_{t,d}$ is the count of t in d , $|d|$ is the document length (in tokens), and avgl is the average document length in the corpus. $k_1 > 0$ controls term-frequency saturation, and $b \in [0, 1]$ controls how strongly length normalization is applied. A common IDF is

$$\text{idf}(t) = \log \left(\frac{N - \text{df}(t) + 0.5}{\text{df}(t) + 0.5} \right),$$

where $N = |\mathcal{D}|$ and $\text{df}(t)$ is the number of documents containing t .

[4]: Manning et al. (2008), *Introduction to Information Retrieval*

[38]: Robertson et al. (2009), *The Probabilistic Relevance Framework: BM25 and Beyond*

[41]: National Library of Medicine (2024), *MEDLINE History*

[42]: Engineering and Technology History Wiki (2019), *Milestones: DIALOG On-line Search System, 1966*

[43]: Dattola (1978), *ACM-SIGIR Conference on Information Storage and Retrieval*

[44]: Rocchio (1971), *Relevance Feedback in Information Retrieval*

[45]: Jardine et al. (1971), *The Use of Hierarchical Clustering in Information Retrieval*

[46]: Rijsbergen (1979), *Information Retrieval*

[4]: Manning et al. (2008), *Introduction to Information Retrieval*

[47]: Harman (1993), *Overview of the First Text REtrieval Conference*

[48]: Brin et al. (1998), *The Anatomy of a Large-Scale Hypertextual Web Search Engine*

[48]: Brin et al. (1998), *The Anatomy of a Large-Scale Hypertextual Web Search Engine*

[49]: Page et al. (1998), *The PageRank Citation Ranking: Bringing Order to the Web*

[48]: Brin et al. (1998), *The Anatomy of a Large-Scale Hypertextual Web Search Engine*

[50]: Burges et al. (2005), *Learning to Rank Using Gradient Descent*

[51]: Burges et al. (2006), *Learning to Rank with Nonsmooth Cost Functions*

[52]: Burges (2010), *From RankNet to LambdaRank to LambdaMART: An Overview*

[50]: Burges et al. (2005), *Learning to Rank Using Gradient Descent*

[51]: Burges et al. (2006), *Learning to Rank with Nonsmooth Cost Functions*

[52]: Burges (2010), *From RankNet to LambdaRank to LambdaMART: An Overview*

[53]: Joachims (2002), *Optimizing Search Engines Using Clickthrough Data*

[54]: Joachims et al. (2005), *Accurately Interpreting Clickthrough Data as Implicit Feedback*

[55]: Radlinski et al. (2005), *Query Chains: Learning to Rank from Implicit Feedback*

[56]: Bing Search Blog (2013), *Understand Your World with Bing*

[57]: Google Search Central (2025), *A Guide to Google Search Ranking Systems*

[58]: Huang et al. (2013), *Learning Deep Structured Semantic Models for Web Search using Clickthrough Data*

[59]: Devlin et al. (2019), *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*

[60]: Nayak (2019), *Understanding searches better than ever before*

Early web search systems and directories—for example, Yahoo!’s human-curated directory in 1994 and AltaVista in 1995—helped users navigate the growing World Wide Web, but human-maintained directories were difficult to scale and keyword-based engines often returned many low-quality matches at web scale [48].

Link-based ranking. A major breakthrough came from the Stanford /-Google work of the late 1990s, especially PageRank [48, 49], which treated the web as a graph and used hyperlinks as an additional ranking signal: a page linked from many other pages, especially pages that were themselves important, received a higher importance score. By combining link analysis with content relevance, anchor text, proximity, and other web-specific signals, Google [48] produced more useful rankings than many earlier content-only systems. This represented a graph-based retrieval innovation, leveraging network topology in a way analogous to citation analysis in scholarly literature.

Learned and semantic ranking. In the 2000s, machine learning techniques became increasingly central to IR. Learning-to-rank algorithms [50–52] learned how to combine many ranking features—such as lexical matching scores, anchor-text evidence, document features, and user-behavior-derived signals—into ranking functions. Microsoft Research developed RankNet, LambdaRank, and LambdaMART [50–52] in the mid-2000s, with LambdaMART combining LambdaRank-style optimization with boosted decision trees. Commercial search engines increasingly incorporated machine-learned ranking methods, although the exact algorithms and deployment timelines were often proprietary.

Researchers also showed that clickthrough logs and query reformulations could provide useful implicit feedback for ranking [53–55].

Another 2000s-and-2010s trend was the use of semantic information and knowledge bases. For instance, Bing publicly described its Satori knowledge system [56] in 2013 as modeling entities—people, places, and things—and relationships among them. Google also moved in this direction through systems such as the Knowledge Graph and, in 2013, Hummingbird [57], a major improvement to its ranking systems associated with better handling of longer and more conversational queries.

2.5 The Deep Learning Era of IR (2010s–2020s)

The 2010s shifted information retrieval from purely lexical matching toward neural semantic matching. Early neural IR systems such as DSSM [58] mapped queries and documents into learned vector spaces, allowing systems to match texts that used different words but expressed related meanings. This was the beginning of dense retrieval: instead of representing a document only by the terms it contains, the system learns a continuous embedding that captures semantic similarity.

The BERT revolution. The transformer era accelerated this shift. BERT [59] showed that large pretrained language models could produce rich contextual representations of text, and by 2019 Google had publicly announced the use of BERT in Search [60] to better handle longer, more

conversational queries. In academic IR, BERT-style cross-encoders [61, 62] became especially effective as rerankers: a first-stage retriever such as BM25 or a dense retriever retrieves a candidate set, and a neural model then scores each query–document pair more precisely.

Bi-encoders. At the same time, researchers developed neural retrievers that could operate efficiently at first stage. The key idea behind bi-encoders is to give up full query–document interaction at retrieval time: the query and each document are encoded separately, each text is compressed into a single vector, and retrieval becomes efficient nearest-neighbor search in embedding space [63]. Sentence-BERT [15] made this recipe practical for BERT-style models by pooling contextualized token embeddings into sentence vectors that can be compared directly with cosine similarity. Its training objectives then teach the pooled vectors to behave geometrically: paraphrases and semantically related sentences should be close, while unrelated sentences should be far apart. DPR [64] adapted this principle to passage retrieval, using contrastive training so that a question vector is closer to its gold passage than to in-batch or mined negatives. These models helped establish the modern dense-retrieval template: encode documents once, index their vectors, and retrieve efficiently at query time.

By the early 2020s, the dominant neural IR systems were often multi-stage: one or several fast lexical, sparse, or dense retriever produces candidates; a stronger neural reranker refines them. Systems often combined lexical and semantic signals rather than replacing one with the other to capture a variety of signals [61, 62].

2.6 The Modern Stack: Hybrid, Contextual, and Agentic Retrieval

While the retrieve–rerank paradigm [61] remains widely used in production systems, recent years have seen substantial improvements to every stage of the retrieval pipeline and the emergence of novel ideas.

Modern general-purpose embeddings. General-purpose embedding models (bi-encoders) remain one of the central model families in modern IR. Their goal is to map queries and documents into a shared representation space that supports a wide range of retrieval and semantic-matching tasks. Progress in this area has often been driven by benchmark optimization, especially on IR benchmarks such as BEIR [65] and broader embedding benchmarks such as MTEB [66].⁴

Recent gains in embedding quality have come from several directions. One is improved training recipes and data construction. GTE [68] trains unified text embeddings through multi-stage contrastive learning over diverse datasets. NV-Retriever [69] emphasizes the importance of hard-negative mining, especially positive-aware methods that reduce the risk of treating genuinely relevant passages as negatives during contrastive training. ReasonIR [70] trains retrievers on synthetic queries and hard negatives designed to teach relevance as usefulness for inference, rather than merely direct lexical overlap or surface-level semantic similarity.

[61]: Nogueira et al. (2019), *Passage Re-ranking with BERT*

[62]: Yates et al. (2021), *Pretrained Transformers for Text Ranking: BERT and Beyond*

[63]: Johnson et al. (2017), *Billion-scale Similarity Search with GPUs*

[15]: Reimers et al. (2019), *Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks*

[64]: Karpukhin et al. (2020), *Dense Passage Retrieval for Open-Domain Question Answering*

[61]: Nogueira et al. (2019), *Passage Re-ranking with BERT*

[62]: Yates et al. (2021), *Pretrained Transformers for Text Ranking: BERT and Beyond*

[61]: Nogueira et al. (2019), *Passage Re-ranking with BERT*

[65]: Thakur et al. (2021), *BEIR: A Heterogenous Benchmark for Zero-shot Evaluation of Information Retrieval Models*

[66]: Muennighoff et al. (2022), *MTEB: Massive Text Embedding Benchmark*

4: General-purpose document embeddings are useful across many tasks, including retrieval, classification, clustering, natural language inference, and semantic similarity. These goals are related, but not identical. In IR, a document representation does not necessarily need to encode all document content; it only needs to support matching with relevant queries. In principle, a retrieval representation could function more like an index key than a faithful semantic summary. In practice, however, modern embedding models often do encode substantial document content, and recent work suggests that information about the original text can mostly be recovered from embeddings [67].

[68]: Li et al. (2023), *Towards General Text Embeddings with Multi-stage Contrastive Learning*

[69]: Moreira et al. (2024), *NV-Retriever: Improving Text Embedding Models with Effective Hard-Negative Mining*

[70]: Shao et al. (2025), *ReasonIR: Training Retrievers for Reasoning Tasks*

[71]: Wang et al. (2024), *Improving Text Embeddings with Large Language Models*

[72]: BehnamGhader et al. (2024), *LLM2Vec: Large Language Models Are Secretly Powerful Text Encoders*

[73]: Lee et al. (2024), *NV-Embed: Improved Techniques for Training LLMs as Generalist Embedding Models*

[74]: Weller et al. (2025), *Promptriever: Instruction-Trained Retrievers Can Be Prompted Like Language Models*

[75]: Weller et al. (2026), *On the Theoretical Limitations of Embedding-Based Retrieval*

[61]: Nogueira et al. (2019), *Passage Re-ranking with BERT*

[76]: Sun et al. (2023), *Is ChatGPT Good at Search? Investigating Large Language Models as Re-Ranking Agents*

[77]: Weller et al. (2025), *Rank1: Test-Time Compute for Reranking in Information Retrieval*

[78]: Khattab et al. (2020), *ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT*

A second major direction is the repurposing of large language models as embedding models. E5-Mistral [71] shows that strong embedding models can be built by adapting LLMs and training them on large synthetic, instruction-style datasets. LLM2Vec [72] converts decoder-only causal LLMs into bidirectional text encoders using bidirectional attention, masked next-token prediction, and contrastive learning. NV-Embed [73] further introduces practical techniques such as latent-attention pooling, removal of the causal mask during contrastive training, and two-stage instruction tuning. More recent work also asks whether the bi-encoder paradigm itself can become more controllable. *Promptriever* [74] shows that dense retrievers can be prompted in natural language, while preserving the efficiency advantages of independently encoded queries and documents. Instead of treating the query as a fixed semantic object, *Promptriever* conditions retrieval on per-instance instructions that redefine what should count as relevant.

Expressive Matching Functions. Single-vector representation must compress all potentially relevant aspects of a document into one point in embedding space, and cosine similarity can express only a limited class of query–document relationships. The LIMIT benchmark [75] shows theoretically and empirically that single-vector embedding models cannot represent all possible top- k relevance patterns unless the embedding dimension is sufficiently large; in practice, even strong embedding models fail on LIMIT, while lexical, multi-vector, and reranking approaches that leverage more expressive matching functions are less affected.

Cross-encoders [61] occupy the high-expressiveness end of this spectrum. Rather than encoding the query and document independently, they concatenate the query and candidate document and process them jointly with transformer self-attention. This allows every query token to attend to every document token, making the transformer itself a rich matching function. The cost is that documents cannot be precomputed as standalone vectors: each query–document pair must be scored separately, so these approaches are often used as a second reranking stage after first filtering with faster models.

LLM rerankers can be viewed as a compute-scaled version of the cross-encoder idea. Instead of using a relatively small supervised transformer reranker, an instruction-tuned LLM can be prompted to judge or rank candidate documents for a query. RankGPT [76] showed that ChatGPT and GPT-4 can act as strong zero-shot rerankers, including in listwise settings where the model compares several candidates and outputs an ordering. This setup leverages both the expressive self-attention mechanism of large transformers and the broader linguistic, reasoning, and instruction-following priors acquired during LLM training. Recent work further explores whether reranking can benefit from explicit test-time computation. Rank1 [77] trains rerankers to produce reasoning traces before judging relevance, distilling traces from a stronger reasoning model into smaller reranking models.

Late-interaction models provide one way to increase expressiveness without moving all the way to full cross-encoder reranking. Instead of representing each query or document with a single vector, models such as ColBERT [78] represent them as sets of contextualized token vectors. Documents can still be indexed in advance, but scoring is performed

through token-level interactions, typically using a MaxSim operator that matches each query token against its best document-token counterpart. Hypencoders [79] propose another way to improve matching expressivity: the query encoder produces a small neural network that scores document representations, enabling query-specific relevance functions. Like late interaction, Hypencoder can be seen as an attempt to move beyond fixed cosine similarity while preserving some of the efficiency advantages of retrieval-time indexing.

Generative information retrieval (GenIR) departs more radically from standard matching-based retrieval: instead of comparing a query representation against indexed document representations, a generative model directly produces relevant document identifiers given a query, typically by having assimilated the corpus content in parametric memory [80, 81]. While scaling GenIR to very large corpora remains difficult, it is an interesting concept because it exploits associative recall in a way standard retrieval models cannot.

Learned sparse retrieval. Sparse retrieval has also returned in neural form. Learned sparse models retain the efficiency and interpretability advantages of inverted indexes while learning term expansion and weighting from data. SPLADE [82] is a representative example: it learns sparse lexical representations using explicit sparsity regularization and log-saturated term weights, making it suitable for first-stage retrieval with standard inverted-index infrastructure.

Recent work pushes sparse retrieval beyond surface vocabulary terms. Sparse-autoencoder-based retrieval models such as SPLARE [83] learn sparse latent embeddings from LLM representations, aiming for more semantic and multilingual sparse features. The Latent Terms [84] line of work similarly shows that dense retrievers can contain extractable sparse, BM25-ready latent features, suggesting that dense and sparse retrieval are not as separate as they first appear.

Context-aware document representations. Many retrieval-augmented generation systems split documents into chunks before embedding them, but this can remove the broader context needed to interpret each passage. Late Chunking [85] addresses this by first encoding a longer text with a long-context embedding model and only then pooling token representations into chunk embeddings. As a result, each chunk vector can retain information from the surrounding document. In this line of thought, Contextual Document Embeddings [86] leverage priors on the corpus distribution when producing document embeddings to better express document specificity.

Modern Retrieval-Augmented Generation. Retrieval-Augmented Generation (RAG) [87] reframes retrieval as a conditioning mechanism for language generation. Instead of returning only a ranked list of documents, a retriever selects a small set of passages, and a language model generates an answer conditioned on this external context.⁵ In practice, this changes the role of ranking: the goal is no longer merely to order documents for a human reader, but to select evidence that will fit within the model's context window and support a faithful synthesis. RAG in its current form has become one of the main industrial use cases of modern generative AI.

[79]: Killingback et al. (2025), *Hypencoder: Hypernetworks for Information Retrieval*

[80]: Tay et al. (2022), *Transformer Memory as a Differentiable Search Index*

[81]: Li et al. (2024), *From Matching to Generation: A Survey on Generative Information Retrieval*

[82]: Formal et al. (2021), *SPLADE: Sparse Lexical and Expansion Model for First Stage Ranking*

[83]: Formal et al. (2026), *Learning Retrieval Models with Sparse Autoencoders*

[84]: Clavié et al. (2026), *Latent Terms: Dense Retrievers Contain Trivially Extractable BM25-ready Zipfian Vocabularies*

[85]: Günther et al. (2024), *Late Chunking: Contextual Chunk Embeddings Using Long-Context Embedding Models*

[86]: Morris et al. (2024), *Contextual Document Embeddings*

5: The original RAG paper introduced a trainable architecture that combined a pretrained seq2seq generator with a dense Wikipedia index, treating retrieved documents as latent variables marginalized during generation.[87] Modern usage of the term is broader: RAG now often refers to a production pattern in which an instruction-tuned LLM is supplied with retrieved external context at prompt time, usually with additional components such as chunking, hybrid retrieval, reranking, citation generation, query rewriting, and sometimes multi-step agentic search.

[88]: Nakano et al. (2021), *WebGPT: Browser-assisted Question-Answering with Human Feedback*

[89]: Yao et al. (2023), *ReAct: Synergizing Reasoning and Acting in Language Models*

[90]: Trivedi et al. (2023), *Interleaving Retrieval with Chain-of-Thought Reasoning for Knowledge-Intensive Multi-Step Questions*

[91]: Asai et al. (2024), *Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection*

[92]: Li et al. (2026), *Beyond Semantic Similarity: Rethinking Retrieval for Agentic Search via Direct Corpus Interaction*

[93]: Chen et al. (2026), *AgentIR: Reasoning-Aware Retrieval for Deep Research Agents*

Agentic Search. Beyond generating from a fixed set of retrieved passages, modern LLMs can participate directly in the search process. WebGPT [88] and ReAct [89] showed that language models can browse, call tools, inspect retrieved evidence, and decide which action to take next. This turns retrieval into an iterative loop: the model can decompose a task, issue searches, revise its query, and update its information need as evidence accumulates. IRCOT [90] makes this loop explicit for multi-hop question answering by interleaving retrieval with chain-of-thought reasoning, while Self-RAG [91] trains a model to decide when to retrieve and to critique both retrieved evidence and its own generations.

A related line of work questions the retrieval interface itself. *Direct Corpus Interaction* (DCI) [92] argues that exposing a corpus only through a fixed top- k retrieval API can be too restrictive for agents. Instead, an agent can interact directly with the raw corpus using general-purpose tools such as `grep`, file reads, shell commands, and lightweight scripts.

These new usage patterns change what retrievers should optimize for. When the retriever serves an agent rather than a human, the input is no longer just a standalone search query: it may include intermediate reasoning, subgoals, failed attempts, and partially discovered evidence. AgentIR [93] formalizes this setting by treating deep-research agents as retrieval users. Instead of embedding only the final query, it jointly embeds the agent's reasoning trace and query, allowing retrieval to reflect the agent's evolving intent.

Despite all this progress, the core objective remains similar: finding relevant information hidden in the sum of human knowledge.

"Presumably man's spirit should be elevated if he can better review his shady past and analyze more completely and objectively his present problems. He has built a civilization so complex that he needs to mechanize his record more fully if he is to push his experiment to its logical conclusion and not merely become bogged down part way there by overtaxing his limited memory. His excursion may be more enjoyable if he can reacquire the privilege of forgetting the manifold things he does not need to have immediately at hand, with some assurance that he can find them again if they prove important."

– Vannevar Bush, *As We May Think* (1945)

VISUAL DOCUMENT RETRIEVAL

ColPali: Efficient Document Retrieval with Vision Language Models

3

Paper Context

Released in June 2024 and published at ICLR 2025, this paper first proposed repurposing modern multimodal LLMs for representation learning. Its main contribution was to show that this idea is especially powerful for document retrieval: a field of major industrial value, where end-to-end visual retrieval offers a practical alternative to traditional parsing-heavy systems. In doing so, the paper helped launch what is now the active field of *Visual Document Retrieval* (VDR). A secondary interest is the use of *Late Interaction*. In this work, we show that late interaction is not only effective, but particularly well suited to VDR: it allows fine-grained associations between textual queries and visual document patches, while preserving strong retrieval performance.

Today, the open models released with this work have been downloaded more than 10 million times, and most major industrial players in the field—including NVIDIA, Meta, Amazon, Cohere, Google DeepMind, and MixedBread—have experimented with Visual Document Retrieval. Many of these efforts build on the training ideas introduced in this chapter, evaluate on the benchmarks released with this work, and often find their strongest performance in the late-interaction setting.

In the broader context of this thesis, this line of work offers two central takeaways. First, useful *priors* should be exploited when they are available. In document retrieval, layout, structure, and visual context provide rich signals that can substantially strengthen retrieval capabilities. At the same time, this work illustrates Sutton’s bitter lesson: by replacing handcrafted parsing pipelines with end-to-end systems, we remove brittle inductive biases that cap performance, and obtain stronger results with simpler frameworks. Second, *ColPali* shows that retrieval models can create and exploit associative links across modalities—here, between query terms and image patches. These links are powerful precisely because they go beyond words. They align with Vannevar Bush’s vision of search as a process of connection-making: not merely matching against a fixed rule set, but building useful associations across heterogeneous information.

- 3.1 Introduction 22
- 3.2 Problem Formulation & Related Work 23
 - 3.2.1 Textual Retrieval Methods 24
 - 3.2.2 Integrating Visual features 25
- 3.3 The ViDoRe Benchmark 25
 - 3.3.1 Benchmark Design 25
 - 3.3.2 Assessing Current Systems 26
- 3.4 Late interaction based Vision Retrieval 28
 - 3.4.1 Architecture 28
 - 3.4.2 Model training 28
- 3.5 Results 29
 - 3.5.1 Performance (R1) 29
 - 3.5.2 Latencies & Memory Footprint 31
 - 3.5.3 Interpretability 31
- 3.6 Ablation study 32
- 3.7 Conclusions 34

Paper Abstract

Documents are visually rich structures that convey information through text, but also figures, page layouts, tables, or even fonts. Since modern retrieval systems mainly rely on the textual information they extract from document pages to index documents -often through lengthy and brittle processes-, they struggle to exploit key visual cues efficiently. This limits their capabilities in many practical document retrieval applications such as Retrieval Augmented Generation (RAG). To benchmark current systems on visually rich document retrieval, we introduce the Visual Document Retrieval Benchmark ViDoRe, composed of various page-level retrieval tasks spanning multiple domains, languages, and practical settings. The inherent complexity and performance shortcomings of modern systems motivate a new concept; doing document retrieval by directly embedding the images of the document pages. We release *ColPali*, a Vision Language Model trained to produce high-quality multi-vector embeddings from images of document pages. Combined with a late interaction matching mechanism, *ColPali* largely outperforms modern document retrieval pipelines while being drastically simpler, faster and end-to-end trainable. We release models, data, code and benchmarks under open licenses at <https://hf.co/vidore>.

3.1 Introduction

Document Retrieval consists of matching a user query to relevant documents in a given corpus. It is central to many widespread industrial applications, either as a standalone ranking system (search engines) or as part of more complex information extraction or Retrieval Augmented Generation (RAG) pipelines.

Over recent years, pretrained language models have enabled large improvements in text embedding models. In practical industrial settings, however, the primary performance bottleneck for efficient document retrieval stems not from embedding model performance but from the prior data ingestion pipeline. Indexing a standard PDF document involves several steps. First, PDF parsers or Optical Character Recognition (OCR) systems are used to extract words from the pages. Document layout detection models can then be run to segment paragraphs, titles, and other page objects such as tables, figures, and headers. A chunking strategy is then defined to group text passages with some semantical coherence, and modern retrieval setups may even integrate a captioning step to describe visually rich elements in a natural language form, more suitable for embedding models. In our experiments (Table 3.2), we typically find that optimizing the ingestion pipeline yields much better performance on visually rich document retrieval than optimizing the text embedding model.

Contribution 1: ViDoRe. In this work, we argue that document retrieval systems should not be evaluated solely on the capabilities of text embedding models [65, 66, 94], but should also consider the context and visual elements of the documents to be retrieved. To this end, we create and openly release ViDoRe, a comprehensive benchmark to evaluate systems

[65]: Thakur et al. (2021), *BEIR: A Heterogenous Benchmark for Zero-shot Evaluation of Information Retrieval Models*

[66]: Muennighoff et al. (2022), *MTEB: Massive Text Embedding Benchmark*

[94]: Bajaj et al. (2016), *MS MARCO: A Human Generated Machine Reading Comprehension Dataset*

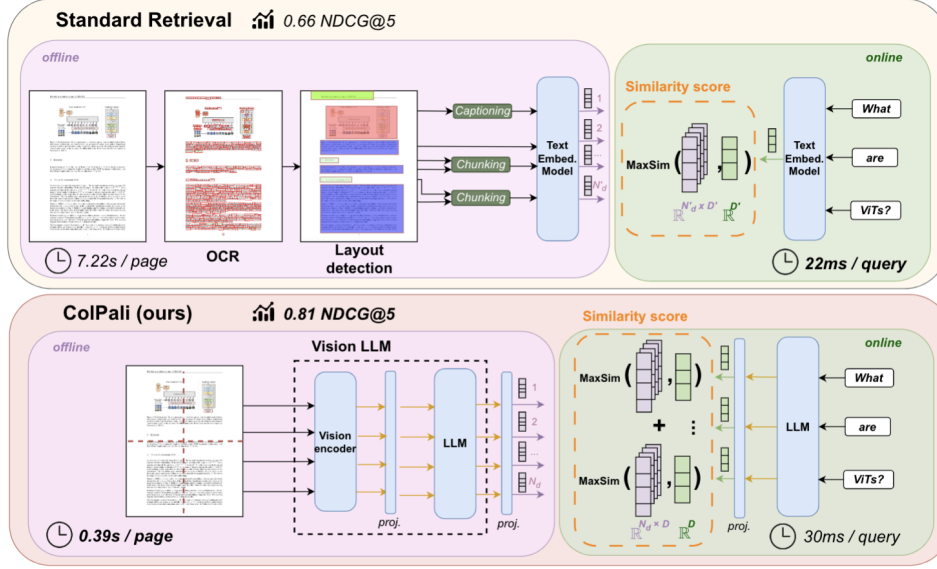


Figure 3.1: ColPali simplifies document retrieval w.r.t. standard retrieval methods while achieving stronger performances with better latencies. Latencies and results are detailed in section 3.5 and subsection C.2.4.

on page-level document retrieval with a wide coverage of domains, visual elements, and languages. ViDoRe addresses practical document retrieval scenarios, where queries often necessitate both textual and visual understanding for accurate document matching. We highlight the shortcomings of current text-centric systems in these settings.¹

Contribution 2: ColPali. We propose a novel concept and model architecture based on Vision Language Models (VLMs) to efficiently index documents purely from their visual features, allowing for subsequent fast query matching with late interaction mechanisms [78]. Our method, ColPali, significantly outperforms all other retrieval systems on ViDoRe while being fast and end-to-end trainable. These results demonstrate the potential and the many benefits of this novel *Retrieval in Vision Space* concept, which could significantly alter the way document retrieval is approached in the industry moving forward. We release all resources at <https://hf.co/vidore>.

1: The ViDoRe benchmark leaderboard is hosted publicly at <https://huggingface.co/spaces/vidore/vidore-leaderboard> to encourage further developments.

[78]: Khattab et al. (2020), ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT

3.2 Problem Formulation & Related Work

Problem Setting. In our setting, a retrieval system scores how relevant a document d from corpus $\mathcal{D}$ is with respect to a query q . Computing the similarity score $s(q, d) \in \mathbb{R}$ for each of the $|\mathcal{D}|$ documents in the corpus creates a ranking we can use to extract the most relevant documents. In this work, we focus on page-level retrieval: *given a query, is the correct document page retrieved by the system?* For coherence with existing literature, we further use the term *document* to refer to individual pages, i.e. the atomic retrieved elements in our setting. As we focus on practical industrial retrieval applications (RAG, search engines) with potentially large corpora sizes, latency constraints are imposed on scoring systems. Most current retrieval systems can be decomposed into (1) an *offline* indexing phase in which a document index is built and (2) an *online* querying phase in

which a query is matched to documents from the index and where low latency is vital to the user experience.

Under these industrial constraints, we identify three main properties an efficient document retrieval system should exhibit: (R1) strong retrieval performance, as measured by standard retrieval metrics; (R2) fast online querying, measured through average latencies; and (R3) high-throughput corpus indexation, i.e. the number of pages that can be embedded in a given time frame.

3.2.1 Textual Retrieval Methods

Document Retrieval in Text Space.

Statistical methods based on word frequency like TF-IDF [35] and BM25 [40] are still widely used due to their simplicity and efficiency. More recently, neural embedding models based on fine-tuned large language models display state-of-the-art performance on a variety of text embedding tasks and top the retrieval leaderboards [66].

Neural Retrievers. In bi-encoder models [15, 64, 95], documents are independently mapped *offline* to a dense vector space. Queries are embedded *online* and matched to documents through a fast cosine distance computation. A slower, but slightly more performant alternative, cross-encoder systems [96, 97] concatenate query and document as a single input sequence and iteratively attribute matching scores to each possible combination. This enables full attention computation between query and document terms but comes at the cost of computational efficiency, as $|\mathcal{D}|$ encoding passes must be done online.

Multi-Vector retrieval via late interaction. In the late interaction paradigm introduced by ColBERT [78], an embedding is pre-computed and indexed per document token. At runtime, similarity can be computed with individual query token embeddings. The idea is to benefit from the rich interaction between individual query and document terms while taking advantage of the offline computation and fast query matching enabled by bi-encoders. See section C.5 for more details.

Retrieval Evaluation. Although benchmarks and leaderboards have been developed to evaluate text embedding models [65, 66], much of the performance improvements in industrial use cases of embedding models stem from the prior data ingestion pipeline. While documents often rely on visual elements to more efficiently convey information to human readers, text-only systems barely tap into these visual cues. Other work has also independently studied table or chart retrieval systems through repurposed Question Answering datasets [98, 99], but these studies assess only specialized methods for each task.

To our knowledge, no benchmark evaluates document retrieval systems in practical settings; in an end-to-end manner, across several document types and topics, and by evaluating the use of both textual and visual document features.

[35]: Spärck Jones (1972), *A Statistical Interpretation of Term Specificity and Its Application in Retrieval*

[40]: Robertson et al. (1994), *Okapi at TREC-3*

[66]: Muennighoff et al. (2022), *MTEB: Massive Text Embedding Benchmark*

[15]: Reimers et al. (2019), *Sentence-BERT: Sentence Embeddings using Siamese BERT- Networks*

[64]: Karpukhin et al. (2020), *Dense Passage Retrieval for Open-Domain Question Answering*

[95]: Wang et al. (2022), *Text Embeddings by Weakly-Supervised Contrastive Pre-training*

[96]: Wang et al. (2020), *MiniLM: Deep Self-Attention Distillation for Task-Agnostic Compression of Pre-Trained Transformers*

[97]: Cohere (2024), *Introducing Rerank 3: A New Foundation Model for Efficient Enterprise Search & Retrieval*

[78]: Khattab et al. (2020), *ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT*

[65]: Thakur et al. (2021), *BEIR: A Heterogenous Benchmark for Zero-shot Evaluation of Information Retrieval Models*

[66]: Muennighoff et al. (2022), *MTEB: Massive Text Embedding Benchmark*

[98]: Zhang et al. (2019), *Table2Vec: Neural Word and Entity Embeddings for Table Population and Retrieval*

[99]: Nowak et al. (2024), *Multimodal Chart Retrieval: A Comparison of Text, Table and Image Based Approaches*

3.2.2 Integrating Visual features

Contrastive Vision Language Models. Mapping latent representations of textual content to corresponding representations of visual content has been done by aligning disjoint visual and text encoders through contrastive losses [100, 101]. While some OCR capabilities exist in these models, the visual component is often not optimized for text understanding.

The Fine-grained Interactive Language-Image Pre-training [102] framework extends the late interaction mechanism to cross-modal Vision Language Models, relying on max similarity operations between text tokens and image patches.

Visually Rich Document Understanding. To go beyond text, some document-focused models jointly encode text tokens alongside visual or document layout features [103–106]. Large Language transformer Models (LLMs) with strong reasoning capabilities have recently been combined with Vision Transformers (ViTs) [107] to create VLMs [108–111] where image patch vectors from contrastively trained ViT models [101] are fed as input embeddings to the LLM and concatenated with the text-token embeddings.

PaliGemma. The PaliGemma-3B model [112] extends concepts from Pali3 [113], and projects SigLIP-So400m/14 [114] patch embeddings into Gemma-2B’s text vector space [115]. Along with its reasonable size w.r.t. other performant VLMs, an interesting property of PaliGemma’s text model is that it is fine-tuned with full-block attention on the prefix (instruction text and image tokens). See section C.5 for more details.

VLMs display enhanced capabilities in Visual Question Answering, captioning, and document understanding [116], but are not optimized for retrieval tasks.

3.3 The ViDoRE Benchmark

Existing benchmarks for contrastive vision-language models primarily evaluate retrieval for natural images [117–119]. On the other hand, textual retrieval benchmarks [66] are evaluated at the textual passage level and are not tailored for document retrieval tasks. We fill the gap with ViDoRE, a comprehensive benchmark for document retrieval using visual features.

3.3.1 Benchmark Design

ViDoRE is designed to comprehensively evaluate retrieval systems on their capacity to match queries to relevant documents at the page level. This benchmark encompasses multiple orthogonal subtasks, with focuses on various modalities - text, figures, infographics, tables; thematic domains - medical, business, scientific, administrative; or languages - English, French. Tasks also span varying levels of complexity, in order to capture signals from both weaker and stronger systems. As many systems require large amounts of time to index pages (captioning-based approaches can take dozens of seconds per page for instance), we limit the number of

[100]: Radford et al. (2021), *Learning Transferable Visual Models From Natural Language Supervision*

[101]: Zhai et al. (2023), *Sigmoid Loss for Language Image Pre-Training*

[102]: Yao et al. (2021), *FILIP: Fine-grained Interactive Language-Image Pre-Training*

[103]: Appalaraju et al. (2021), *DocFormer: End-to-End Transformer for Document Understanding*

[104]: Kim et al. (2021), *OCR-free Document Understanding Transformer*

[105]: Huang et al. (2022), *LayoutLMv3: Pre-training for Document AI with Unified Text and Image Masking*

[106]: Tang et al. (2022), *Unifying Vision, Text, and Layout for Universal Document Processing*

[107]: Dosovitskiy et al. (2020), *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*

[108]: Alayrac et al. (2022), *Flamingo: a Visual Language Model for Few-Shot Learning*

[109]: Liu et al. (2023), *Visual Instruction Tuning*

[110]: Bai et al. (2023), *Qwen-VL: A Versatile Vision-Language Model for Understanding, Localization, Text Reading, and Beyond*

[111]: Laurençon et al. (2024), *What matters when building vision-language models?*

[101]: Zhai et al. (2023), *Sigmoid Loss for Language Image Pre-Training*

[112]: Beyer et al. (2024), *PaliGemma: A versatile 3B VLM for transfer*

[113]: Chen et al. (2023), *PaLI-3 Vision Language Models: Smaller, Faster, Stronger*

[114]: Alabdulmohsin et al. (2023), *Getting ViT in Shape: Scaling Laws for Compute-Optimal Model Design*

[115]: Gemma Team et al. (2024), *Gemma: Open Models Based on Gemini Research and Technology*

[116]: Yue et al. (2023), *MMMU: A Massive Multi-discipline Multimodal Understanding and Reasoning Benchmark for Expert AGI*

[117]: Lin et al. (2014), *Microsoft COCO: Common Objects in Context*

[118]: Borchmann et al. (2021), *DUE: End-to-End Document Understanding Benchmark*

[119]: Thapliyal et al. (2022), *Crossmodal-3600: A Massively Multilingual Multimodal Evaluation Dataset*

[66]: Muennighoff et al. (2022), *MTEB: Massive Text Embedding Benchmark*

candidate documents for each retrieval task in order to evaluate even complex systems in a reasonable timeframe without sacrificing quality. For trainable retrieval systems, we provide a reference training set that can be used to facilitate comparisons.

Table 3.1: ViDoRe comprehensively evaluates multimodal retrieval methods.

Dataset	Language	# Queries	# Documents	Description
Academic Tasks				
DocVQA	English	500	500	Scanned documents from UCSF Industry
InfoVQA	English	500	500	Infographics scraped from the web
TAT-DQA	English	1600	1600	High-quality financial reports
arXivVQA	English	500	500	Scientific Figures from arXiv
TabFQuAD	French	210	210	Tables scraped from the web
Practical Tasks				
Energy	English	100	1000	Documents about energy
Government	English	100	1000	Administrative documents
Healthcare	English	100	1000	Medical documents
AI	English	100	1000	Scientific documents related to AI
Shift Project	French	100	1000	Environmental reports

Academic Tasks. We repurpose widely used visual question-answering benchmarks for retrieval tasks: for each page-question-answer triplet, we use the question as the query, and the associated page as the gold document (Table 3.1). These academic datasets either focus on single specific modalities [120–122] or target more varied visually rich documents [123]. Moreover, we consider TabFQuAD, a human-labeled dataset on tables extracted from French industrial PDF documents released with this work. Details can be found in subsection C.1.1.

Practical tasks. We construct topic-specific retrieval benchmarks spanning multiple domains to go beyond repurposed QA datasets and evaluate retrieval in more realistic industrial situations (e.g. RAG). To achieve this, we collect publicly accessible PDF documents and generate queries pertaining to document pages using Claude-3 Sonnet, a high-quality proprietary vision-language model [124]. In total, we collect 1,000 document pages per topic, which we associate with 100 queries extensively filtered for quality and relevance by human annotators. The corpus topics are intentionally specific to maximize syntactic proximity between documents, creating more challenging retrieval tasks and covering an array of orthogonal domains (Table 3.1).²

Evaluation Metrics. We evaluate performance on our benchmark (Requirement R1) using standard metrics from the retrieval literature (nDCG, Recall@K, MRR). We report nDCG@5 values as the main performance metric in this work and release the complete sets of results along with the models³. To validate compliance with practical industrial requirements (section 3.2), we also consider query latencies (R2) and indexing throughputs (R3).

3.3.2 Assessing Current Systems

Unstructured. We evaluate retrieval systems representative of those found in standard industrial RAG pipelines. As is common practice, we rely on the Unstructured⁴ off-the-shelf tool in the highest resolution

[120]: Mathew et al. (2020), *DocVQA: A Dataset for VQA on Document Images*

[121]: Mathew et al. (2021), *InfographicVQA*

[122]: Li et al. (2024), *Multimodal ArXiv: A Dataset for Improving Scientific Comprehension of Large Vision-Language Models*

[123]: Zhu et al. (2022), *Towards Complex Document Understanding By Discrete Reasoning*

[124]: Anthropic (2024), *The Claude 3 Model Family: Opus, Sonnet, Haiku*

2: Answers are generated alongside queries to (1) ground queries and improve their quality and (2) provide resources to foster future work.

3: <https://huggingface.co/vidore>

4: www.unstructured.io

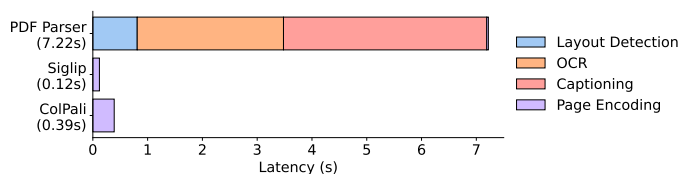
settings to construct high-quality text chunks from PDF documents. Unstructured orchestrates the document parsing pipeline, relying on deep learning vision models to detect titles and document layouts [125], OCR engines [126] to extract text in non-native PDFs, specialized methods or models to detect and reconstruct tables, and implements a chunking strategy (by-title) that leverages the detected document structure to preserve section boundaries when concatenating texts. As is common practice, in our simplest Unstructured configuration (*text-only*), only textual elements are kept and figures, images, and tables are considered noisy information and are filtered out.

Unstructured + X. While Unstructured is a strong baseline by itself, we further augment Unstructured’s output by integrating the visual elements. In (+ OCR), tables, charts, and images are run through an OCR engine, processed by Unstructured, and chunked independently. In (+ Captioning), we set up a fully-fledged captioning strategy [127], in which we feed visual elements to a strong proprietary Vision Language Model (Claude-3 Sonnet [124]) to obtain highly detailed textual descriptions of the elements. Both strategies aim to integrate visual elements in the retrieval pipeline but incur significant latency and resource costs (subsection 3.5.2).

Embedding Model. To embed textual chunks, we evaluate Okapi BM25, the *de facto* standard sparse statistical retrieval method, and the dense encoder of BGE-M3 [128], a multilingual neural method with SOTA performance in its size category. Chunks are embedded and scored independently, and page-level scores are obtained by max-pooling over the page’s chunk scores.⁵

Contrastive VLMs. We also evaluate the strongest available vision-language embedding models; Jina CLIP [129], Nomic Embed Vision [130], and SigLIP-So400m/14 [114].

Results. From a performance perspective, the best results are obtained by combining the Unstructured parser with visual information, either from captioning strategies or by running OCR on the visual elements (Table 3.2). Little difference is seen between BM25 and BGE-M3 embeddings, highlighting the visual information bottleneck. Contrastive VLMs lag behind. Beyond retrieval performance (R1), the indexing latencies (R2) reported in Figure 3.2 illustrate that PDF parsing pipelines can be very lengthy, especially when incorporating OCR or captioning strategies. Querying latencies at runtime (R3) are very good for all evaluated systems (≤ 22 ms on an NVIDIA L4) due to fast query encoding and cosine similarity matching.



[125]: Ge et al. (2021), *YOLOX: Exceeding YOLO Series in 2021*

[126]: Smith (2007), *An Overview of the Tesseract OCR Engine*

[127]: Zhao et al. (2023), *Retrieving Multimodal Information for Augmented Generation: A Survey*

[124]: Anthropic (2024), *The Claude 3 Model Family: Opus, Sonnet, Haiku*

[128]: Chen et al. (2024), *BGE M3-Embedding: Multi-Lingual, Multi-Functionality, Multi-Granularity Text Embeddings Through Self-Knowledge Distillation*

5: We empirically validated the max-pooling strategy over sub-page chunks to be more effective than concatenating all page chunks before embedding page-wise.

[129]: Koukounas et al. (2024), *Jina CLIP: Your CLIP Model Is Also Your Text Retriever*

[130]: Nomic (2024), *Nomic Embed Vision: Expanding The Nomic Latent Space*

[114]: Alabdulmohsin et al. (2023), *Getting ViT in Shape: Scaling Laws for Compute-Optimal Model Design*

Figure 3.2: Offline document indexing with *ColPali* is much simpler and faster compared to standard retrieval methods. The PDF Parser results are obtained following the *Unstructured* settings with BGE-M3 detailed in subsection 3.3.2. All indexing speeds are averaged per-page latencies. More details in subsection C.2.4

3.4 Late interaction based Vision Retrieval

3.4.1 Architecture

Vision-Language Models. Encouraged by their strong document understanding capabilities, we propose adapting recent VLMs for retrieval. The key concept is to leverage the alignment between output embeddings of text and image tokens acquired during multi-modal fine-tuning. To this extent, we introduce *ColPali*, a Paligemma-3B extension that is capable of generating ColBERT-style multi-vector representations of text and images (Figure 3.1). PaliGemma-3B is a strong candidate due to its small size, the many released checkpoints fine-tuned for different image resolutions and tasks, and its promising performance on various document understanding benchmarks. We add a projection layer to map each of the language model’s output token embeddings (whether from text or image tokens) to a vector space of reduced dimension $D = 128$ as used in the ColBERT paper [78] to keep lightweight bag-of-embedding representations.

[78]: Khattab et al. (2020), *ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT*

Late Interaction. Given query q and document d , we denote as $\mathbf{E}_q \in \mathbb{R}^{N_q \times D}$ and $\mathbf{E}_d \in \mathbb{R}^{N_d \times D}$ their respective multi-vector representation in the common embedding space $\mathbb{R}^D$, where N_q and N_d are respectively the number of vectors in the query and in the document page embeddings. The late interaction operator, $\text{LI}(q, d)$, is the sum over all query vectors $\mathbf{E}_q^{(j)}$, of its maximum dot product $\langle \cdot | \cdot \rangle$ with each of the N_d document embedding vectors $\mathbf{E}_d^{(1:N_d)}$.

$$\text{LI}(q, d) = \sum_{i \in [1, N_q]} \max_{j \in [1, N_d]} \langle \mathbf{E}_q^{(i)} | \mathbf{E}_d^{(j)} \rangle \quad (3.1)$$

Contrastive Loss. The Late Interaction operation is fully differentiable, enabling backpropagation. Let a batch $\{q_k, d_k\}_{k \in [1, b]}$ composed of b query-page pairs, where for all $k \in [1, b]$, the document page d_k is the document corresponding to query q_k . Following [78], we define our in-batch contrastive loss $\mathcal{L}$ as the softmaxed cross-entropy of the positive scores $s_k^+ = \text{LI}(q_k, d_k)$ w.r.t. the maximal in-batch negative scores $s_k^- = \max_{l, l \neq k} \text{LI}(q_k, d_l)$ ⁶:

[78]: Khattab et al. (2020), *ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT*

6: We reformulate the loss to leverage the numerically stable `softplus` function where $\text{softplus}(x) = \log(1 + \exp(x))$

$$\mathcal{L} = -\frac{1}{b} \sum_{k=1}^b \log \left[\frac{\exp(s_k^+)}{\exp(s_k^+) + \exp(s_k^-)} \right] = \frac{1}{b} \sum_{k=1}^b \log(1 + \exp(s_k^- - s_k^+)) \quad (3.2)$$

3.4.2 Model training

Dataset. Our training dataset of 118,695 query-page pairs comprises training sets of openly available academic datasets (63%) and a synthetic dataset made up of pages from web-crawled PDF documents and augmented with VLM-generated (Claude-3 Sonnet) pseudo-questions (37%). Dataset split details are given in subsection C.1.3. Our training

set is fully English by design, enabling us to study zero-shot generalization to non-English languages⁷. We explicitly verify no multi-page PDF document is used both ViDoRe and in the train set to prevent evaluation contamination. A validation set is created with 2% of the samples to tune hyperparameters. We openly release the training dataset⁸ for reproducibility and to encourage further research.

Parameters. All models are trained for 1 epoch on the train set. Unless specified otherwise, we train models in bfloat16 format, use low-rank adapters (LoRA, [131]) with $\alpha = 32$ and $r = 32$ on the transformer layers from the language model, as well as the final randomly initialized projection layer, and use a paged_adamw_8bit optimizer. We train on an 8 GPU setup with data parallelism, a learning rate of $5e - 5$ with linear decay with 2.5% warmup steps, and a batch size of 32.

Query Augmentation. As in [78], we append 5 <unused0> tokens to the query tokens to serve as a soft, differentiable query expansion or re-weighting mechanism.

7: Multilingual data is present in the pre-training corpus of the language model (Gemma-2B) and potentially occurs during PaliGemma-3B’s multimodal training.

8: https://huggingface.co/datasets/vidore/colpali_train_set

[131]: Hu et al. (2021), *LoRA: Low-Rank Adaptation of Large Language Models*

[78]: Khattab et al. (2020), *ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT*

3.5 Results

Table 3.2: Comprehensive evaluation of baseline models and our proposed method on ViDoRe. Results are presented using nDCG@5 metrics, and illustrate the impact of different components. Text-only metrics are not computed for benchmarks with only visual elements.

	ArxivQ	DocQ	InfoQ	TabF	TATQ	Shift	AI	Energy	Gov.	Health.	Avg.
Unstructured text-only											
- BM25	-	34.1	-	-	44.0	59.6	90.4	78.3	78.8	82.6	-
- BGE-M3	-	28.4 _{↓5.7}	-	-	36.1 _{↓7.9}	68.5 _{↑8.9}	88.4 _{↓2.0}	76.8 _{↓1.5}	77.7 _{↓1.1}	84.6 _{↑2.0}	-
Unstructured + OCR											
- BM25	31.6	36.8	62.9	46.5	62.7	64.3	92.8	85.9	83.9	87.2	65.5
- BGE-M3	31.4 _{↓0.2}	25.7 _{↓11.1}	60.1 _{↓2.8}	70.8 _{↑24.3}	50.5 _{↓12.2}	73.2 _{↑8.9}	90.2 _{↓2.6}	83.6 _{↓2.3}	84.9 _{↑1.0}	91.1 _{↑3.9}	66.1 _{↑0.6}
Unstructured + Captioning											
- BM25	40.1	38.4	70.0	35.4	61.5	60.9	88.0	84.7	82.7	89.2	65.1
- BGE-M3	35.7 _{↓4.4}	32.9 _{↓5.4}	71.9 _{↑1.9}	69.1 _{↑33.7}	43.8 _{↓17.7}	73.1 _{↑12.2}	88.8 _{↑0.8}	83.3 _{↓1.4}	80.4 _{↓2.3}	91.3 _{↑2.1}	67.0 _{↑1.9}
Contrastive VLMs											
Jina-CLIP	25.4	11.9	35.5	20.2	3.3	3.8	15.2	19.7	21.4	20.8	17.7
Nomic-vision	17.1	10.7	30.1	16.3	2.7	1.1	12.9	10.9	11.4	15.7	12.9
SigLIP (Vanilla)	43.2	30.3	64.1	58.1	26.2	18.7	62.5	65.7	66.1	79.1	51.4
Ours											
SigLIP (Vanilla)	43.2	30.3	64.1	58.1	26.2	18.7	62.5	65.7	66.1	79.1	51.4
BiSigLIP (+fine-tuning)	58.5 _{↑15.3}	32.9 _{↑2.6}	70.5 _{↑6.4}	62.7 _{↑4.6}	30.5 _{↑4.3}	26.5 _{↑7.8}	74.3 _{↑11.8}	73.7 _{↑8.0}	74.2 _{↑8.1}	82.3 _{↑3.2}	58.6 _{↑7.2}
BiPali (+LLM)	56.5 _{↓2.0}	30.0 _{↓2.9}	67.4 _{↓3.1}	76.9 _{↑14.2}	33.4 _{↑2.9}	43.7 _{↑17.2}	71.2 _{↓3.1}	61.9 _{↓11.7}	73.8 _{↓0.4}	73.6 _{↓8.8}	58.8 _{↑0.2}
ColPali (+Late Inter.)	79.1 _{↑22.6}	54.4 _{↑24.5}	81.8 _{↑14.4}	83.9 _{↑7.0}	65.8 _{↑32.4}	73.2 _{↑29.5}	96.2 _{↑25.0}	91.0 _{↑29.1}	92.7 _{↑18.9}	94.4 _{↑20.8}	81.3 _{↑22.5}

3.5.1 Performance (R1)

We show performance is achieved iteratively through the combination of three factors; (1) a carefully crafted task-specific dataset, (2) pairing a pretrained LLM to a vision model to better leverage text semantics from the image, and (3) using multi-vector embeddings rather than a single vector representation to better capture the vast amount of visual information present in a document.

9: <https://huggingface.co/google/siglip-so400m-patch14-384>

[113]: Chen et al. (2023), *PaLI-3 Vision Language Models: Smaller, Faster, Stronger*

10: Note that the SigLIP model used in PaliGemma slightly differs in terms of number patches - 1024 patches for PaliGemma’s vision encoder, and 729 for the standalone SigLIP model.

11: This can be explained by the fact that contrary to SigLIP, the original PaliGemma is not trained on contrastive matching tasks, but rather on next token prediction. Our contrastive fine-tuning phase on 119K images to transform PaliGemma into a bi-encoder is 5 orders of magnitude smaller than SigLIP’s original contrastive training.

Fine-tuning a Vision Model on a document-retrieval-oriented dataset:

BiSigLIP. SigLIP⁹ is a strong vision-language bi-encoder producing single-vector embeddings, pretrained on billions of image-text pairs from the English split of WebLI [113]. Further fine-tuning the textual component of this model on our document-oriented dataset (BiSigLIP) yields clear improvements across the board, particularly on figure retrieval (ArxivQA) and table retrieval tasks (TabFQuAD).

Feeding image patches to a LLM: BiPali. In the PaliGemma model architecture, SigLIP-generated patch embeddings are fed to a text language model and we can obtain LLM contextualized output patch embeddings.¹⁰ This technique aligns the image token representations with the text token embeddings in the LLM’s embeddings space, and augments the vision model embeddings with the language model’s text understanding capabilities. We average pool these representations to obtain a single dense vector, effectively creating a PaliGemma bi-encoder model (BiPali). After fine-tuning on the training dataset, we obtain a model that performs slightly worse in English than the tuned BiSigLIP variant.¹¹ However, we see notable improvements in French tasks, indicating that BiPali’s LLM (Gemma 2B) helps multilingual text understanding. This is particularly notable as our training dataset does not contain non-English samples.

Leveraging Multi-Vector Embeddings through Late Interaction: ColPali.

One benefit of inputting image patch embeddings through a language model is that they are natively mapped to a latent space similar to the textual input (query). This enables leveraging the ColBERT strategy to construct one embedding per image patch token, and at inference compute all interactions between text tokens and image patches, resulting in a step-change improvement in performance compared to BiPali. Results in Table 3.2 show that our *ColPali* model also largely outperforms the strong baselines based on Unstructured and captioning, as well as all evaluated text-image embedding models. The difference is particularly stark on the more visually complex benchmark tasks, such as InfographicVQA, ArxivQA, and TabFQuAD, respectively representing infographics, figures, and tables. However, text-centric documents are also better retrieved by the *ColPali* models across all evaluated domains and languages, making our approach the overall best-performing document-retrieval model.

Negative Results. For extensiveness, we also train ColSigLIP, a late interaction variant of the BiSigLIP model, but obtain abysmal performance. We attribute this to the large gaps w.r.t. SigLIP’s pre-training, in which only a pooled latent representation is used in the contrastive loss, which does not optimize the representations of individual patch and token embeddings. Similarly, we train a BiSigLIP_{PaliGemma} variant, in which we retrieve the image representations from the SigLIP model that has been further updated by PaliGemma fine-tuning, and use the text representations from PaliGemma’s text model. After fine-tuning on our dataset, performance is severely inferior to that of SigLIP_{Vanilla}, which simply encodes with SigLIP’s original text and vision components. This indicates a logical misalignment between SigLIP embeddings and Gemma embeddings after PaliGemma training. We detail these results in subsection C.3.1.

3.5.2 Latencies & Memory Footprint

Online Querying. (R2) Logically, querying latencies differ between *ColPali* and a BGE-M3 embedding model. For BGE, encoding takes about 22 ms for 15 tokens, while encoding a query with *ColPali*'s language model takes about 30 ms¹². For smaller corpus sizes, computing the late interaction operation induces only marginal overhead (≈ 1 ms per 1000 pages in the corpus), and the cosine similarity computation between bi-encoder vectors is even faster. Optimized late interaction engines [132, 133] make it easy to scale corpus sizes to millions of documents with reduced latency degradation.

Offline Indexing. (R3) Standard retrieval methods using bi-encoders represent each chunk as a single vector embedding, which is easy to store and fast to compute. However, processing a PDF to obtain the different chunks is the most time-consuming part (layout detection, OCR, chunking), and using captioning to handle multimodal data will only exacerbate this already lengthy process. On the other hand, *ColPali* directly encodes pages from their image representation. Although the model is larger than standard retrieval encoders, skipping preprocessing allows large indexing speedups¹³ (Figure 3.2). As pages are embedded end-to-end in a single forward pass, VRAM usage depends exclusively on the sequence length (number of patches per image), which is fixed, enabling efficient batching strategies that fully leverage hardware acceleration. *ColPali* also benefits from most LLM efficiency improvements introduced in the ecosystem, such as Flash Attention [134].

Storage Footprint. Our method requires storing a vector per image patch, along with 6 extra text tokens “Describe the image” concatenated to image patches. We project each PaliGemma vector to a lower dimensional space ($D = 128$) to maximize efficiency, leading to a memory footprint of 257.5 KB per page (subsection C.2.3). Importantly, the memory footprint of the naive ColBERT indexing strategy can be drastically improved through compression and clustering mechanisms [132, 135].

Token pooling. Token pooling [135] is a CRUDE-compliant method (document addition/deletion-friendly) that aims to reduce the number of vectors in multi-vector embeddings. For *ColPali*, many image patches share redundant information, e.g. white background patches. By pooling these patches together, we can reduce the number of embeddings while retaining most information. Retrieval performance with hierarchical mean token pooling on image embeddings is shown in Figure 3.3. With a pool factor of 3, the total number of vectors is reduced by 66.7% while 97.8% of the original performance is maintained. We note that the Shift dataset—composed of the most text-dense documents—is a clear outlier, showing that more information-dense documents contain fewer redundant patches and may be prone to greater performance degradation with such pooling techniques.

3.5.3 Interpretability

By superimposing the late interaction heatmap on top of the original image, we can visualize the most salient image patches with respect to each term of the query, yielding interpretable insights into model

12: Computed for a batch size of 1 (online), and averaged over 1000 queries. See subsection C.2.4

[132]: Santhanam et al. (2022), *PLAID: An Efficient Engine for Late Interaction Retrieval*

[133]: Lee et al. (2023), *Rethinking the Role of Token Retrieval in Multi-Vector Retrieval*

13: Measured on an NVIDIA L4 GPU and averaged over 100 pages, with a batch size of 4 pages for *ColPali* and 8 text chunks for Bi-Encoders. On average, a page is divided into 2.1 chunks. See subsection C.2.4.

[134]: Dao (2023), *FlashAttention-2: Faster Attention with Better Parallelism and Work Partitioning*

[132]: Santhanam et al. (2022), *PLAID: An Efficient Engine for Late Interaction Retrieval*

[135]: Clavié et al. (2024), *Reducing the Footprint of Multi-Vector Retrieval with Minimal Performance Impact via Token Pooling*

[135]: Clavié et al. (2024), *Reducing the Footprint of Multi-Vector Retrieval with Minimal Performance Impact via Token Pooling*

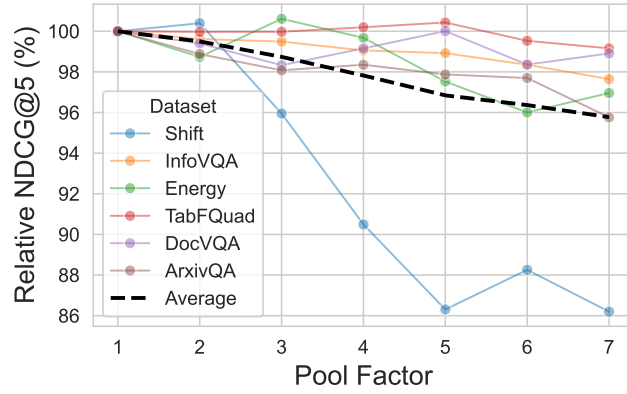
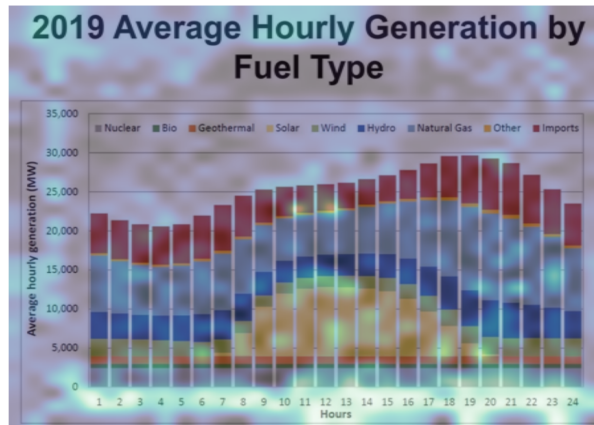


Figure 3.3: Relative performance degradation when reducing the number of stored embeddings per document.



Query: "Which hour of the day had the highest overall eletricity generation in 2019?"

Figure 3.4: For each term in a user query, ColPali identifies the most relevant document image patches (highlighted zones) and computes a query-to-page matching score.

focus zones. As epitomized in Figure 3.4, we observe ColPali exhibits strong OCR capabilities as both the words “hourly” and “hours” present a high similarity score with the query token $\langle_hour\rangle$. We also note particular focus on other non-trivial image features such as the x-axis representing hours being salient. Other visualization examples are shown in section C.4.

3.6 Ablation study

We run various ablations to better understand the mechanisms at play. By default, result deltas reported below refer to nDCG@5 values averaged over all ViDoRe tasks. Detailed results in C.3.2.

Tradeoffs between model size and the number of image patches. We train a variant of PaliGemma with half the number of image patches (512). While we observe a clear performance degradation with respect to the 1024-patch ColPali model (-24.8 nDCG@5), memory usage is much lower. As an alternative to PaliGemma, we train Idefics2-8B [111], a VLM with a similar architecture based on a Mistral-7B [136] language backbone and a SigLIP vision encoder paired with a perceiver resampler. The most notable differences from PaliGemma lie in the size of the language model (2B and 7B, respectively) and the number of image patches (between 512 and 2048 for PaliGemma, and 64 post-resampling for Idefics2¹⁴). Our results

[111]: Laurençon et al. (2024), *What matters when building vision-language models?*

[136]: Jiang et al. (2023), *Mistral 7B*

14: With the option of adding 4 sub-image crops of 64 tokens each to the sequence, for a total of 320 tokens

suggest that better language models enable more efficient representations of image embeddings: ColIdetics2 with 64 patches largely outperforms ColPali with 512 patches (+20.1 nDCG@5). However, ColIdetics2 (64) remains less accurate than ColPali (1024) (−4.7 nDCG@5) while being about twice as slow in terms of training and inference latency. These results suggest there are *tradeoffs* between performance (R1), latencies during online querying (R2) and offline indexation phases (R3), and index memory size.

Unfreezing the vision component. We train a ColPali variant by also backpropagating through and updating the vision encoder and the projection layer. This leads to a slight performance degradation (−0.7 nDCG@5). These conclusions may change with larger scales of training data.

Impact of “query augmentation” tokens. In ColBERT, special tokens are concatenated to the input query to serve as soft query augmentation buffers. Training without these tokens, we observe no significant performance difference in the English benchmarks. However, performance on the French tasks seems to improve (+9.8 nDCG@G on Shift, +6.3 nDCG@5 on TabFQuAD, Table C.5).

Impact of the Pairwise CE loss. Training with an in-batch negative contrastive loss, instead of the pairwise CE loss that only considers the hardest negative sample, leads to a slight performance degradation (−1.6 nDCG@5) on the aggregated benchmark.

Adapting models to new tasks. Contrary to more complex multi-step retrieval pipelines, ColPali can be trained end-to-end, directly optimizing the downstream retrieval task which greatly facilitates fine-tuning to boost performance on specialized domains, multilingual retrieval, or specific visual elements the model struggles with. To demonstrate, we add 1552 samples representing French tables and associated queries to the training set. This represents the only French data in the training set, with all other examples being kept unchanged. We see clear nDCG@5 improvements (+2.6) and even starker Recall@1 gains (+5) on the TabFQuAD benchmark, with no performance degradation on the rest of the benchmark tasks (+0.4 nDCG@5 overall).

Better VLMs lead to better visual retrievers. As improved VLMs are released, it is interesting to observe whether improved performance on generative tasks translates once these models are adapted for image retrieval tasks through ColPali training strategies. We train the recently released Qwen2-VL 2B [137], a SOTA 2-billion-parameter generative VLM, with the same data and training strategy, obtaining ColQwen2-VL. To approximately match ColPali’s memory requirements, we limit the number of image patches to 768, slightly fewer than ColPali’s 1024 patches. We observe a clear improvement of +5.3 nDCG@5 over ColPali, showcasing a clear correlation between performance on generative benchmarks and retrieval metrics.

Out-of-domain generalization. Some of the datasets in the ViDoRe benchmark have train sets, which we have integrated within the ColPali train set (e.g. academic tasks). This is standard in embedding models [71, 73], and while ColPali also exhibits strong performance on tasks in which this is not the case (French data is never seen by the model

[137]: Wang et al. (2024), *Qwen2-VL: Enhancing Vision-Language Model’s Perception of the World at Any Resolution*

[71]: Wang et al. (2024), *Improving Text Embeddings with Large Language Models*

[73]: Lee et al. (2024), *NV-Embed: Improved Techniques for Training LLMs as Generalist Embedding Models*

[138]: Laurençon et al. (2024), *Building and better understanding vision-language models: insights and future directions*.

15: Training with data resembling what BGE-M3 models would see at inference time would require running complex extraction pipelines for the more than 100K documents in the training set, notably relying on external proprietary captioning models, which would be both too costly and time-consuming. This is not needed to train vision-based models.

[139]: Ma et al. (2024), *MMLongBench-Doc: Benchmarking Long-context Document Understanding with Visualizations*

[18]: Gisserot-Boukhlef et al. (2024), *Towards Trustworthy Reranking: A Simple yet Effective Abstention Mechanism*

[140]: Jiang et al. (2024), *VLM2Vec: Training Vision-Language Models for Massive Multimodal Embedding Tasks*

during training, for instance), it remains interesting to evaluate model performance when training is done on a fully disjoint data distribution. We train a ColPali variant solely using the recent DocMatix dataset [138], a large-scale, synthetically annotated visual document question-answering dataset, which we subsample to obtain a comparably sized train set. Results on ViDoRe show the performance drop is minor (-2.2 nDCG@5), still outperforming the closest baseline method by over 12 points. These results show that ColPali generalizes well outside its training distribution and demonstrate that our results are not unreasonably boosted with respect to baselines (BGE-M3) that cannot be fine-tuned on the same data¹⁵

3.7 Conclusions

In this work, we introduced the Visual Document Retrieval Benchmark (ViDoRe), which evaluates document retrieval systems in realistic settings involving visually complex documents. We demonstrated that current retrieval pipelines and contrastive vision-language models struggle to efficiently exploit visual information embedded in documents, leading to suboptimal performance. To address this, we presented *ColPali*, a novel retrieval method that leverages Vision-Language Models to create high-quality, multi-vector embeddings purely from visual document features. *ColPali* largely outperforms the best existing document retrieval methods while enabling faster corpus indexing times and maintaining low querying latencies, thus circumventing many pain points of modern document retrieval applications. We hope to drive industrial adoption, and to encourage future work by publicly releasing the ViDoRe benchmark, the data, the codebase, and all models and baselines from our work.

Future Work. Beyond performance improvements that could be obtained through better data, backbone models or training strategies, our vision at term is to combine visual retrieval systems and visually grounded query answering to create end-to-end RAG systems that purely function from image features. This idea is supported by concurrent work [139] showcasing the strong promises of VLMs for visual QA, and may eventually become a new industrial standard for document processing. In this line of work, reliability is key, and confidence estimation techniques for Information Retrieval methods could become central to implement abstention mechanisms [18], and are particularly interesting given the information rich multi-vector scoring mechanisms of late interaction systems. Expanding benchmarking efforts to cover more languages, modalities, and tasks is also a crucial future research direction [140].

Paper Context

Released in January 2026 and published at ACL 2026, this paper introduces the third iteration of the ViDoRe benchmark, extending the line of work initiated by our first two releases.

Since the original ViDoRe Benchmark [6], visual retrieval models have progressed rapidly. The first *ColPali* model reported an average score of 81.3 nDCG@5; today, leading models on the leaderboard now exceed 90 nDCG@5, and several tasks have become too easy to provide a meaningful signal. As the benchmark approached saturation, it became harder to measure real progress or diagnose model capabilities in realistic retrieval settings. This motivated ViDoRe V2: a benchmark designed not to reward incremental leaderboard gains, but to challenge the next generation of visual retrieval systems.

With ViDoRe V2 [7], our goal was to make evaluation closer to real-world search: difficult, diverse, and meaningful. Existing benchmarks often fail to capture how users actually query document collections [141]. We identified three recurring limitations:

1. **Extractive Nature of Queries:** Current benchmarks typically rely on extractive queries, providing unrealistic retrieval contexts since real users rarely formulate queries from exact phrases in documents.
2. **Single-Page Query Bias:** Many benchmarks overly emphasize retrieval from single-page contexts, neglecting complex, multi-document or cross-document queries common in real-world applications.
3. **Challenges in Synthetic Query Generation:** Purely synthetic benchmarks, while appealing in theory, are difficult to implement effectively without extensive manual oversight. They often produce outliers, irrelevant or trivial queries, making human filtering essential yet costly.

To address these issues, ViDoRe V2¹ introduced three key design choices:

- ▶ **Blind contextual querying:** Query-generation models were given only limited information about each document, such as summaries or metadata, rather than direct access to the full content. This reduced extractive bias and better approximated how users query unfamiliar corpora.
- ▶ **Long and cross-document queries:** The benchmark emphasizes realistic information needs that may require reasoning over long documents or retrieving across multiple documents.
- ▶ **Hybrid synthetic and human-in-the-loop creation:** Synthetic queries were extensively reviewed and refined by humans, producing a substantially cleaner and more reliable evaluation set.

4.1	Introduction	36
4.2	Related Work	38
4.3	Benchmark Creation . . .	39
4.3.1	Document Collection . .	39
4.3.2	Query Generation	39
4.3.3	Answer Detection and Generation	40
4.3.4	Final Query Distribution	41
4.3.5	Dataset Release and Distribution	42
4.4	Experiments and Results	43
4.4.1	Retrieval	43
4.4.2	Final Answer Generation	45
4.4.3	Visual Grounding	46
4.5	Conclusion	47

[141]: Thakur et al. (2025), *FreshStack: Building Realistic Benchmarks for Evaluating Retrieval on Technical Documents*

1: As reflected in the author lists, I am last author of ViDoRe V2 and co-author of ViDoRe V3. For thesis examiners, both papers should be understood as direct continuations of the work initiated in ViDoRe V1. After V1, I led a research group at Illuin Technology on Document Retrieval, defining the scientific direction and initiating the projects. Quentin Macé and Antonio Loison played central roles in the V2 and V3 releases and carried out most of the day-to-day work, while I remained closely involved at a higher level through regular discussions, supervision, and research guidance. Although this is not a typical first-author contribution, I consider these papers a clear part of my thesis work.

We released ViDoRe V2 in 2025. Its initial results showed substantial headroom for future models, while offering new insights into generalization and multilingual retrieval.

Shortly after the V2 release, we collaborated with NVIDIA to scale up the recipe with orders of magnitude more annotation resources and create ViDoRe V3. A major effort was devoted to constructing non-trivial queries that better reflect real search-engine usage. This was enabled by an improved annotation pipeline combining 12,000 hours of expert human annotation with a synthetic generation process. The resulting benchmark is the largest of its kind and has been well received by the community. Together, V2 and V3 have become standard evaluation datasets for newly released Visual Document Retrieval models.

More broadly, the three ViDoRe releases illustrate a central lesson of this thesis: progress in retrieval depends on rigorous evaluation. ViDoRe has provided the field with a shared canvas for measuring progress in Visual Document Retrieval. When V3 eventually saturates, it will be because the field has moved substantially forward.

Paper Abstract

Retrieval-Augmented Generation (RAG) pipelines must address challenges beyond simple single-document retrieval, such as interpreting visual elements (tables, charts, images), synthesizing information across documents, and providing accurate source grounding. Existing benchmarks fail to capture this complexity, often focusing on textual data, single-document comprehension, or evaluating retrieval and generation in isolation. We introduce **ViDoRe V3**, a comprehensive multimodal RAG benchmark featuring multi-type queries over visually rich document corpora. It covers 10 datasets across diverse professional domains, comprising 26,000 document pages paired with 3,099 human-verified queries, each available in 6 languages. Through 12,000 hours of human annotation effort, we provide high-quality annotations for retrieval relevance, bounding box localization, and verified reference answers. Our evaluation of state-of-the-art RAG pipelines reveals that visual retrievers outperform textual ones, late-interaction models and textual reranking substantially improve performance, and hybrid or purely visual contexts enhance answer generation quality. However, current models still struggle with non-textual elements, open-ended queries, and fine-grained visual grounding. To encourage progress in addressing these challenges, the benchmark is released under a commercially permissive license.

4.1 Introduction

Retrieval-Augmented Generation (RAG) [87] has become the dominant paradigm for knowledge-intensive NLP tasks [142, 143]. Yet practical deployments introduce complexities that academic benchmarks often overlook when focusing on single-document textual retrieval. First, documents encode critical information in visual elements such as tables,

[87]: Lewis et al. (2020), *Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks*

[142]: Gao et al. (2023), *Retrieval-Augmented Generation for Large Language Models: A Survey*

[143]: Fan et al. (2024), *A Survey on RAG Meeting LLMs: Towards Retrieval-Augmented Large Language Models*

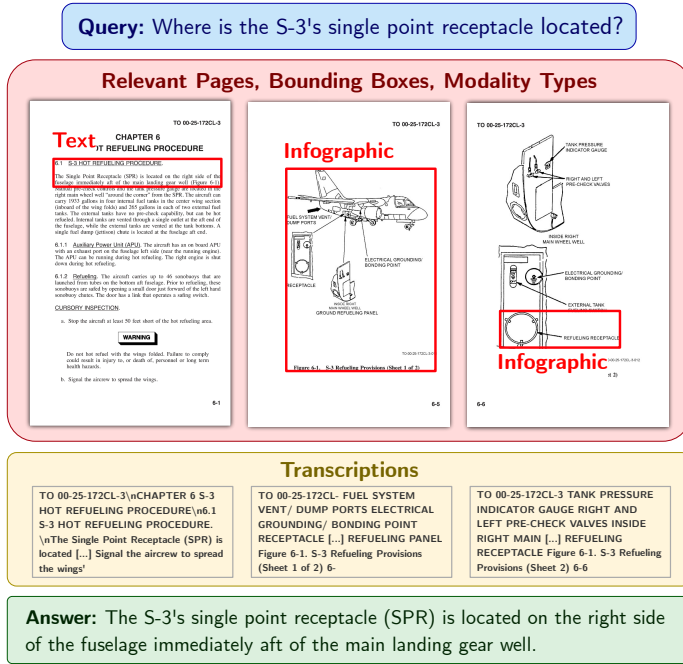


Figure 4.1: ViDoRe V3 annotated sample. Each query is paired with relevant pages, a document-grounded reference answer, evidence bounding boxes, and modality labels for each box. Documents are provided in image, text, and PDF formats.

charts, and images designed for human interpretation, which text-only pipelines often ignore [144, 145]. Second, user queries often require open-ended synthesis, comparison, and reasoning over scattered information, not simple factoid lookup [10, 141, 146]. Third, trustworthy systems must ground responses to specific source locations (e.g., bounding boxes), to mitigate hallucinations [147, 148].

Existing benchmarks leave these requirements only partially addressed. Early Visual Document Understanding (VDU) benchmarks focus on single-page comprehension, ignoring the complexity of large document corpora [120]. Recent retrieval-centric benchmarks do not evaluate generation quality and grounding [6, 149]. Some multimodal datasets attempt to bridge this gap but rely on extractive, short-answer tasks that fail to exercise complex reasoning [145], or lack multilingual diversity and fine-grained visual grounding [150].

To address these limitations, we introduce **ViDoRe V3**, a benchmark designed for complex and realistic end-to-end RAG evaluation on visually rich document corpora. Our contributions are:

1. A Human Annotation Methodology for Realistic Queries We propose an annotation protocol for generating diverse queries and fine-grained query-page annotations. By restricting annotator access to document content during query formulation, we capture authentic search behaviors and mitigate bias toward simple extractive queries. Vision-Language Model (VLM) filtering combined with human expert verification enables efficient, high-quality annotation at scale.

2. The ViDoRe V3 Benchmark Applying this methodology to 10 industry-relevant document corpora, we build ViDoRe V3, a multilingual RAG benchmark comprising 26,000 pages and 3,099 queries, each

[144]: Abootorabi et al. (2025), *Ask in Any Modality: A Comprehensive Survey on Multimodal Retrieval-Augmented Generation*

[145]: Cho et al. (2024), *M3DocRAG: Multi-modal Retrieval is What You Need for Multi-page Multi-document Understanding*

[10]: Conti*, Faysse* et al. (2025), *Context is Gold to find the Gold Passage: Evaluating and Training Contextual Document Embeddings*

[141]: Thakur et al. (2025), *FreshStack: Building Realistic Benchmarks for Evaluating Retrieval on Technical Documents*

[146]: Tang et al. (2024), *MultiHop-RAG: Benchmarking Retrieval-Augmented Generation for Multi-Hop Queries*

[147]: Gao et al. (2023), *Enabling Large Language Models to Generate Text with Citations*

[148]: Ma et al. (2024), *VISA: Retrieval Augmented Generation with Visual Source Attribution*

[120]: Mathew et al. (2020), *DocVQA: A Dataset for VQA on Document Images*

[6]: Faysse et al. (2025), *ColPali: Efficient Document Retrieval with Vision Language Models*

[149]: Günther et al. (2025), *jina-embeddings-v4: Universal Embeddings for Multimodal Multilingual Retrieval*

[145]: Cho et al. (2024), *M3DocRAG: Multi-modal Retrieval is What You Need for Multi-page Multi-document Understanding*

[150]: Peng et al. (2025), *UNIDOC-BENCH: A Unified Benchmark for Document-Centric Multimodal RAG*

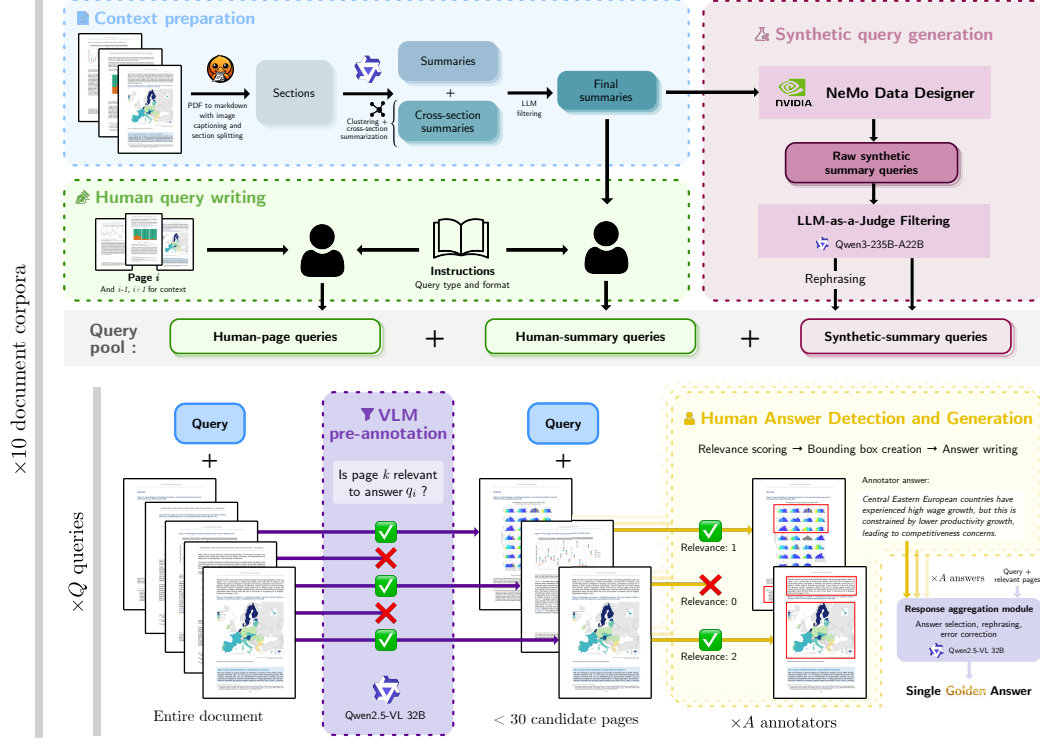


Figure 4.2: Overview of the ViDoRe V3 data creation workflow. Queries are collected from three streams: *human extractive* (raw pages), *human blind contextual* (summaries to reduce extractive bias), and *synthetic blind contextual*. For each query, a VLM pre-filters candidate pages, then 1–3 human annotators perform relevance scoring, bounding-box localization, and answer writing; final responses are aggregated into a single reference answer.

available in 6 languages. Two datasets are held out as a private test set to mitigate overfitting. The public datasets are released under a commercially permissive license.

[120]: Mathew et al. (2020), *DocVQA: A Dataset for VQA on Document Images*

[121]: Mathew et al. (2021), *InfographicVQA*

[123]: Zhu et al. (2022), *Towards Complex Document Understanding By Discrete Reasoning*

[151]: Wang et al. (2024), *Charxiv: Charting gaps in realistic chart understanding in multimodal llms*

[152]: Yu et al. (2025), *BBox DocVQA: A Large Scale Bounding Box Grounded Dataset for Enhancing Reasoning in Document Visual Question Answer*

[6]: Faysse et al. (2025), *ColPali: Efficient Document Retrieval with Vision Language Models*

[153]: Ma et al. (2024), *Unifying Multimodal Retrieval via Document Screenshot Embedding*

[154]: Yu et al. (2024), *VisRAG: Vision-based Retrieval-augmented Generation on Multi-modality Documents*

[155]: Xu et al. (2025), *Llama NemoRetriever Colombed: Top-Performing Text-Image Retrieval Model*

[149]: Günther et al. (2025), *jina-embeddings-v4: Universal Embeddings for Multimodal Multilingual Retrieval*

3. Comprehensive Evaluation and Insights Leveraging our granular annotations, we benchmark state-of-the-art models on (i) retrieval accuracy by modality and language, (ii) answer quality across diverse retrieval pipeline configurations, and (iii) visual grounding fidelity. Our analysis surfaces actionable findings for RAG practitioners.

4.2 Related Work

Component-Level Benchmarks (VDU and Retrieval) VDU has traditionally relied on single-page datasets like DocVQA [120], alongside domain-specialized variants [121, 123, 151]. These ignore the multi-page context inherent to RAG. Recent work evaluating bounding-box source grounding [152] proposes single-page and multi-page tasks but does not address the retrieval component. Conversely, the emergence of late-interaction visual retrievers [6, 153–155] spurred the creation of retrieval-centric visual benchmarks like Jina-VDR [149] and ViDoRe V1&V2 [6, 7], but none of these benchmarks jointly evaluate retrieval and answer generation.

End-to-End Multimodal RAG While recent textual RAG benchmarks now capture complex user needs like reasoning or summarizing [141, 146, 156], multimodal evaluation often remains limited to single page queries [6]. Multi-page datasets like DUDE [157], M3DocRAG [145], ViDoSeek [158] or Real-MM-RAG [159] prioritize extractive retrieval, lacking the diversity of queries encountered in realistic settings. UniDocBench [150] represents a concurrent effort that similarly addresses diverse query types and provides comparative evaluation across multiple RAG paradigms. While this benchmark offers valuable contributions, it relies on synthetically generated queries via knowledge-graph traversal, is restricted to English documents, and constrains grounding annotations to parsed document elements. In contrast, our benchmark offers several complementary strengths: fully human-verified annotations, multilingual coverage, free-form bounding box annotations, and a more systematic evaluation of individual visual RAG pipeline components.

4.3 Benchmark Creation

We design the benchmark to mirror the diversity of information retrieval situations in large-scale realistic environments. To enable pipeline-agnostic evaluation of the 3 core RAG components (retrieval, generation and grounding), while avoiding limitations of synthetic benchmarks, we employ a rigorous three-stage human-in-the-loop annotation process involving document collection, query generation and grounded query answering (Figure 4.2).

4.3.1 Document Collection

We curate 10 diverse corpora by manually selecting openly-licensed documents from governmental, educational, and industry sources, focusing on English and French documents (7 and 3 corpora respectively). The corpora span Finance, Computer Science, Energy, Pharmaceuticals, Human Resources, Industrial Maintenance, Telecom, and Physics. Each features domain-specific terminology and document structures representative of realistic retrieval tasks (details in Table D.2).

4.3.2 Query Generation

Query Taxonomy To evaluate document visual retrieval systems across diverse realistic scenarios, we develop a query taxonomy with two orthogonal dimensions: *Query Type*, defining the user’s information need, and *Query Format*, describing the query’s syntactic structure. This dual-axis classification enables more nuanced performance analysis than benchmarks focusing solely on interrogative extractive queries. We define 7 Query Types: *open-ended*, *extractive*, *numerical*, *multi-hop*, *compare-contrast*, *boolean*, and *enumerative*, and 3 Query Formats: *question*, *keyword*, and *instruction*.

[6]: Faysse et al. (2025), *ColPali: Efficient Document Retrieval with Vision Language Models*

[7]: Macé et al. (2026), *ViDoRe Benchmark V2: Raising the Bar for Visual Retrieval*

[141]: Thakur et al. (2025), *FreshStack: Building Realistic Benchmarks for Evaluating Retrieval on Technical Documents*

[146]: Tang et al. (2024), *MultiHop-RAG: Benchmarking Retrieval-Augmented Generation for Multi-Hop Queries*

[156]: Su et al. (2024), *Bright: A realistic and challenging benchmark for reasoning-intensive retrieval*

[6]: Faysse et al. (2025), *ColPali: Efficient Document Retrieval with Vision Language Models*

[157]: Van Landeghem et al. (2023), *Document understanding dataset and evaluation (dude)*

[145]: Cho et al. (2024), *M3DocRAG: Multi-modal Retrieval is What You Need for Multi-page Multi-document Understanding*

[158]: Wang et al. (2025), *Vidorag: Visual document retrieval-augmented generation via dynamic iterative reasoning agents*

[159]: Wasserman et al. (2025), *REAL-MM-RAG: A Real-World Multi-Modal Retrieval Benchmark*

[150]: Peng et al. (2025), *UNIDOC-BENCH: A Unified Benchmark for Document-Centric Multimodal RAG*

[7]: Macé et al. (2026), *ViDoRe Benchmark V2: Raising the Bar for Visual Retrieval*

[160]: Deep Search Team (2024), *Docling Technical Report*

[161]: Qwen Team (2025), *Qwen3 Technical Report*

[162]: Zhang et al. (2025), *Qwen3 Embedding: Advancing Text Embedding and Reranking Through Foundation Models*

[163]: McInnes et al. (2020), *UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction*

[164]: Campello et al. (2013), *Density-based clustering based on hierarchical density estimates*

[165]: NeMo Data Designer Team (2025), *NeMo Data Designer: A framework for generating synthetic data from scratch or based on your own seed data*

[166]: Bai et al. (2025), *Qwen2.5-VL Technical Report*

Context Preparation We further ensure query diversity by pulling summaries from a heterogeneous set of contexts during the generation process. Two types of input contexts are used: specific document sections that target local information retrieval and cross-section summaries that target multi-document context retrieval. These summaries are produced through a refined process inspired by ViDoRe V2 [7]. First, the text is extracted from PDFs using Docling [160] along with image descriptions. Then, summaries are generated with Qwen3-235B-Instruct [161] from each document section. They are clustered to group similar summaries together using Qwen3-Embedding-0.6B [162] as embedder, UMAP [163] for dimension reduction and HDBSCAN [164] for clustering. Additionally, cross-section summaries are produced by synthesizing the summaries of 2 to 3 randomly selected sections per cluster. From this pool of summaries, a final subset is curated to maintain a strict balance between single-section and cross-section summaries. The selection also ensures an even distribution across section modalities (text, images, and tables) as defined by the Docling element classification.

Synthetic Query Generation Queries are generated from the summaries using a first synthetic generation pipeline based on Qwen3-235B. For each summary, a prompt is constructed by sampling a query type and format at random, together with variable attributes such as length and difficulty, in order to promote diversity. The generated queries are subsequently evaluated by the same LLM acting as an automatic judge, which filters outputs according to 4 criteria: information richness, domain relevance, clarity and adherence to query type/format. Finally, 50% of the retained queries are rephrased to further enhance linguistic variance. This pipeline is implemented using NeMo Data Designer [165] to facilitate generation scaling.

Human Query Writing Human annotators are provided 2 kinds of contexts: synthetic summaries or specific PDF pages. They are tasked with generating one query following a specific query type and format and one query of their choice that is most adapted to the context provided.

4.3.3 Answer Detection and Generation

Queries are filtered and linked to relevant pages using a hybrid pipeline of VLM pre-filtering and human annotation. It is followed by human answer annotation and visual grounding.

Query-Page Linking Given the scale of our corpora, manual verification of each page relevance for each query is intractable. We therefore adopt a two-stage annotation pipeline. First, Qwen2.5-VL-32B-Instruct [166] pre-filters candidate pages by assessing whether each page image is relevant to the query. Queries whose answers span more than 30 flagged pages are discarded. Human annotators then review the remaining query-page pairs, evaluating query quality and rating page relevance on a three-point scale (Not Relevant, Critically Relevant, Fully Relevant).

Relevant Page Selection To ensure annotation quality, each task is completed by multiple annotators and reviewed by annotation supervisors. Since VLM pre-filtering biases the distribution toward relevant pages, we report Gwet’s AC2, as it remains stable under prevalence skew, at 0.760 (see Appendix D.4 for dataset-level breakdowns). Given this strong but imperfect agreement, we implement a tiered review process: extractive queries require at least one annotator and one reviewer, while more complex non-extractive queries require at least two annotators and one reviewer. A page is retained as relevant if marked by either (i) one annotator and one reviewer, or (ii) at least two annotators.

Answer Generation For each selected query, annotators were tasked with writing an answer based on the pages they marked as relevant. Given that different annotators might have different answer interpretations and tend not to be exhaustive in their answers, we use Qwen2.5-VL-32B-Instruct to generate a final answer based on the relevant page images marked by the annotators and their answers.

Bounding Boxes and Modality Types For each relevant page, annotators delineate bounding boxes around content supporting the query and attribute a modality type to each bounding box: Text, Table, Chart, Infographic, Image, Mixed or Other. Because multiple valid interpretations of bounding boxes can exist, we perform a consistency study to evaluate inter-annotator agreement and establish a human performance upper bound for the task.

We compute inter-annotator agreement on the subset of query-page pairs labeled by two or three annotators. For each annotator, we merge all their bounding boxes into a single zone. We then compare zones across annotators by measuring pixel-level overlap, reporting Intersection over Union (IoU) and F1 score (Dice coefficient). When three annotators label the same sample, we average over all pairwise comparisons.

Across all 10 datasets, we observe an average IoU of 0.50 and F1 of 0.60. These moderate agreement scores reflect the inherent subjectivity of the task: annotators typically agreed on the relevant content but differed in granularity (Appendix D.9), with some marking tight bounds around specific content while others included surrounding context.

Quality Control The annotation was conducted by a curated pool of 76 domain-qualified experts with native-level language proficiency. Quality control was performed by 13 senior annotators with enhanced domain knowledge and extensive annotation experience. Detailed protocols regarding the annotator pool and training are provided in Appendix D.3.

4.3.4 Final Query Distribution

We conducted a final human review to remove low-quality queries and resolve labeling ambiguities. Figure 4.3 shows the resulting distribution. Extractive queries predominate due to human annotator preference, followed by open-ended queries from targeted sampling. Multi-hop queries

were the hardest to scale, suggesting a need for dedicated pipelines. Figure 4.4 details page modalities; while text is most prevalent, visual elements like tables, charts, and infographics are well-represented.

Figure 4.3: Query type distribution by domain. Relative frequencies of query types are reported for each benchmark domain.

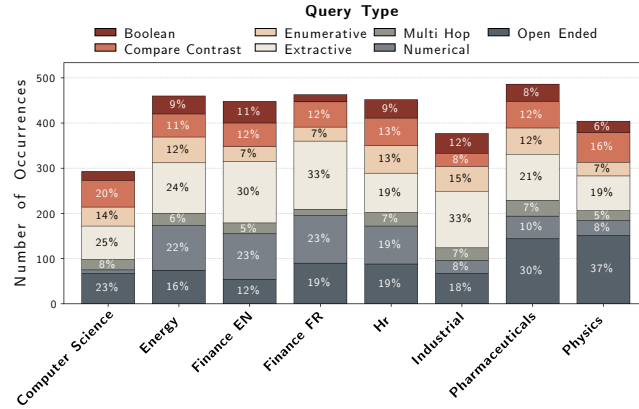
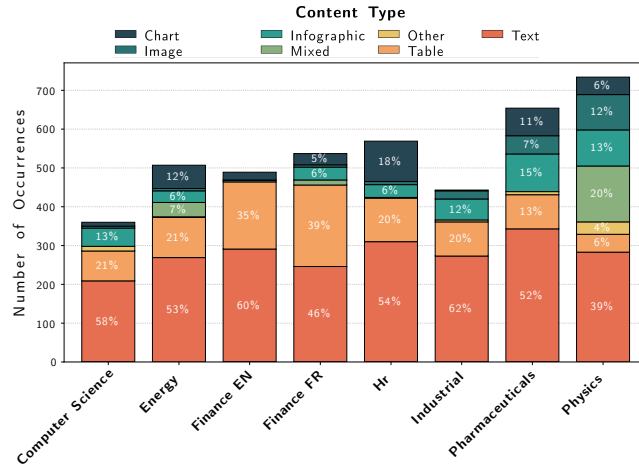


Figure 4.4: Content type distribution by domain. Relative frequencies of evidence content types are reported for each benchmark domain.



4.3.5 Dataset Release and Distribution

We extend the benchmark to rigorously assess cross-lingual retrieval. While source documents are maintained in English and French, we use Qwen3-235B-Instruct to provide translations in 6 languages: English, French, Spanish, German, Italian, and Portuguese. This configuration challenges models to bridge the semantic gap between the query language and the document language, a critical requirement for modern RAG systems.

Finally, to ensure the integrity of evaluation and mitigate data contamination (which was shown to be a major preoccupation for Information Retrieval [167]), we adopt a split-release strategy. 8 datasets are made public to facilitate research, while 2 are retained as private hold-out sets. This enables blind evaluation, ensuring that performance metrics reflect true generalization rather than overfitting to public samples.

[167]: Liu et al. (2025), *Introducing RTEB: A New Standard for Retrieval Evaluation*

4.4 Experiments and Results

Using our benchmark, we conduct extensive evaluations across all 3 components of RAG pipelines. We assess textual and visual retrievers and rerankers on retrieval performance, evaluate leading VLMs and LLMs on their ability to generate accurate answers from various retrieved contexts, and test VLMs on bounding box generation for visual grounding. From these results, we compile practical insights for RAG practitioners.

Table 4.1: Retrieval performance (NDCG@10) across ViDoRe V3. Best results per category are in bold; ★ marks single-vector models. Following MTEB conventions, the average is a macro-average across datasets. Full model names and references are provided in Table D.4.

Model	Size (B)	English Datasets							French Datasets			Avg.
		C.S.	Nucl.	Fin.	Phar.	H.R.	Ind.	Tel.	Phys.	Ener.	Fin.	
Textual Retrievers												
Qwen3-8B★	8	71.7	39.0	49.4	59.2	47.6	40.4	62.8	45.6	58.9	35.8	51.0
Jina-v4	3	64.3	44.3	48.4	54.9	52.8	38.4	56.3	43.6	60.1	41.3	50.4
LFM2-350M	0.35	63.5	37.8	39.0	56.4	43.5	34.4	56.9	41.8	47.0	28.2	44.9
Qwen3-0.6B★	0.6	66.4	32.8	42.7	50.6	37.7	31.6	55.7	43.3	51.3	25.8	43.8
BGE-M3★	0.57	58.0	30.2	39.8	52.0	42.4	28.5	51.6	35.9	49.8	25.2	41.3
BM25S	-	28.7	17.4	17.6	27.3	12.8	15.6	33.3	14.8	21.9	14.0	20.3
Visual Retrievers												
ColEmbed-3B-v2	3	77.1	50.7	64.2	66.0	62.3	51.7	69.7	47.0	64.9	44.4	59.8
Jina-v4	3	71.8	50.0	59.3	63.1	59.5	50.4	64.8	46.6	64.0	46.1	57.6
ColNomic-7B	7	76.2	45.0	56.6	62.3	58.7	50.1	67.2	48.3	64.0	45.5	57.4
ColEmbed-3B	3	75.2	49.1	60.9	63.7	58.7	47.1	67.0	45.1	62.1	43.8	57.3
ColNomic-3B	3	72.7	42.1	56.3	61.1	57.3	47.4	64.5	47.5	65.0	44.3	55.8
ColEmbed-1B	1	71.3	47.3	58.9	62.6	57.0	46.6	64.7	44.1	60.9	42.4	55.6
ColQwen2.5	3	72.3	38.1	52.3	57.9	51.2	41.3	61.3	45.9	59.7	39.1	51.9
Nomic-7B★	7	66.6	36.7	48.8	58.9	46.2	37.9	57.8	44.2	57.5	36.0	49.0
ColQwen2	2	68.6	35.7	39.0	52.2	45.1	38.3	57.4	41.6	48.8	20.0	44.7
Nomic-3B★	3	58.5	32.2	44.2	55.3	43.3	33.2	53.7	42.0	51.4	28.9	44.3
ColPali	7	65.3	32.9	34.4	53.1	44.8	35.6	54.0	41.7	47.1	21.8	43.1

4.4.1 Retrieval

We evaluate a large panel of visual and textual retrievers on page-level retrieval ability. Visual retrievers are given page images, while textual retrievers process the Markdown text of each page processed by the NeMo Retriever extraction service² [168]. The results reported in Table 4.1 corroborate findings from existing document retrieval benchmarks [6, 149]: for a given parameter count, visual retrievers outperform textual retrievers, and late interaction methods score higher than dense methods.

We analyze ColEmbed-3B-v2, the best-performing retriever we evaluated across query type, content modality, and query language.

Performance is aligned with query complexity Figure 4.5 shows that performance is inversely correlated with query complexity: simple query types such as Boolean and Numerical score significantly higher than Open-ended and Multi-hop queries. Question formulations consistently outperform Instruction and Keyword formats across nearly all categories, underscoring the need for improved handling of these query structures.

Visual Content and multi-page queries are hardest for retrievers Figure 4.6 highlights that queries involving visual content like tables or images tend to be more difficult. The Mixed content type scores the lowest,

2: Chunking within pages or providing image descriptions did not improve our results. Thus, we report the results of the simplest pipeline.

[168]: NVIDIA Ingest Development Team (2024), *NVIDIA Ingest: An accelerated pipeline for document ingestion*

[6]: Faysse et al. (2025), *ColPali: Efficient Document Retrieval with Vision Language Models*

[149]: Günther et al. (2025), *jina-embeddings-v4: Universal Embeddings for Multimodal Multilingual Retrieval*

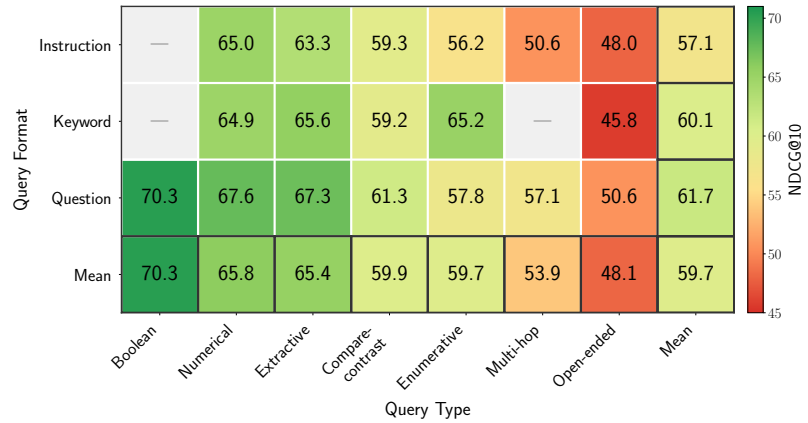


Figure 4.5: ColEmbed-3B-v2 NDCG@10 by query type and format. Performance is broken down jointly by semantic query type and surface query format.

which suggests that integrating information across different modalities within a single page remains a challenge. Additionally, we observe a consistent decline in performance as the number of annotated pages increases (Figure 4.7), suggesting that retriever effectiveness decreases when aggregating information from multiple sources is required.

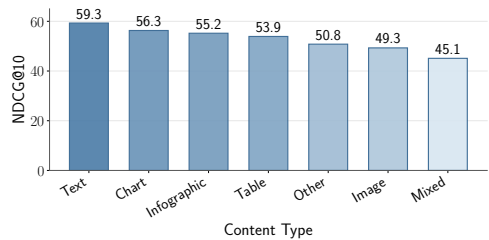


Figure 4.6: ColEmbed-3B-v2 NDCG@10 by content modality. Scores are reported for text, table, image, and mixed evidence categories.

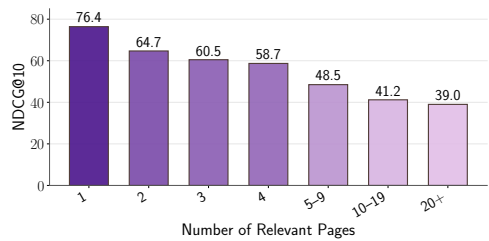


Figure 4.7: ColEmbed-3B-v2 NDCG@10 by number of annotated relevant pages. Retrieval performance is shown as relevance is distributed across more pages.

Cross-language queries degrade performance Retrieval performance is 2–3 points higher in mono-lingual settings (Table D.5 and Table D.6) than cross-lingual settings (Table 4.1), showing that models need to better adapt to these settings.

Textual rerankers outperform visual ones We evaluate the impact of adding a reranker to the textual and visual pipelines of the Jina-v4 retriever. We select zerank-2 [169] and jina-reranker-m0 [170] as two of the leading textual and visual rerankers to date. Results in Table 4.2 reveal a significant disparity in reranking efficacy between modalities. While the visual retriever initially outperforms the textual base, the textual reranker yields substantial gains (+13.2 NDCG@10), enabling the textual pipeline to achieve the highest overall retrieval performance. In contrast, the visual reranker provides only a marginal average improvement of +0.2 and degrades performance in 4 datasets, underscoring the need for better multilingual visual rerankers.

[169]: Zero Entropy (2025), *Introducing zerank-2*

[170]: Jina AI (2025), *jina-reranker-m0: Multilingual Multimodal Document Reranker*

Table 4.2: Retrieval performance (NDCG@10) of retriever + reranker pipelines. The table compares textual and visual reranking setups across datasets.

Model	English Datasets							French Datasets			Avg.
	C.S.	Nucl.	Fin.	Phar.	H.R.	Ind.	Tel.	Phys.	Ener.	Fin.	
<i>Textual pipeline</i>											
Jina-v4 ^{textual}	64.3	44.3	48.4	54.9	52.8	38.4	56.3	43.6	60.1	41.3	50.4
+ zerank-2	82.1 ↑17.8	53.5 ↑9.2	69.2 ↑20.8	66.2 ↑11.3	66.5 ↑13.7	53.2 ↑14.8	71.5 ↑15.2	48.2 ↑4.6	71.5 ↑11.4	53.7 ↑12.4	63.6 ↑13.2
<i>Visual pipeline</i>											
Jina-v4 ^{visual}	71.8	50.0	59.3	63.1	59.5	50.4	64.8	46.6	64.0	46.1	57.6
+ jina-reranker-m0	76.7 ↑4.9	50.8 ↑0.8	59.2 ↓0.1	65.4 ↑2.3	56.0 ↓3.5	50.9 ↑0.5	70.8 ↑6.0	46.9 ↑0.3	61.7 ↓2.3	39.8 ↓6.3	57.8 ↑0.2

Table 4.3: End-to-end evaluation of final answer generation. We report the percentage of correct answers, as judged by an LLM, across the eight public datasets. "Oracle" rows denote the upper bound using gold-standard contexts. Average Easy and Average Hard are stratified by query difficulty. For each column, the best result is **bolded** and the best non-oracle result is underlined.

Retrieval pipeline	Context modality	Generation model	English Datasets					French Datasets			Avg. Hard	Avg. Easy	Avg. Global
			C.S.	Fin.	Phar.	H.R.	Ind.	Phys.	Ener.	Fin.			
Oracle	Text	Gemini 3 Pro	80.9	70.2	71.4	72.3	66.4	71.2	69.2	62.8	62.3	79.3	70.6
	Image		86.5	70.6	76.1	71.1	68.2	74.5	69.8	64.1	64.7	79.7	72.6
	Hybrid		86.0	68.9	73.4	70.4	65.4	69.2	69.5	62.8	63.4	77.5	70.7
Jina-v4 ^{text.} + zerank-2	Text	Gemini 3 Pro	80.9	66.0	59.9	63.2	60.4	69.2	64.9	<u>54.7</u>	52.1	75.5	64.9
Jina-v4 ^{text.} + zerank-2 & ColEmbed-3B-v2	Hybrid	Gemini 3 Pro	85.1	65.0	65.9	64.8	59.4	69.9	62.7	52.8	<u>54.7</u>	76.6	65.7
ColEmbed-3B-v2	Text	Gemini 3 Pro	82.3	62.5	61.0	62.9	56.2	64.9	62.3	49.4	51.7	73.0	62.7
		Kimi K2	81.4	56.6	59.1	55.7	55.8	73.8	60.4	43.1	44.6	74.3	60.7
	Image	Gemini 3 Pro	83.3	<u>67.3</u>	62.9	65.4	57.2	67.9	64.3	47.8	54.5	74.1	64.5
		Gemini 3 Flash	80.9	64.1	63.5	63.8	55.1	68.2	63.3	47.8	50.3	74.4	63.3
		GPT-5.2	86.5	59.5	<u>68.1</u>	<u>66.0</u>	<u>61.5</u>	76.5	<u>66.2</u>	49.1	54.1	<u>78.1</u>	<u>66.7</u>
		Qwen3-VL-235B	86.0	59.9	64.0	60.7	57.2	71.9	59.7	44.4	51.0	74.1	63.0

4.4.2 Final Answer Generation

We evaluate end-to-end answer quality by providing LLMs and VLMs with queries and their corresponding retrieved pages, examining the effects of retrieval pipeline selection, context modality, and generation model choice (Table 4.3). For this evaluation, we use the best-performing textual and visual retrieval pipelines. We additionally establish an upper bound using an *oracle* pipeline that supplies the model with ground-truth annotated pages.

In the *hybrid* configuration, we concatenate the top-5 results from the visual retriever (images) with the top-5 results from the textual retriever (text), without removing duplicates; the retrieval performance is detailed in Table D.7. We also consider a hybrid *oracle* setup, which provides the model with all the ground-truth pages in both modalities.

The correctness of generated answers is assessed against the ground truth final answer by an LLM judge (details in Appendix D.8). Private datasets are omitted to maintain their integrity.

Some benchmark queries involve general knowledge manageable by LLMs without retrieval. To prevent memorization from confounding our assessment of the RAG pipeline, we stratify queries by difficulty based on parametric knowledge. A query is categorized as *easy* if any model in a 6-LLM panel answers it correctly without context; otherwise, it is labeled *hard*. Overall, 48.6 % of queries are easy (see Table D.12 for details).

Visual context helps generation With a fixed Gemini 3 Pro generator, image-based context outperforms text-based context on the hard subset by 2.4 and 2.8 percentage points for the oracle and ColEmbed-3B-v2 pipelines, respectively (Table 4.3). This confirms that preserving the visual content of document pages provides better grounding for complex answer generation.

Hybrid retrieval yields the best performance on challenging queries

The hybrid pipeline achieves 54.7 % accuracy on hard queries, surpassing both the strongest textual (52.1 %) and visual (54.5 %) baselines. This complementary effect suggests that text and image representations capture different aspects of document content, and their combination can provide more robust evidence for downstream generation.

Hard queries expose the limits of parametric knowledge in current models

Even with oracle context, performance on hard queries lags behind easy queries by more than 10 percentage points. This gap suggests that the multi-step reasoning and long-context synthesis required for difficult queries remain challenging for current models. While the models we evaluate achieve comparable overall scores, their relative ranking may shift when parametric knowledge is less of an advantage, as shown by GPT 5.2 outperforming Gemini 3 Pro on easy queries but trailing on hard ones.

ViDoRe V3 leaves significant room for future retriever improvements

The 10-point gap between the best non-oracle result (54.7 %) and the image oracle (64.7 %) on hard queries underscores substantial opportunities for improving the retrieval pipeline. Moreover, even with oracle contexts, Gemini 3 Pro performance remains modest, indicating that generation models still struggle to fully exploit the provided information.

4.4.3 Visual Grounding

Beyond generating correct answers, it is highly desirable for RAG pipelines to identify where in the source documents the answer originates, enabling users to verify the grounding of the query answer. We therefore evaluate the ability of LLMs to generate accurate bounding boxes within their final answer. Among the few LLM families with visual grounding capabilities, we select Qwen3-VL-30B-A3B-Instruct and Gemini 3 Pro for evaluation. For each query, we provide the model with the candidate pages shown to the human annotators and prompt it to answer the query while inserting inline bounding boxes in XML format `<bboxes image="N"> ... </bboxes>` to delimit relevant content (full instructions in Appendix D.7).

We use the bounding boxes produced by the human annotators as our ground truth. Since each query may have 1–3 human annotators, we evaluate VLM predictions independently against each annotator using the same zone-based methodology as the inter-annotator consistency analysis (Section 4.3.3), and report the highest F1 score. This best-match strategy reflects the inherent subjectivity of evidence selection: annotators may

legitimately highlight different regions to support the same answer, and a model should not be penalized for matching any valid interpretation.

Visual grounding lags human performance Inter-annotator agreement on evidence localization reaches an F1 of 0.602, whereas the best-performing models achieve markedly lower scores: 0.089 for Qwen3-VL-30B-A3B-Instruct and 0.065 for Gemini 3 Pro. A page-level analysis (Table 4.4) reveals that on pages where humans provided bounding boxes, both models annotated the same page only 16–17 % of the time, while 26–27 % of human-annotated pages received no model annotation at all—highlighting recall as the primary bottleneck. Detailed per-domain results and qualitative analysis appear in Appendix D.7 and D.9.

Category	Outcome	Qwen3-VL-30B-A3B	Gemini 3 Pro
Agreement	Both annotated	17 %	16 %
	Neither annotated	46 %	49 %
Discrepancy	Model only	10 %	7 %
	Human only	26 %	27 %

Table 4.4: Page-level bounding box agreement between models and human annotators. Each page is categorized as both annotated, both unannotated, model-only, or human-only.

4.5 Conclusion

This work introduces ViDoRe V3, a multilingual, human-annotated RAG benchmark that evaluates retrieval, final answer generation, and visual grounding on large industry-relevant document corpora. We design a human-in-the-loop annotation methodology, deployed in a 12,000-hour annotation campaign, that produces diverse realistic queries paired with relevant pages, bounding boxes, and reference answers. Evaluating state-of-the-art RAG pipelines, we find that visual retrievers outperform textual ones, late interaction and textual reranking yield substantial gains, and visual context improves answer generation quality. Looking ahead, ViDoRe V3 highlights several concrete research directions for practical multimodal RAG. Retriever models still struggle on cross-lingual and open-ended queries requiring visual interpretation, while VLMs need improvement in answer generation from multi-page contexts as well as accurate visual grounding. By providing a rigorous framework for evaluating these limitations, ViDoRe V3 serves as a catalyst for the development of more robust, intelligent document understanding models.

ModernVBERT: Towards Smaller Visual Document Retrievers

5

Paper Context

Following the release of *ColPali*, Visual Document Retrieval has rapidly evolved, with academic and industrial teams worldwide training increasingly capable models—including our own contributions. Much of this progress has come from scaling to stronger VLM backbones, a trend we demonstrated with the *ColQwen* series, where better vision-language models translated directly into stronger retrieval performance.

In this work, released in September 2025 and published at ICML 2026, we take a different path. With *ModernVBERT*, we match ColPali-level retrieval performance using a model that is **10× smaller**. Beyond releasing a compact and practical resource, this paper offers a systematic study of what truly drives performance when repurposing VLMs for visual embeddings. Our ablations cover architecture design, data mixtures, cross-modal positive transfer, model souping, and more.

Most importantly, we show that strong visual retrieval does not require heavy query-time resources. *ModernVBERT* makes high-quality VDR cheaper to deploy, easier to scale, and more accessible for widespread industrial use.

Paper Abstract

Retrieving specific information from a large corpus of documents is a prevalent industrial use case of modern AI, notably due to the popularity of Retrieval-Augmented Generation (RAG) systems. Although neural document retrieval models have historically operated exclusively in the text space, Visual Document Retrieval (VDR) models - large vision-language decoders repurposed as embedding models which directly work with page screenshots as inputs - are increasingly popular due to the performance and indexing latency gains they offer. In this work, we show that, while cost-efficient, this approach of repurposing generative models bottlenecks retrieval performance. Through controlled experiments, we revisit the entire training pipeline, and establish a principled recipe for improving visual document retrieval models. We notably measure the impact of attention masking, image resolution, modality alignment data regimes, and late interaction centered contrastive objectives which emerge as central performance factors. Building on these insights, we release *ModernVBERT*, a compact 250M-parameter vision-language encoder that outperforms recent models up to 10 times larger when fine-tuned on document retrieval tasks, enabling efficient inference on cheap CPU hardware and greatly reducing latency and costs while maintaining strong performance. Models, code and data are available at <https://huggingface.co/ModernVBERT>.

5.1	Introduction	50
5.2	Methodology	52
5.2.1	Modality Alignment . . .	53
5.2.2	Contrastive Post-Training	54
5.2.3	Ablation Evaluation	
	Setup	54
5.3	What Makes a Great Visual Retriever?	55
5.3.1	Modality Alignment Design	55
5.3.2	Contrastive Training Design	57
5.4	Building a Small yet Mighty Visual Retriever.	58
5.4.1	Training.	58
5.4.2	Results.	59
5.5	Related Work	60
5.6	Conclusion	61

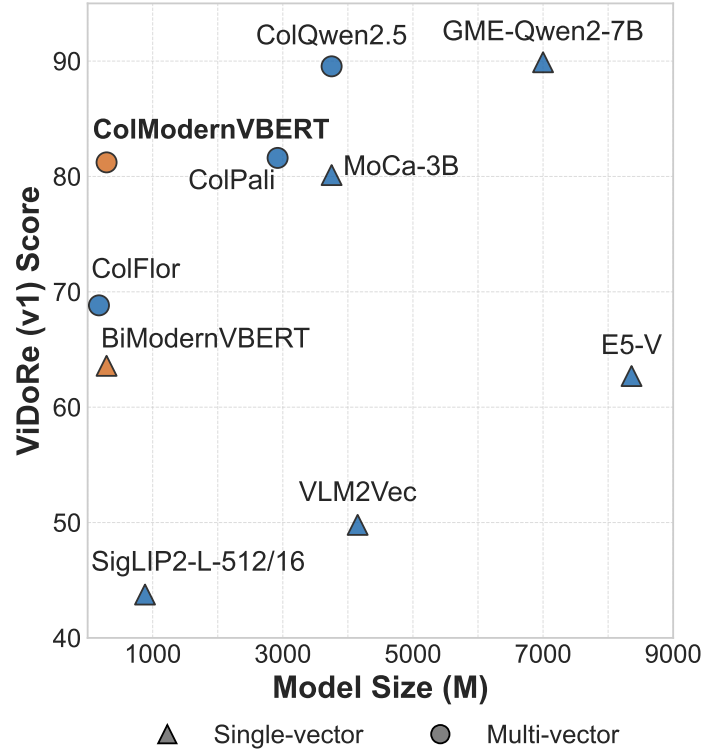


Figure 5.1: Pareto efficiency. ColModernVBERT outperforms models in its category on ViDoRe, achieving a leading performance-size tradeoff.

5.1 Introduction

The ability to quickly locate specific information in vast document collections is a core building block of digital systems today, supporting use cases that range from web search and virtual assistants to enterprise knowledge management. Neural information retrieval (IR) models, and in particular dense retrievers, have become the de facto backbone of modern search systems thanks to their strong semantic matching capabilities and good scalability properties [15, 64, 95].

This trend is amplified by the widespread adoption of Retrieval-Augmented Generation (RAG) [87], where a retriever is used to select a small set of relevant documents that condition a downstream generator. In such systems, the first-stage retrieval module is a well-known bottleneck: its recall directly upper-bounds the quality of the generated answers, while its latency and indexing costs partially drive the overall system efficiency [171]. As a result, improving document retrieval, especially for long, complex files such as PDFs, scientific articles, and reports, is a key lever for making industrial RAG deployments more accurate and cost-effective.

Visual Document Retrieval. Historically, document retrieval in these settings has operated purely in the text space. To index PDFs or scans, practitioners first run heavy preprocessing pipelines that include Optical Character Recognition (OCR), layout analysis, and heuristic passage segmentation, before embedding the resulting text spans with a neural encoder. This approach suffers from several limitations: OCR and layout parsing can be brittle and slow, complex visual elements such as tables, figures, and typography are often poorly captured, and any error or bias introduced during preprocessing is propagated to the retriever.

[15]: Reimers et al. (2019), *Sentence-BERT: Sentence Embeddings using Siamese BERT- Networks*

[64]: Karpukhin et al. (2020), *Dense Passage Retrieval for Open-Domain Question Answering*

[95]: Wang et al. (2022), *Text Embeddings by Weakly-Supervised Contrastive Pre-training*

[87]: Lewis et al. (2020), *Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks*

[171]: Lin et al. (2022), *Retrieval Augmented Visual Question Answering with Outside Knowledge*

Visual Document Retrieval (VDR) has emerged as a compelling alternative to such text-based systems. Rather than indexing pre-extracted textual content, VDR models directly operate on page screenshots: given a user query, they retrieve relevant document pages by matching the query against image-based representations of the pages [6]. By bypassing OCR and layout parsing, VDR yields simpler end-to-end pipelines, significantly reduces indexing latency, and better exploits visual cues such as layout, figures, and fonts, while achieving strong performance on visually rich benchmarks like ViDoRe.

Limits of Generative VLM Repurposing. Most current VDR systems are obtained by repurposing large generative vision–language decoders [108] as retrieval encoders via post-hoc contrastive fine-tuning [6, 140, 153]. While cost-efficient, this design choice bottlenecks retrieval performance and efficiency: model sizes, attention patterns, image resolutions, and training objectives are designed for generative use cases rather than optimized for retrieval which has been shown in text models to be suboptimal [23, 73]. Furthermore, scaling trends [172] are less pronounced for embedding models; while correlated with model size, strong retrieval performance remains attainable with small models [173].

Recent papers and model releases in the visual retrieval space have claimed performance improvements by scaling the amount of contrastive data and the compute budget [155, 174], modifying the attention mask [175], increasing image resolutions [97] or by introducing more diverse tasks and data sources [140].

In this work, we attempt to centralize these efforts and systematically disentangle the impact of core design decisions in visual retriever training. Through controlled experiments—ranging from language model pretraining to multi-stage, domain-specific fine-tuning, we aim to answer a central question:

Which design choices best boost performance in modern visual document retrievers?

Contribution 1. We revisit core assumptions in visual retriever design, showing that token-level training objectives benefit retrievers by strengthening image–text token alignment—rather than merely producing stronger image embeddings. Our results indicate that causal attention is suboptimal in document retrieval, with bidirectional masking offering clear improvements in multi-vector settings, and that other parameters such as image resolution data mixes should not be overlooked in the training pipeline.

Contribution 2: *ModernVBERT*. Building on these insights, we release *ModernVBERT*, a small 250M multimodal encoder that aligns a pretrained language encoder with a vision encoder through Masked Language Modeling (MLM) objective, and *ColModernVBERT* a variant fine-tuned for document retrieval. Despite its modest size and limited training budget, *ColModernVBERT* matches models 10x larger on standard visual document retrieval benchmarks, demonstrating the interest of designing a retrieval focused model from the ground up. We release the model, intermediate checkpoints, and the training code at <https://huggingface.co/ModernVBERT>.

[6]: Faysse et al. (2025), *ColPali: Efficient Document Retrieval with Vision Language Models*

[108]: Alayrac et al. (2022), *Flamingo: a Visual Language Model for Few-Shot Learning*

[6]: Faysse et al. (2025), *ColPali: Efficient Document Retrieval with Vision Language Models*

[140]: Jiang et al. (2024), *VLM2Vec: Training Vision-Language Models for Massive Multimodal Embedding Tasks*

[153]: Ma et al. (2024), *Unifying Multimodal Retrieval via Document Screenshot Embedding*

[23]: Gisserot-Boukhlef et al. (2026), *Should We Still Pretrain Encoders with Masked Language Modeling?*

[73]: Lee et al. (2024), *NV-Embed: Improved Techniques for Training LLMs as Generalist Embedding Models*

[172]: Wei et al. (2022), *Emergent Abilities of Large Language Models*

[173]: Clavié (2024), *Towards Better Monolingual Japanese Retrievers with Multi-Vector Models*

[155]: Xu et al. (2025), *Llama Nemoretriever ColEmbed: Top-Performing Text-Image Retrieval Model*

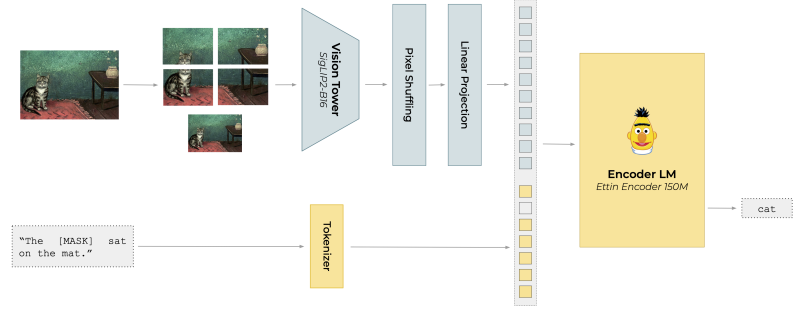
[174]: Zhang et al. (2025), *GME: Improving Universal Multimodal Retrieval by Multimodal LLMs*

[175]: Chen et al. (2025), *MoCa: Modality-aware Continual Pre-training Makes Better Bidirectional Multimodal Embeddings*

[97]: Cohere (2024), *Introducing Rerank 3: A New Foundation Model for Efficient Enterprise Search & Retrieval*

[140]: Jiang et al. (2024), *VLM2Vec: Training Vision-Language Models for Massive Multimodal Embedding Tasks*

Figure 5.2: MLM-based early fusion architecture. The visual encoder produces patch representations, which are passed to a language model. Our end-to-end bidirectional attention fused architecture is trained with Masked Language Modeling objectives and is perfectly suited for sequence and token-level representation tasks.



5.2 Methodology

Our analysis aims at quantifying the impact of design decisions made when training visual retrievers. In opposition to previous work, we begin our analysis as early as language model modality alignment and iteratively study design choices by modifying design choices independently to reduce confounding factors as much as possible [176].

Controlled Experimental Setup. A central point of interest is the impact of causal and bidirectional attention masks. While recently studied for textual representation applications [23, 177], we extend the experiment to the vision modality. We use checkpoints released by [23] which consist in a series of identical 210M parameter transformer models based on the Llama architecture [178] trained on 100B tokens that differ only in their attention masking strategy during language model training but that are perfectly identical in terms of training data seen, model size and architecture, learning rate scheduling, etc... The checkpoints we use are enc a bidirectional encoder trained with Masked Language Modeling (MLM), dec, a causal decoder trained with next token prediction, and dec-enc a causal decoder annealed over the end of its textual training by removing the causal mask and switching the training objective to MLM. For the vision tower, we employ the vision component of siglip2-base-16b-512 [179], a 86M parameter vision transformer contrastively trained on billions of text-image pairs. All ablations thus stem from iso-data controlled setups, and as further described, are further trained on the same data sequence, with the same batch sizes, optimizers, schedulers and on the same hardware.

Model Architecture. Our analyses are not centered around model architectures and to draw broadly applicable insights, we design vision-language models following current standard training practices. In line with most recent work, we employ the early fusion architecture [108] illustrated in Figure 5.2, in which visual patch embeddings produced by the vision encoder are projected into the language model input embedding space and concatenated with text token embeddings to encourage joint processing [108, 137, 180–182]. As described in subsection 5.2.1, we generalize the training loss to function both with causal and masked language modeling objectives. To handle dynamic resolutions, we split large images into 512×512 pixel tiles as expected by the SigLIP encoder¹. Following current standard practices, we further process a downsampled version of the full image to improve inter-tile consistency and global visual understanding [183, 184]. The vision tower produces 1024 patch representations for each tile², which we compress to 64 tokens

[176]: Allen-Zhu et al. (2025), *Physics of Language Models: Part 1, Learning Hierarchical Language Structures*

[23]: Gisserot-Boukhlef et al. (2026), *Should We Still Pretrain Encoders with Masked Language Modeling?*

[177]: Weller et al. (2025), *Seq vs Seq: An Open Suite of Paired Encoders and Decoders*

[23]: Gisserot-Boukhlef et al. (2026), *Should We Still Pretrain Encoders with Masked Language Modeling?*

[178]: Touvron et al. (2023), *Llama 2: Open Foundation and Fine-Tuned Chat Models*

[179]: Tschannen et al. (2025), *SigLIP 2: Multilingual Vision-Language Encoders with Improved Semantic Understanding, Localization, and Dense Features*

[108]: Alayrac et al. (2022), *Flamingo: a Visual Language Model for Few-Shot Learning*

[108]: Alayrac et al. (2022), *Flamingo: a Visual Language Model for Few-Shot Learning*

[137]: Wang et al. (2024), *Qwen2-VL: Enhancing Vision-Language Model's Perception of the World at Any Resolution*

[180]: Li et al. (2022), *BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation*

[181]: Yang et al. (2025), *Qwen2.5-1M Technical Report*

[182]: Marafioti et al. (2025), *SmolVLM: Redefining small and efficient multimodal models*

1: Images are downsampled (or upsampled) so that the lengths and widths reach a multiple of 512 pixels to preserve the aspect ratio, padding is used on the smaller side when necessary (i.e. a 1024×1000 px image would be padded to 1024×1024 px).

[183]: Lin et al. (2023), *SPHINX: The Joint Mixing of Weights, Tasks, and Visual Embeddings for Multi-modal Large Language Models*

[184]: Ye et al. (2023), *UReader: Universal OCR-free Visually-situated Language Understanding with Multimodal Large Language Model*

2: The SigLIP tower takes 512×512 px images and processes them by 16×16 px patches [107]. This results in $(512/16)^2 = 1024$ patches.

through *pixel shuffling* [185] with a compression ratio $r = 4$, following prior work on models of comparable size [182]. We highlight the impact of image resolution and this parameter on the number of visual tokens in Appendix E.3.6.

Training Procedure. Our experiments focus on retrieval performance. We employ a standard biphasic training procedure, in which we first run modality alignment to train a pretrained textual language model to understand visual inputs through language modeling objectives [109] (subsection 5.2.1), then rely on a second text-image contrastive learning phase to learn efficient image representations [100] (subsection 5.2.2). We further describe the general setup, and detail specific modifications to the default training procedure in the experiment section.

5.2.1 Modality Alignment

We align the vision encoder tower with the language model by training the image embedding projection layer to map visual features into the language model embedding space. The pretrained language model is also fine-tuned with Low-Rank Adapters (LoRA) [131], allowing both image and text models to adapt jointly while reducing the risk of monomodal performance collapse [108, 109, 111, 182, 186].

Alignment Loss. For decoder-based models, we train with Causal Language Modeling (CLM) loss on the text tokens, as standardly done in VLM modality alignment:

$$\mathcal{L}_{\text{CLM}} = - \sum_{t=1}^T \log P_{\theta}(x_t \mid x_{<t}), \quad (5.1)$$

where $x_{<t}$ denotes all tokens preceding position t . We generalize this training scheme to bidirectional encoders models, by using the Masked Language Modeling (MLM) loss on the textual tokens:

$$\mathcal{L}_{\text{MLM}} = - \sum_{t \in \mathcal{M}} \log P_{\theta}(x_t \mid x_{\setminus \mathcal{M}}), \quad (5.2)$$

where $\mathcal{M}$ is the set of masked token positions and $x_{\setminus \mathcal{M}}$ is the input with those tokens masked out.

Modality Alignment Corpus. Models are modality aligned on a large corpus in large parts derived from The Cauldron 2 [111] and Docmatix [138]. Our objective being to train document focused retrieval models, we use an adjusted training mixture that upsamples images containing text and documents with varying level of complexities. Our final training corpus consists of approximately 2B text tokens, and includes diverse sources such as web pages, books, and scientific papers. Mixture details are given in Appendix E.1.3. We note that controlling the exact data distribution during this phase enables the models we train to specialize early and achieve good document focused downstream performances which many large models struggle with [187].

Parameters. All models are trained using a masking ratio of 0.5 and user-prompt masking to avoid overfitting on chat-template format [188–190]. We employ WSD scheduler [191] with the first 5% of the training

[185]: Shi et al. (2016), *Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network*

[182]: Marafioti et al. (2025), *SmolVLM: Redefining small and efficient multimodal models*

[109]: Liu et al. (2023), *Visual Instruction Tuning*

[100]: Radford et al. (2021), *Learning Transferable Visual Models From Natural Language Supervision*

[131]: Hu et al. (2021), *LoRA: Low-Rank Adaptation of Large Language Models*

[108]: Alayrac et al. (2022), *Flamingo: a Visual Language Model for Few-Shot Learning*

[109]: Liu et al. (2023), *Visual Instruction Tuning*

[111]: Laurençon et al. (2024), *What matters when building vision-language models?*

[182]: Marafioti et al. (2025), *SmolVLM: Redefining small and efficient multimodal models*

[186]: McKinzie et al. (2024), *MM1: Methods, Analysis & Insights from Multimodal LLM Pre-training*

[111]: Laurençon et al. (2024), *What matters when building vision-language models?*

[138]: Laurençon et al. (2024), *Building and better understanding vision-language models: insights and future directions.*

[187]: Liu et al. (2023), *Improved Baselines with Visual Instruction Tuning*

[188]: Huerta-Enochian et al. (2024), *Instruction Fine-Tuning: Does Prompt Loss Matter?*

[189]: Shi et al. (2024), *Instruction Tuning With Loss Over Instructions*

[190]: Allal et al. (2025), *SmolLM2: When Smol Goes Big – Data-Centric Training of a Small Language Model*

[191]: Hu et al. (2024), *MiniCPM: Unveiling the Potential of Small Language Models with Scalable Training Strategies*

as warmup, the last 20% as decay and a maximum learning rate of $1e-4$. The ablation models are aligned on 3.5B tokens. We provide additional details on the training setup in Appendix E.1.1.

5.2.2 Contrastive Post-Training

Once the language model has learned to process image tokens jointly with text tokens, we specialize models through a contrastive post-training stage designed to enhance the semantic representation of the output embeddings produced by the model [15].

Post-training Pairs. The post-training dataset used as starting point in our ablations comprises 118k document-query pairs from the ColPali corpus [6] as well as another 118k of natural image-description pairs from the MSCOCO train set [117].

Contrastive Loss. We employ the InfoNCE loss [192], defined as

$$\mathcal{L}_{\text{InfoNCE}}(\mathbf{q}, \mathbf{d}^+) = -\log \frac{\Phi(\mathbf{q}, \mathbf{d}^+)}{\Phi(\mathbf{q}, \mathbf{d}^+) + \sum_{\mathbf{d}^- \in \mathcal{N}_q} \Phi(\mathbf{q}, \mathbf{d}^-)}, \quad (5.3)$$

where $\mathbf{d}^+$ denotes the positive target for the query $\mathbf{q}$, $\mathcal{N}_q = \mathcal{N}_q^{\text{in}} \cup \mathcal{N}_q^{\text{hard}}$ the set of negative targets (in-batch and hard negatives when mentioned), and $\Phi(\mathbf{q}, \mathbf{d})$ a similarity function between the token(s) of the query and the documents.³ For general-domain post-training we compute the loss symmetrically [100].

Batches Curation. In contrastive learning, batch diversity critically impacts retrieval entropy. Overly heterogeneous batches lead to trivial retrievals, while curated batches yield richer training signals. We employ task-aware batching [68], grouping documents by source to ensure a homogeneous batch composition.

5.2.3 Ablation Evaluation Setup

The contrastively trained models are evaluated on retrieval and zero-shot classification tasks across multiple domains. Although the main focus remains document retrieval capabilities, evaluated by aggregating scores from the ViDoRe and ViDoRe v2⁴ [7] benchmarks (nDCG@5), we also assess more generalist image retrieval capabilities by selecting tasks from MIEB [193]. For natural image retrieval, we aggregate MSCOCO retrieval [117] and Flickr30k retrieval (nDCG@10) [194] test sets. Finally, following practices in [66], we assess both zero-shot and fine-tuning abilities of our models on general classification tasks. Specifically, we measure classification accuracy by fine-tuning a logistic regression head on top of our model’s embedding on Stanford Cars [195] and Food101 [196], and we evaluate zero-shot performance on FER2013 [197] and EuroSAT [198] and aggregate the results.

[15]: Reimers et al. (2019), *Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks*

[117]: Lin et al. (2014), *Microsoft COCO: Common Objects in Context*

[192]: Oord et al. (2018), *Representation Learning with Contrastive Predictive Coding*

3: We use the last (EOS) token for causal models, and mean pool all sequence tokens for bidirectional encoders for single-vector models. Alternatively, we use all document and query tokens without pooling for late interaction matching [6]. Details in Appendix E.1.2

[100]: Radford et al. (2021), *Learning Transferable Visual Models From Natural Language Supervision*

[68]: Li et al. (2023), *Towards General Text Embeddings with Multi-stage Contrastive Learning*

4: We report only the English splits of ViDoRe v2, as our base models are trained on English data only.

[7]: Macé et al. (2026), *ViDoRe Benchmark V2: Raising the Bar for Visual Retrieval*

[193]: Xiao et al. (2025), *MIEB: Massive Image Embedding Benchmark*

[117]: Lin et al. (2014), *Microsoft COCO: Common Objects in Context*

[194]: Plummer et al. (2015), *Flickr30k Entities: Collecting Region-to-Phrase Correspondences for Richer Image-to-Sentence Models*

[66]: Muennighoff et al. (2022), *MTEB: Massive Text Embedding Benchmark*

[195]: Krause et al. (2013), *3D Object Representations for Fine-Grained Categorization*

[196]: Bossard et al. (2014), *Food-101 – Mining Discriminative Components with Random Forests*

[197]: Khairuddin et al. (2021), *Facial Emotion Recognition: State of the Art Performance on FER2013*

[198]: Helber et al. (2019), *EuroSAT: A Novel Dataset and Deep Learning Benchmark for Land Use and Land Cover Classification*

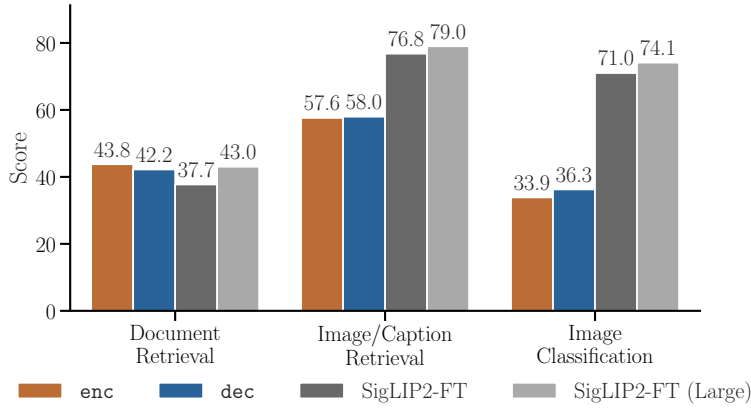


Figure 5.3: Impact of Modality Alignment objective on downstream tasks. Early Fusion of vision and text models boosts document retrieval tasks regardless of the LM objective, but degrades natural image and classification tasks w.r.t. the standalone *fine-tuned* vision model SigLIP. Reported scores are aggregated MIEB scores (nDCG, Accuracy.)

5.3 What Makes a Great Visual Retriever?

Vision-language retrievers built upon existing generative VLMs often inherit design choices and weights that may not be well suited for all embedding tasks. Here, we analyze these critical design choices hoping to derive clear insights for developing efficient visual retrievers. Importantly, although we assess design decisions at different stages of the training pipelines, evaluation are always done end-to-end on the final evaluation signal.

5.3.1 Modality Alignment Design

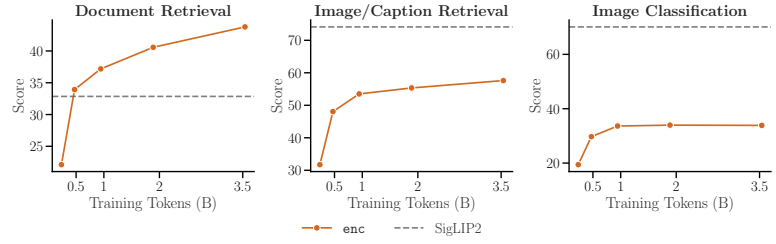
Language modeling Modality Alignment improves document understanding. According to benchmarks such as MIEB [193], dual encoder models explicitly trained on contrastive image-text tasks outperform repurposed VLMs in natural image classification tasks. To assess this, we train an encoder and a decoder vision-language model using the methodology described in section 5.2 on a mix of natural image and document data (alignment and contrastive training). We compare them with *SigLIP2-FT*, the 378M dual vision encoder model whose vision component is used by the vision tower of both VLMs, and with the larger *SigLIP2-FT Large* (881M parameters). Both SigLIP-FT models are finetuned in the same conditions as the VLMs, and initialized from pre-trained weights from scratch on billions of text-image pairs⁵. As shown in Figure 5.3, the two early fusion VLM variants severely underperform the SigLIP2-FT dual encoders on natural image tasks. In contrast, they achieve significant gains on document retrieval tasks (+6.1 nDCG@5 on ViDoRe and ViDoRe v2 datasets w.r.t. base), even edging out *SigLIP2-FT Large* that contains 3.5x vision parameters more than both VLMs.

This confirms large-scale contrastive training remains best for high-level image representation tasks (natural images), but sequentially combining a vision model with a pretrained language model facilitates document representation tasks, even with significantly less contrastive post-training. As the rest of this paper shows, steering away from the dual encoder architecture further enables improving performance through many avenues other than text to image contrastive training, for which supervised training samples can be hard to obtain.

[193]: Xiao et al. (2025), *MIEB: Massive Image Embedding Benchmark*

5: We report the performance of the untrained *off-the-shelf* SigLIP in Appendix E.3.1

Figure 5.4: Modality alignment scaling of early fusion encoders for up to 1 epoch (3.5B tokens) of data. The dashed line indicates the vision encoder evaluated standalone without further training. Our findings show that retrieval tasks benefits from extended modality alignment phase, particularly in document retrieval, where performance quickly surpasses that of the standalone vision encoder.



[112]: Beyer et al. (2024), *PaliGemma: A versatile 3B VLM for transfer*
 [137]: Wang et al. (2024), *Qwen2-VL: Enhancing Vision-Language Model's Perception of the World at Any Resolution*
 [186]: McKinzie et al. (2024), *MM1: Methods, Analysis & Insights from Multimodal LLM Pre-training*

[23]: Gisserot-Boukhlef et al. (2026), *Should We Still Pretrain Encoders with Masked Language Modeling?*

[177]: Weller et al. (2025), *Seq vs Seq: An Open Suite of Paired Encoders and Decoders*

6: [175] investigate post-hoc removal of the attention mask during visual retrieval fine-tuning.

[6]: Faysse et al. (2025), *ColPali: Efficient Document Retrieval with Vision Language Models*

[78]: Khattab et al. (2020), *ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT*

[177]: Weller et al. (2025), *Seq vs Seq: An Open Suite of Paired Encoders and Decoders*

Scaling the modality alignment phase for better token representations.

Prior work shows that scaling the modality alignment phase of VLMs improves their generative abilities [112, 137, 186]. We test whether similar gains hold in retrieval by contrastively fine-tuning enc checkpoints during MLM modality alignment. Figure 5.4 illustrates the results of post-trained checkpoints on diverse tasks. Although document retrieval improves consistently with more modality alignment data – largely surpassing the vision tower evaluated in isolation and showing clear scaling benefits – natural image tasks plateau past 1B tokens, far from the standalone dual encoder baseline. This shows that document and natural image retrieval leverage different mechanisms and should not be optimized the same way. *Document Retrieval benefits from learning fine-grained interactions between image and text tokens through the language model, while the LM has limited utility for high level natural image tasks.*

Bidirectional attention fully unlocks Late Interaction. Inspired by the effectiveness of bidirectional attention in text-only retrieval [23, 177]⁶, we investigate if it surpasses causal attention in *visual document retrieval*, particularly when using the multi-vector late interaction matching common in SOTA visual retrievers [6, 78]. Figure 5.5 reports single vector and late interaction results on the ViDoRe benchmark for various model variants. On top of the standard enc (MLM) and dec (CLM) models, we evaluate the dec-enc and the dec models modality aligned with MLM objectives to determine whether bidirectional attention capabilities can be obtained in later stages of training.

Single-vector embedding results are close between bidirectional and causal attention models for document retrieval, with enc slightly outperforming dec by +1.6 nDCG@5.

Intuitively however, bidirectional attention makes a huge difference when used in late interaction settings, substantially exceeding the causal counterpart by +10.6 nDCG@5. Causal decoders are incapable of correctly contextualizing image or text token representations seen at the beginning of the sequences. *This is a key result as almost all current visual retrievers, including late interaction variants, are causal models, clearly indicating some performance is left on the table.*

Removing the causal attention mask during training does not suffice to recover the enc late interaction performance at these data regimes. This indicates converting trained decoders as late interaction retrievers is highly non trivial, and confirms the insights from [177]; when possible, training encoder models from scratch remain better for retrieval tasks.

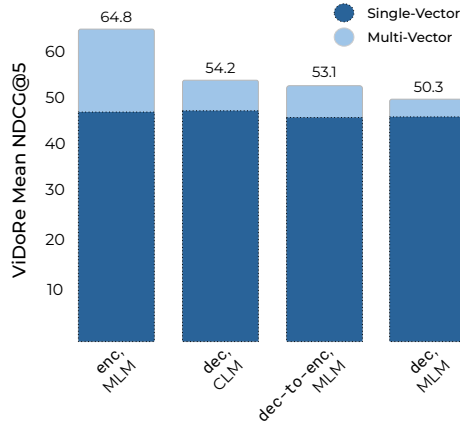


Figure 5.5: Impact of attention masks and training objectives on document retrieval performances. We report the average nDCG@5 on English splits of ViDoRe benchmarks for models post-trained on ColPali.

HR Cooldown		Document Retrieval	Image/Caption Retrieval	Image Classification	Average
512px	✗	30.7	<u>58.8</u>	41.4	43.6
1024px	✗	42.2	58.9	<u>37.2</u>	46.1
2048px	✗	<u>43.8</u>	57.6	33.9	45.1
2048px	✓	45.8	57.8	33.7	<u>45.8</u>

Figure 5.6: Effect of image resolution on VL encoder abilities. Document retrieval performance increases with higher image resolution. Further annealing the encoder on high-resolution images (HR Cooldown) at the end of modality alignment yields additional gains. By contrast, for non-document tasks, raising the resolution tends to degrade performance.

5.3.2 Contrastive Training Design

The previous subsection established bidirectional encoder models to often be the best option when training visual retrievers. In the following experiments, we assess contrastive training choices and only report results for the encoder model for simplicity.

Image resolution benefits are task-specific. Image resolution plays a critical role in VLM generative capabilities, notably in document-focused tasks, as higher-resolution inputs enables the model to capture finer visual cues [182, 199]. Modality alignment is done at a fixed image resolution of 1024 pixels (longer side) and we report scores of contrastive training runs with varying settings in Table 5.6. To vary the resolution, images of the highest quality available are scaled to the desired size (often downsampled) before being fed to the image tokenizer. Our findings confirm that embedding tasks are strongly sensitive to image-resolution. In particular, *training with higher resolution inputs substantially improves the results on visual document retrieval benchmarks*, consistent with prior work in generative settings [112, 186]. Furthermore, adding a cool-down phase by showing higher-resolution images towards the end of the modality alignment phase yields additional gains. This suggests that models can adapt their attention mechanisms to finer details when exposed to increased resolution. Interestingly, these findings do not hold in natural image tasks, where high resolution can even degrade performance.

Increasing the pool of contrastive pairs. A severe limitation that current visual retrievers face is the lack of large volumes of high quality (document

[182]: Marafioti et al. (2025), *SmolVLM: Redefining small and efficient multimodal models*

[199]: Hu et al. (2024), *mPLUG-DocOwl2: High-resolution Compressing for OCR-free Multi-page Document Understanding*

Table 5.1: Impact of contrastive training mixtures on downstream tasks. Incorporating text-only pairs improves performance on document retrieval, but degrades other performances. Adding natural images-captions pairs substantially enhances performance on classification tasks.

	Document Retrieval	Image/Caption Retrieval	Image Classification	Average
Baseline CL Mix	43.9	57.2	36.1	45.7
+ Text→Text Pairs	45.6	53.2	35.7	44.8
+ Image→Caption Pairs	45.8	54.4	49.9	50.0

[6]: Faysse et al. (2025), *ColPali: Efficient Document Retrieval with Vision Language Models*

[140]: Jiang et al. (2024), *VLM2Vec: Training Vision-Language Models for Massive Multimodal Embedding Tasks*

[153]: Ma et al. (2024), *Unifying Multimodal Retrieval via Document Screenshot Embedding*

[174]: Zhang et al. (2025), *GME: Improving Universal Multimodal Retrieval by Multimodal LLMs*

[153]: Ma et al. (2024), *Unifying Multimodal Retrieval via Document Screenshot Embedding*

[200]: Jiang et al. (2024), *E5-V: Universal Embeddings with Multimodal Large Language Models*

image, query pairs). Previous work [6, 140, 153, 174] has relied on a mix of repurposed existing visual question answering datasets and synthetically generated queries with external LLMs. Even put together however, the field is only a year old, and these datasets remain small in size and often of poor quality.

A central question in our study is whether the abundance of *text-only* query–document pairs can be exploited to improve *visual* retrieval via cross-modal capability transfer. To probe this, we run contrastive training under three regimes. Unlike prior work that “warms up” visual retrievers or trains exclusively with text-only pairs [153, 200], we *interleave* text-only pairs and text–image pairs throughout training at a 1:1 ratio. The dataset sources are detailed in Appendix 5

As reported in Table 5.1, incorporating text-only pairs yields a sizeable improvement on visual document retrieval (+1.7 nDCG@5), indicating clear cross-modal transfer—likely facilitated by the backbone’s jointly learned text–image embedding space. This result suggests that domain-specific training corpora can be assembled irrespective of native modality, reducing duplication of effort and lowering data-collection costs.

We further evaluate training with *NatCap*, a corpus of natural images paired with synthetic, highly detailed captions (see Appendix E.1.3). This scaling step improves downstream performance across the board—most notably on natural-image tasks, and with a smaller but consistent gain on document retrieval (+0.2 nDCG@5). Together, these findings underscore the importance of scaling contrastive learning with high-quality data, but which doesn’t need to be exclusively image document focused.

5.4 Building a Small yet Mighty Visual Retriever.

5.4.1 Training.

Recipe. Putting together the results from our experiments, we devise a training recipe for a small visual document retriever *ModernVBERT*. It combines a state-of-the-art 150M text bidirectional encoder [177] with the ModernBERT architecture [201] and a small vision encoder SigLIP2-16B-512 of 100M parameters [179]. We modality align both models with a MLM objective for 10B tokens, 3 times longer than during our experiments. To boost document understanding, we augment the input image resolution from 1024px to 2048px during a modality alignment cooldown stage (2B tokens). We call the resulting model *ModernVBERT*. Following the

[177]: Weller et al. (2025), *Seq vs Seq: An Open Suite of Paired Encoders and Decoders*

[201]: Warner et al. (2024), *Smarter, Better, Faster, Longer: A Modern Bidirectional Encoder for Fast, Memory Efficient, and Long Context Finetuning and Inference*

[179]: Tschannen et al. (2025), *SigLIP 2: Multilingual Vision-Language Encoders with Improved Semantic Understanding, Localization, and Dense Features*

Table 5.2: Performance on ViDoRe. Our model *ColModernVBERT* offers the best performance-size tradeoff, significantly outperforming existing sub-1B models and matching the performance of models up to 10x larger with substantially lower inference CPU latency. Details and GPU latencies in Appendix E.3.6. Models marked with * are not specifically trained for VDR. Bold values indicate the best performance amongst sub-1B models.

	Late Interaction	Model Size (B)	ViDoRe(v1)	ViDoRe(v2,eng)	Average	Latency (ms)
$\geq 1B$ Parameters						
MoCa-3B [175]		3.75	80.1	53.8	66.9	158
VLM2Vec [140]		4.15	49.8	36.5	43.1	211
GME-Qwen2 [174]		8.29	89.9	61.8	75.8	412
E5-V [200]		8.36	62.7	49.4	56.1	434
ColPali [6]	✓	2.92	81.6	56.8	69.2	175
ColQwen2.5 [6]	✓	3.75	89.5	61.5	75.5	158
Jina-v4 [149]	✓	3.75	90.4	60.1	75.2	158
NemoRetriever-3B [155]	✓	4.40	91.0	66.3	78.7	155
$\leq 1B$ Parameters						
Jina CLIP* [129]		0.22	17.6	14.0	15.8	14
BGE Visualized M3* [202]		0.87	12.4	10.2	11.3	38
SigLIP2-L-512/16* [179]		0.88	43.8	27.0	35.4	25
ColFlor [203]	✓	0.17	68.8	43.0	55.9	17
<i>BiModernVBERT</i> (ours)		0.25	63.6	35.7	49.7	20
<i>ColModernVBERT</i> (ours)	✓	0.25	81.2	56.0	68.6	20

findings of Section 5.3.2, we then scale the contrastive training mix from previous experiments to combine document–query pairs with text-only pairs, and use 1 hard negatives for each document–query pair and 2 for each text-only pairs. We opt for a 2/1 text-to-image ratio following our ablation results introduced in Appendix 7. This results in *ColModernVBERT*, a compact late interaction model. For reference, we also train *BiModernVBERT*, a single vector variant. More training details are provided in Appendix E.1.1.

5.4.2 Results.

***ColModernVBERT*.** The resulting model, *ColModernVBERT*, showcases strong performance on visual document retrieval benchmarks, especially relative to its size category (Figure 5.1). Despite having over 10 times fewer parameters than models such as ColPali released only a year ago, it is only 0.6 nDCG@5 points below on the aggregated ViDoRe benchmark scores (Table 5.2). It also edges many larger single-vector repurposed VLM models released within the year [140, 175, 200]. It however falls short of top model performance on ViDoRe, which is achieved by larger decoder VLMs pretrained and aligned on billions of tokens of text and image data.

Most sub-1B parameter models evaluated on document retrieval benchmarks are dual encoder models, since early fusion generative models that perform well are not common at this scale. The most related model is a 176M late interaction model, ColFlor [203], trained from the Florence2 model [204]. ColFlor is 12.7 nDCG@5 points under *ColModernVBERT*.

[140]: Jiang et al. (2024), *VLM2Vec: Training Vision-Language Models for Massive Multimodal Embedding Tasks*

[175]: Chen et al. (2025), *MoCa: Modality-aware Continual Pre-training Makes Better Bidirectional Multimodal Embeddings*

[200]: Jiang et al. (2024), *E5-V: Universal Embeddings with Multimodal Large Language Models*

[203]: Masry et al. (2024), *ColFlor: Towards BERT-Size Vision-Language Document Retrieval Models*

[204]: Xiao et al. (2023), *Florence-2: Advancing a Unified Representation for a Variety of Vision Tasks*

[205]: Xiao et al. (2025), *MetaEmbed: Scaling Multimodal Retrieval at Test-Time with Flexible Late Interaction*

7: With more standard CPU RAM settings such as those found in low-end servers or Google Colab (12GB RAM), models above 3B parameters must rely on memory offloading to run, which adds up to dozens of seconds of latency per query.

[108]: Alayrac et al. (2022), *Flamingo: a Visual Language Model for Few-Shot Learning*

[110]: Bai et al. (2023), *Qwen-VL: A Versatile Vision-Language Model for Understanding, Localization, Text Reading, and Beyond*

[206]: Lucas Beyer* et al. (2024), *PaliGemma*

[6]: Faysse et al. (2025), *ColPali: Efficient Document Retrieval with Vision Language Models*

[140]: Jiang et al. (2024), *VLM2Vec: Training Vision-Language Models for Massive Multimodal Embedding Tasks*

[153]: Ma et al. (2024), *Unifying Multimodal Retrieval via Document Screenshot Embedding*

[174]: Zhang et al. (2025), *GME: Improving Universal Multimodal Retrieval by Multimodal LLMs*

[155]: Xu et al. (2025), *Llama Nemoretriever Colembd: Top-Performing Text-Image Retrieval Model*

[207]: Nussbaum et al. (2024), *Nomic Embed: Training a Reproducible Long Context Text Embedder*

[23]: Gisserot-Boukhlef et al. (2026), *Should We Still Pretrain Encoders with Masked Language Modeling?*

[177]: Weller et al. (2025), *Seq vs Seq: An Open Suite of Paired Encoders and Decoders*

[175]: Chen et al. (2025), *MoCa: Modality-aware Continual Pre-training Makes Better Bidirectional Multimodal Embeddings*

[140]: Jiang et al. (2024), *VLM2Vec: Training Vision-Language Models for Massive Multimodal Embedding Tasks*

[155]: Xu et al. (2025), *Llama Nemoretriever Colembd: Top-Performing Text-Image Retrieval Model*

[22]: Boizard et al. (2025), *EuroBERT: Scaling Multilingual Encoders for European Languages*

[201]: Warner et al. (2024), *Smarter, Better, Faster, Longer: A Modern Bidirectional Encoder for Fast, Memory Efficient, and Long Context Finetuning and Inference*

[78]: Khattab et al. (2020), *ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT*

ColModernVBERT also largely outperforms off-the-shelf dual encoders, even when those have substantially larger parameter counts. These results highlight the benefits of multi-phase training and early fusion architectures for multimodal document-related tasks, even at smaller parameter counts. We also attribute the strong performance of *ColModernVBERT* at smaller model sizes to the symbiosis of native bidirectional attention and Late Interaction matching, which largely boosts performance relative to comparable decoder models (Section 5.3.1).

Speed. As noted by [205], multi-vector visual retrievers are not bottlenecked in their inference speed by the late interaction matching operation, but rather by the latency required to encode queries with the text model. Our model demonstrates that strong performance is not incompatible with speed, even when running inference on consumer CPUs, which is the standard setting in most industrial local deployments of text embedding models. Latencies are computed by averaging query encoding times of all NanoBEIR queries, which are 23.4 words and 147.5 characters long on average, and are run with batch size 1 to replicate online use cases. To prevent RAM bottlenecks, we benchmark on very high RAM (2TB) CPU cloud environments, but note models larger than 3B parameters require more than 12 GB RAM to run optimally.⁷ (Table 5.2). *ModernVBERT* achieves more than a 7x speedup on CPU over models with similar performances on ViDoRe. We further report model latency results on GPU hardware in Appendix E.3.6. We notably demonstrate that with batched inference, *ModernVBERT* based query encoders are able to encode 5000 queries per second on Nvidia H100 GPUs. *ModernVBERT*'s small model size also enables efficient batching when encoding documents.

5.5 Related Work

Repurposing VLMs for Representation Learning. Motivated by the zero-shot performances of generative VLMs [108, 110, 206], recent studies have explored repurposing these for multimodal embedding tasks [6, 140, 153, 174]. As backbone generative models improved, retriever performance improved as well showcasing the central impact of language model pretraining and modality alignment [155, 207]. These models remain inherently constrained by their causal attention mechanisms, which have been shown in text settings to limit representational efficiency [23, 177]. Recent work attempts to address this issue by modifying VLM attention during continual pretraining [175] or contrastive tuning [140, 155], but no recent work attempts to align natively bidirectional language encoder models with vision encoders. The recent release of long sequence text encoders [22, 201] makes this possible.

Late Interaction in Visual Document Retrieval To further boost performance, visual document retrievers leverage the late interaction mechanism [78] which matches multiple query embeddings with multiple document embeddings through the MaxSim operation [6, 149, 155]. This enables more granular interactions between image and query tokens, at the cost of additional storage and a slight compute overhead during the matching operation. Efficiency gains have come from improving the storage costs through quantization [208], token pruning [12] and more

recently the use of Matrioshka losses to compact multi-token representations [205]. Ultimately, the performance bottleneck when running visual retrieval inference with such models now resides mostly in the necessity to rely on costly GPU hardware to encode queries, which sets apart text from vision retrieval. This paper fills this gap by using encoders that run on CPU, of parameter sizes comparable to commonly used local text embedding models [19, 128].

5.6 Conclusion

In this paper, we question design decisions of current VLM-based retriever models, providing crucial insights into what matters when training early-fusion vision encoders. Our study notably shows that these models generally do not improve retrieval on natural-image tasks compared to dual encoders, yet strong vision-language alignment is essential for document-centric retrieval. We uncover a tight synergy between bidirectional attention and late-interaction retrieval, which underscores a fundamental limitation of repurposing decoder-style generative VLMs for retrieval. To mitigate data scarcity in contrastive learning, we propose augmenting limited image-document/text-query pairs with larger, lower-cost corpora from other modalities. Guided by these insights, we trained *ColModernVBERT*, a compact yet powerful 250M-parameter multimodal encoder that matches the performance of models up to 10× larger on visual retrieval benchmarks. We release models and training code to help practitioners reduce cost and latency when deploying visual retrievers in real-world applications, and to encourage research on efficient multimodal embedding models.

Future Work & Limitations. By design, our analysis targets relatively small models. An important next step is to test whether the observed patterns persist at larger scales—for example, to more rigorously probe the interplay between late interaction and bidirectional attention. Our study also focuses exclusively on English. While we expect the broad trends to generalize and see clear value in releasing multilingual variants, it remains unclear how allocating parameters to additional languages trades off against the understanding of the vision modality, and to what extent this penalizes English retrieval performance as the number of languages is scaled [209]. Finally, although we center on retrieval and sequence-level zero-shot classification, the modality-aligned encoder can be fine-tuned for a range of token-level tasks, including OCR error detection, token-level classification, visual named entity recognition, visually grounded token-level object detection, contextual embeddings [10]. We release our base model to encourage exploration of these directions.

[6]: Faysse et al. (2025), *ColPali: Efficient Document Retrieval with Vision Language Models*

[149]: Günther et al. (2025), *jina-embeddings-v4: Universal Embeddings for Multimodal Multilingual Retrieval*

[155]: Xu et al. (2025), *Llama Nemoretriever ColEmbed: Top-Performing Text-Image Retrieval Model*

[208]: Bergum (2025), *Scaling ColPali to billions of PDFs with Vespa — blog.vespa.ai*

[12]: Faysse et al. (2025), *CroissantLLM: A Truly Bilingual French-English Language Model*

[205]: Xiao et al. (2025), *MetaEmbed: Scaling Multimodal Retrieval at Test-Time with Flexible Late Interaction*

[19]: Enevoldsen et al. (2025), *MMTEB: Massive Multilingual Text Embedding Benchmark*

[128]: Chen et al. (2024), *BGE M3-Embedding: Multi-Lingual, Multi-Functionality, Multi-Granularity Text Embeddings Through Self-Knowledge Distillation*

[209]: Fernandes et al. (2023), *Scaling Laws for Multilingual Neural Machine Translation*

[10]: Conti*, Faysse* et al. (2025), *Context is Gold to find the Gold Passage: Evaluating and Training Contextual Document Embeddings*

CONTEXTUALIZED DOCUMENT RETRIEVAL

Evaluating and Training Contextual Document Embeddings

6

Paper Context

In this paper released in April 2025 and presented as an Oral at EMNLP 2025, we investigate whether we can embed *priors* on the corpus structure (associative links) in document embeddings. The main idea is that corpus documents are not constructed in isolation: for instance, pages of a book are not independent of one another. Understanding documents requires context, and representing them should as well. This paper proposes propagating information from neighboring documents linked by metadata, into the learned document representation. The goal is to boost future associative recall mechanisms to trigger on links that could not exist solely based on the content of the document itself. This paper and our proposed mechanism were notably the basis of subsequent work by Perplexity AI, a large *search* company.

Within the framework of this thesis, the same conclusions can be drawn as in the *ColPali* chapter. First, leveraging available *priors* on a corpus boosts the performance of standard retrieval patterns and enables novel use cases. Second, once again, Vannevar Bush’s philosophy of driving search through associative links is successfully demonstrated.

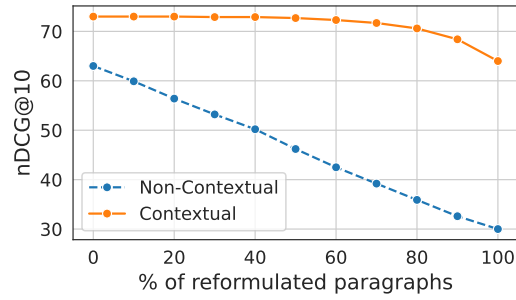
Paper Abstract

A limitation of modern document retrieval embedding methods is that they typically encode passages (chunks) from the same documents independently, often overlooking crucial contextual information from the rest of the document that could greatly improve individual chunk representations.

In this work, we introduce *ConTEB* (Context-aware Text Embedding Benchmark), a benchmark designed to evaluate retrieval models on their ability to leverage document-wide context. Our results show that state-of-the-art embedding models struggle in retrieval scenarios where context is required. To address this limitation, we propose *InSeNT* (In-sequence Negative Training), a novel contrastive post-training approach that, combined with *late chunking* pooling, enhances contextual representation learning while preserving computational efficiency. Our method significantly improves retrieval quality on *ConTEB* without sacrificing base model performance. We further find chunks embedded with our method are more robust to suboptimal chunking strategies and larger retrieval corpus sizes. We open-source all artifacts at <https://github.com/illuin-tech/contextual-embeddings>.

6.1	Introduction	66
6.2	Problem Formulation & Related Work	67
6.2.1	Retrieval Frameworks . .	67
6.2.2	Integrating Contextual Information	68
6.3	<i>ConTEB</i> : Context-aware Text Embedding Bench- mark	69
6.3.1	Benchmark Design . . .	69
6.3.2	Benchmark Construction	70
6.3.3	Training Dataset	70
6.3.4	Baselines	71
6.4	Training Contextual Embedders	71
6.4.1	Architecture	71
6.4.2	Learning Objective . . .	72
6.4.3	Model training	73
6.5	Results	73
6.6	Ablations	76
6.7	Conclusions	76

Figure 6.1: Importance of Contextual Information: Starting from a set of queries and mostly self-contained document paragraphs from the *Football*, we progressively reformulate paragraphs to remove information redundant with the rest of the document. This leads to sharp performance declines in standard retrieval approaches, but not in contextual retrieval approaches.



6.1 Introduction

The ability to rapidly process and query large-scale textual corpora is a cornerstone of many industrial applications, ranging from the analysis of medical records and legal briefs to large-scale administrative archives. As these collections grow in size and complexity, advanced approaches to information retrieval (IR), particularly Retrieval-Augmented Generation (RAG) [87], have attracted widespread interest; yet dealing with long documents remains an open challenge.

[87]: Lewis et al. (2020), *Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks*

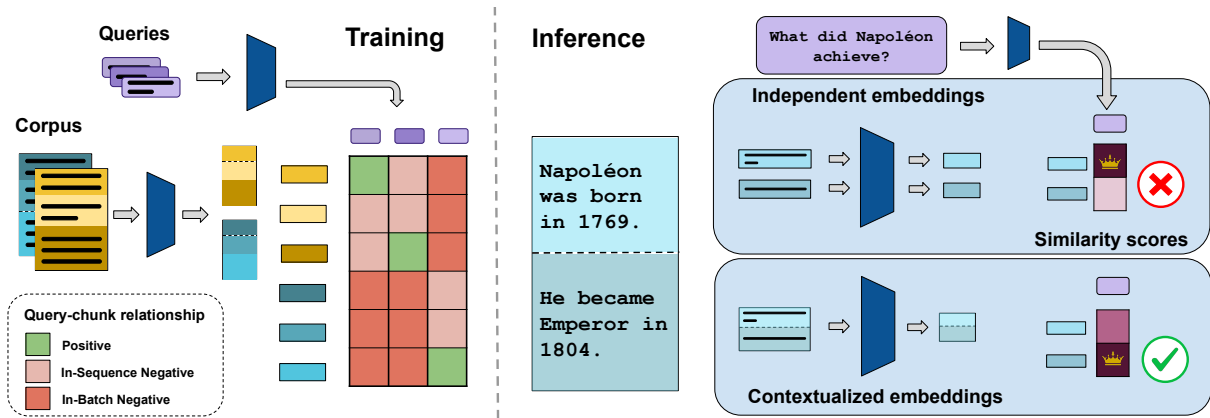


Figure 6.2: Training (Left). With respect to a single query, each chunk inside a batch plays a different role, depending on its original document, and the positive chunk. **Inference (Right).** Traditional embedding methods (top) produce embeddings that do not include potentially essential contextual information. Contextualized embeddings (bottom) can integrate document-wide information in individual chunk representations, augmenting embedding relevance and improving downstream retrieval performance.

[22]: Boizard et al. (2025), *EuroBERT: Scaling Multilingual Encoders for European Languages*

[201]: Warner et al. (2024), *Smarter, Better, Faster, Longer: A Modern Bidirectional Encoder for Fast, Memory Efficient, and Long Context Finetuning and Inference*

[210]: Zhang et al. (2024), *mGTE: Generalized Long-Context Text Representation and Reranking Models for Multilingual Text Retrieval*

[211]: Zhu et al. (2024), *LongEmbed: Extending Embedding Models for Long Context Retrieval*

[212]: Xu et al. (2024), *Retrieval meets Long Context Large Language Models*

[213]: Jiang et al. (2024), *LongRAG: Enhancing Retrieval-Augmented Generation with Long-context LLMs*

While long context encoders have been recently developed [22, 201, 210] along with long context embedding models [211], modern document retrieval pipelines typically segment lengthy documents into smaller chunks to optimize the granularity for efficient retrieval and readability of the retrieved content [212, 213]. Traditionally, these chunks are then *independently* fed to an embedding model, and stored in a vector database for efficient future query matching. By doing so, these systems remove strong semantic and conceptual links between the split passages, directly affecting the resulting representations. An example is illustrated in Figure 6.2: embedding the sentence "He became emperor in 1804." without leveraging information about the person at hand (*Napoléon*) given in previous paragraphs will make matching queries related to *Napoléon* difficult.

Recognizing the significant business value of incorporating broader contextual information into retrieval, major companies have explored

leveraging large generative language models (LLMs) to mitigate this limitation. Some approaches attempt to circumvent retrieval altogether by feeding millions of tokens into the model’s context window at runtime [214], while others reformulate individual passages by concatenating them with document-level summaries and context [215]. However, these methods are prohibitively expensive at scale when dealing with corpora comprising thousands of documents.

Despite the critical importance of contextualized retrieval, standard benchmarks fail to capture this challenge. Evaluations traditionally focus on assessing the effectiveness of embedding models [65, 66, 216], but they rely on datasets where document chunks are, by design, self-contained answers to the queries, which is a largely idealized scenario in practice [141]. Consequently, benchmarks fail to highlight the limitations of current retrieval strategies in handling context-dependent passages. Worse, recent findings by [217] indicate that some widely-used benchmarks exhibit biases that favor standard context-agnostic retrieval methods. Companies such as Anthropic have acknowledged these issues and maintain proprietary contextual retrieval benchmarks that remain unavailable to the public¹, underscoring the gap between academic evaluations and real-world industrial needs.

Contribution 1: *ConTEB*. We introduce the *Context-aware Text Embedding Benchmark*, designed to assess the ability of retrieval systems to leverage information from the entire document when indexing and retrieving document chunks. *ConTEB* comprises both custom-designed tasks for fine-grained analysis, and practical retrieval evaluation settings spanning multiple document types, domains, and situations in which leveraging context is helpful to produce more meaningful chunk representations. We evaluate standard embedding methods on the benchmark and find they struggle when contextual awareness is required, while highlighting the promising contextual capabilities of approaches such as *Late Chunking* [85].

Contribution 2: *Efficient Contextual Training*. Improving upon the *Late Chunking* method [85], we propose a novel embedding post-training method that optimizes information propagation between same-document chunks at indexing time to ensure embeddings are better contextualized. Our method largely boosts performance on *ConTEB* with minimal computational overhead. Through extensive ablations, we detail critical design choices and show our method displays increased robustness to sub-optimal chunking strategies and produces representations that scale better with corpus size.

We open-source all project artifacts, including the benchmark, models and training data².

[214]: Gemini Team et al. (2024), *Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context*

[215]: Anthropic (2024), *Introducing Contextual Retrieval*

[65]: Thakur et al. (2021), *BEIR: A Heterogenous Benchmark for Zero-shot Evaluation of Information Retrieval Models*

[66]: Muennighoff et al. (2022), *MTEB: Massive Text Embedding Benchmark*

[216]: Saad-Falcon et al. (2024), *Benchmarking and Building Long-Context Retrieval Models with LoCo and M2-BERT*

[141]: Thakur et al. (2025), *FreshStack: Building Realistic Benchmarks for Evaluating Retrieval on Technical Documents*

[217]: Zhou et al. (2025), *GSM-Infinite: How Do Your LLMs Behave over Infinitely Increasing Context Length and Reasoning Complexity?*

1: <https://www.anthropic.com/news/contextual-retrieval>

[85]: Günther et al. (2024), *Late Chunking: Contextual Chunk Embeddings Using Long-Context Embedding Models*

[85]: Günther et al. (2024), *Late Chunking: Contextual Chunk Embeddings Using Long-Context Embedding Models*

2: <https://github.com/illuin-tec h/contextual-embeddings>

6.2 Problem Formulation & Related Work

6.2.1 Retrieval Frameworks

In this paper, we consider the traditional retrieval framework where a retrieval system given a query q , searches a corpus $\mathcal{D}$ for relevant documents. Each document $d \in \mathcal{D}$ is scored based on its content by first

embedding the text into a vector space, and then computing a similarity measure. The similarity between a query q and a document d is defined as

$$\text{sim}(q, d) = f(\phi(q), \phi(d))$$

where ϕ maps text into an n -dimensional vector space and $f : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$ is a similarity function, such as cosine similarity or dot product.

In applied settings, individual documents are often too long to be practical for retrieval purposes [218, 219]. Each document d is thus divided into segments called *chunks* by a partitioning function $\mathcal{P}$ defined as

$$\mathcal{P}(d) = \{c_1, c_2, \dots, c_{N_d}\}$$

In the *standard retrieval* setting, the score is computed solely based on chunk content:

$$\text{sim}(q, c) = f(\phi(q), \phi(c))$$

Additional information (priors) is however often available to the document embedding system. Typically, knowledge of the entire corpus $\mathcal{D}$, or of *structural metadata* M_c such as neighboring document chunks obtained through $\mathcal{P}$, can be leveraged by a modified embedding function ϕ_2 , yielding the following similarity score:

$$\text{sim}(q, c) = f(\phi(q), \phi_2(c, M_c, \mathcal{D}))$$

This work is centered on efficiently integrating priors about the entire document when embedding a sub-document chunk.

6.2.2 Integrating Contextual Information

Neural embedding models for passage-level text representation, popularized by SentenceBERT [15], have enabled retrieval systems to move beyond lexical matching [40]. To include contextual information in these retrievers, previous works proposed methods that either operate *offline* during *indexing*, or online during *querying* when faced with a user request.

Indexing. The chunking strategy is a crucial design choice and often aims to optimize chunk self-containment. Fixed-size approaches with overlaps preserve continuity, while structure-aware chunking respects natural text boundaries, such as paragraphs or sentences. Semantic chunking, by contrast, splits text into topic-aligned segments. These methods appear in frameworks such as LlamaIndex [218] and LangChain [220], but different queries may need different chunk sizes. Thus, dynamic chunking techniques have emerged to adapt segmentation on the fly [219, 221]. Beyond optimizing chunking, some indexing approaches enrich chunks with broader context by prepending LLM-generated document summaries, contextual information or metadata [215, 222]. Similarly, [86] demonstrate that appending learned "corpus" embeddings to queries and documents can further improve retrieval. Other indexing-time techniques involve organizing chunks into higher-level data structures. For example,

[218]: Liu (2022), *LlamaIndex*

[219]: Zhong et al. (2025), *Mix-of-Granularity: Optimize the Chunking Granularity for Retrieval-Augmented Generation*

[15]: Reimers et al. (2019), *Sentence-BERT: Sentence Embeddings using Siamese BERT- Networks*

[40]: Robertson et al. (1994), *Okapi at TREC-3*

[218]: Liu (2022), *LlamaIndex*

[220]: Chase (2022), *LangChain*

[219]: Zhong et al. (2025), *Mix-of-Granularity: Optimize the Chunking Granularity for Retrieval-Augmented Generation*

[221]: Qian et al. (2024), *Grounding Language Model with Chunking-Free In-Context Retrieval*

[215]: Anthropic (2024), *Introducing Contextual Retrieval*

[222]: Poliakov et al. (2024), *Multi-Meta-RAG: Improving RAG for Multi-Hop Queries using Database Filtering with LLM-Extracted Metadata*

[86]: Morris et al. (2024), *Contextual Document Embeddings*

[223] and [224] cluster related chunks into semantic graphs or tree hierarchies.

Querying. In contrast, *query-time* solutions rely on iterative or agentic loops to refine retrieval dynamically. LLMs can be used to iteratively update the query or request additional chunks based on partial results [225, 226], or even to run “self-checks” and seek extra context when needed [227]. While these adaptive techniques can better address complex, multi-hop queries, they typically require much more computational resources during inference.

[223]: Edge et al. (2024), *From Local to Global: A Graph RAG Approach to Query-Focused Summarization*

[224]: Sarthi et al. (2024), *RAPTOR: Recursive Abstractive Processing for Tree-Organized Retrieval*

[225]: Xiong et al. (2021), *Answering Complex Open-Domain Questions with Multi-Hop Dense Retrieval*

[226]: Trivedi et al. (2023), *Interleaving Retrieval with Chain-of-Thought Reasoning for Knowledge-Intensive Multi-Step Questions*

[227]: Asai et al. (2023), *Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection*

6.3 ConTEB: Context-aware Text Embedding Benchmark

Table 6.1: Merged ConTEB dataset details. Controlled datasets are highlighted in **bold blue**. NanoBEIR values are summed over the 13 datasets that compose it.

	Dataset	Queries	Docs	Tokens per Chunk	Chunks per Document	Context Utilization
In Domain	MLDR	100	100	170.5	15.4	Document-level reasoning
	NarrativeQA	8575	355	154.5	4.9	Document-level reasoning
	SQuAD	2067	2067	19.1	8.5	Chunk not self-contained
Out of Domain	Football	2682	301	77.4	20.8	Co-reference resolution
	Geography	5283	530	113.6	4.3	Co-reference resolution
	Insurance	120	1	80.7	60.0	Structure understanding
	Covid-QA	1111	115	153.9	29.1	Chunk not self-contained
	ESG Reports	36	30	205.5	123.4	Context disambiguation
	NanoBEIR*	650	56723	199.4	1	No context is needed

6.3.1 Benchmark Design

Existing benchmarks often rely on (or assume) self-contained document chunks. This creates a misleading perception that contextualization offers little to no benefit, which in practice is rarely the case. To address this gap, the ConTEB benchmark philosophy is to explicitly be composed of tasks in which leveraging document-wide context should lead to performance improvements. Our benchmark originates from two sources: new datasets specifically created for ConTEB, and repurposed academic datasets. We take special care in selecting data sources spanning from multiple domains, including realistic industrial scenarios.

Why Context? Context can help resolve ambiguity, such as distinguishing between multiple meanings of a word or resolving pronouns and entity references (co-reference resolution). It is crucial when documents have a structured format, like legal or scientific texts, where understanding table of content hierarchy is key to aid intra-document disambiguation. Inversely, document-level contextual information is key when querying a corpus of documents that follow a strongly similar structure, such as annual company reports, to enable cross-document disambiguation.

Concept. To isolate the importance of contextual cues and diminish other confounding factors, we construct three benchmark tasks to study contextualization in controlled experimental settings [228]. We also

[228]: Allen-Zhu (2024), *ICML 2024 Tutorial: Physics of Language Models*

evaluate more practical retrieval settings at larger scale where we suspect contextualization to help, and in which we rely on organic, pre-existing query-document pairs.

6.3.2 Benchmark Construction

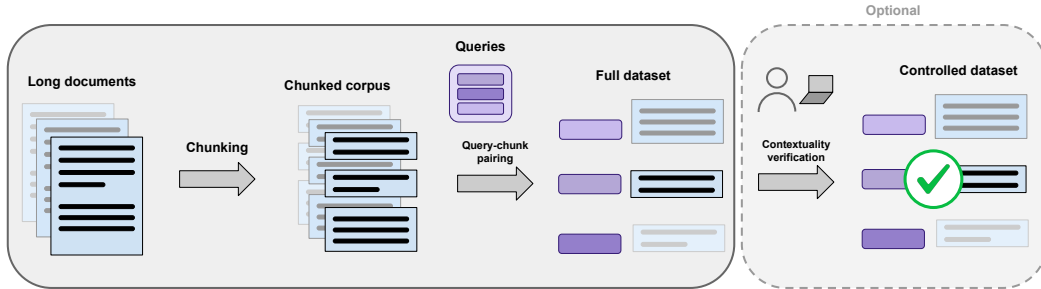


Figure 6.3: Benchmark creation process.

The generic dataset curation pipeline is depicted in Figure 6.3. This three-stage process allows us to obtain queries linked to chunks belonging to long documents that contain context relevant to each chunk. We apply this pipeline to each of our data sources, with minor adjustments depending on the existing data at hand, that we detail in section F.1.

1: Chunking. We select long documents spanning a variety of domains and chunk them through a structure-aware method³ [7, 128, 229–231].

2: Pairing. We use manual answer span annotations (*SQuAD*, *ESG*) or synthetically label them with an LLM (*CovidQA*, *MLDR*, *NarrativeQA*), to match queries with chunks obtained in Stage 1. This ensures queries are not solvable by design [141]. Alternatively, in our controlled experiment tasks, we generate queries pertaining to the chunks manually (*Insurance*) or synthetically using LLMs (*Football*, *Geography*).

3: Manual Verification & Modification. The manually crafted questions in *Insurance* are designed to be ambiguous without prior knowledge of the document structure. This is manually verified in this phase. Going a step further, in *Football* and *Geography*, we reformulate chunks with the help of an LLM to remove explicit mentions of the original document’s theme which all queries mention. We do so in all but the first chunks of each document, explicitly enforcing the need for context.

In addition to our contextual scenarios, we use *NanoBEIR* [65] to evaluate non-regression on standard non-contextualized embedding tasks.

By combining hard tasks in controlled environments, repurposed academic benchmarks, and real-world industrial queries, our benchmark provides a comprehensive assessment of retrieval models in both standard and context-dependent retrieval scenarios.

6.3.3 Training Dataset

Open training data is a key factor to ensure fair comparisons across methods and robust conclusions. In addition to our benchmark, we construct and release a training dataset composed of query and document chunk

3: RecursiveCharacterSplitter with a threshold of 1000 characters [220]

[7]: Macé et al. (2026), *ViDoRe Benchmark V2: Raising the Bar for Visual Retrieval*

[128]: Chen et al. (2024), *BGE M3-Embedding: Multi-Lingual, Multi-Functionality, Multi-Granularity Text Embeddings Through Self-Knowledge Distillation*

[229]: Rajpurkar et al. (2016), *SQuAD: 100,000+ Questions for Machine Comprehension of Text*

[230]: Möller et al. (2020), *COVID-QA: A Question Answering Dataset for COVID-19*

[231]: Kočiský et al. (2017), *The NarrativeQA Reading Comprehension Challenge*

[141]: Thakur et al. (2025), *FreshStack: Building Realistic Benchmarks for Evaluating Retrieval on Technical Documents*

[65]: Thakur et al. (2021), *BEIR: A Heterogenous Benchmark for Zero-shot Evaluation of Information Retrieval Models*

pairs. It includes the training splits of *MLDR* and *NarrativeQA*, repurposed with our previously detailed pipeline. To increase the number of queries, we further use *GPT-4o* to generate relevant supplementary synthetic queries. We also concatenate *SQuAD* chunks from the same Wikipedia article, keeping track of the original question-passage associations. The full dataset contains 9881 unique long documents (3698 tokens on average), corresponding to a total of 232'587 chunks and 307'241 queries (see subsection F.1.6). Scaling the dataset to more sources, through diverse synthetic augmentations and refinement-based augmentation methods [71, 73] is left for future work.

6.3.4 Baselines

Training-Free. We evaluate a selection of off-the-shelf methods that are strong in their size categories such as a standard single-vector embedding model based on ModernBERT (`modernbert-embed-large` [201, 232]), its multi-vector ColBERT equivalent [78, 233] and *Okapi BM25* [40], a strong lexical matching method. Additionally, we compare against various contextualization approaches. Specifically, we include Anthropic's contextual retrieval approach [215]⁴, and evaluate Late Chunking [85] without specific fine-tuning using `modernbert-embed-large`. These methods cover standard practices with varying levels of complexity and indexing budgets.⁵

Training-Based. For fair evaluation, we also fine-tune the sentence embedding method `modernbert-embed-large` on the training dataset with the same batch construction strategy as when training our main method, ensuring performance differences only stem from methodological design.

6.4 Training Contextual Embedders

In this work, we leverage recent advances in long-context embedding models [201, 210] to improve upon existing approaches through novel training strategies.

6.4.1 Architecture

Late Chunking. Late Chunking [85] (LC) is a training-free token pooling technique designed to enable information propagation across same-document chunks. Formally, given a document d split into chunks $\{c_1, \dots, c_{N_d}\}$, dense retrievers compute independent representations:

$$\phi(d) = [\phi(c_1), \phi(c_2), \dots, \phi(c_{N_d})]$$

In Late Chunking, chunks are concatenated and the whole sequence representation is computed in a single-forward pass:

$$H = \phi(c_1 \oplus c_2 \oplus \dots \oplus c_{N_d})$$

[71]: Wang et al. (2024), *Improving Text Embeddings with Large Language Models*

[73]: Lee et al. (2024), *NV-Embed: Improved Techniques for Training LLMs as Generalist Embedding Models*

[78]: Khattab et al. (2020), *ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT*

[233]: Chaffin (2025), *GTE-ModernColBERT*

[40]: Robertson et al. (1994), *Okapi at TREC-3*

[215]: Anthropic (2024), *Introducing Contextual Retrieval*

4: We use `Qwen-2.5-7B-Instruct` as the generative model which we serve on a 80GB A100 GPU with `vLLM` and `modernbert-embed-large` as the embedding model

[85]: Günther et al. (2024), *Late Chunking: Contextual Chunk Embeddings Using Long-Context Embedding Models*

5: We also evaluate *RAPTOR* [224] with `Qwen-2.5-7B-Instruct` and `cde-small-v2` [86] but find them to be poorly adapted to our problem settings.

[201]: Warner et al. (2024), *Smarter, Better, Faster, Longer: A Modern Bidirectional Encoder for Fast, Memory Efficient, and Long Context Finetuning and Inference*

[210]: Zhang et al. (2024), *mGTE: Generalized Long-Context Text Representation and Reranking Models for Multilingual Text Retrieval*

[85]: Günther et al. (2024), *Late Chunking: Contextual Chunk Embeddings Using Long-Context Embedding Models*

where $H = [h_1, h_2, \dots, h_T]$ consists of token-level representations. We then apply average pooling within each original chunk to obtain chunk-wise representations:

$$\phi_{LC}(c_i) = \frac{1}{|c_i|} \sum_{t \in c_i} h_t, \quad \forall i \in \{1, \dots, N_d\}$$

This allows each chunk representation to benefit from contextualization over the full document before aggregation.

Late Interaction. Late Interaction (LI) models [78, 128] are retrieval methods that do not pool token representations and instead store all token embeddings of each document. This approach boosts performance, especially on long-context retrieval tasks [201, 211], at the expense of storage cost. In this work, we propose extending Late Chunking approaches to LI models by applying standard LC but simply forgoing the final pooling and storing token embeddings depending on their original chunk memberships.

$$\phi_{LI}(c_i) = \{h_t : t \in c_i\}, \quad \forall i \in \{1, \dots, N_d\}$$

Setup. As the base single-vector embedding model for our experiments, we use modernbert-embed-large [232] (396M parameters), which is fine-tuned for retrieval tasks using the method from [207]. Respectively, we leverage GTE-ModernColBERT [233] (149M parameters) for our late interaction experiments. Both models are based on ModernBERT [201] which supports a context length of up to 8,192 tokens, significantly surpassing the 512-token limit of traditional BERT models, and thereby enabling the processing of longer documents in a memory efficient manner, which is critical to our method.

6.4.2 Learning Objective

Late Chunking enables information "leakage" between chunks of the same document. While this training-free method showed promises, we construct a learning objective to explicitly optimize contextual embedding models for this setting. Our aim is twofold: optimizing chunk representations to integrate relevant document-level information, all while ensuring they retain their specificity with respect to other same-document chunks, in order to prevent embedding collapse.

Previous works [64, 68, 95, 207, 234, 235] have relied on various learning objectives inspired by the contrastive learning literature [236]. A natural choice is the InfoNCE objective [192], which samples "negative" embeddings from other documents of the same batch.

In our approach, we combine these negatives with an auxiliary *in-sequence* contrastive loss, where chunks originating from the same document as the positive serve as hard negatives during training. Intuitively, training Late Chunking models contrastively with chunks from *different* documents encourages information propagation within each document and improves document identification. On the other hand, the contrastive term between same-document chunks ensures each chunk retains its specificity, and remains identifiable w.r.t. its neighbors. This aspect is further motivated

[78]: Khattab et al. (2020), *ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT*

[128]: Chen et al. (2024), *BGE M3-Embedding: Multi-Lingual, Multi-Functionality, Multi-Granularity Text Embeddings Through Self-Knowledge Distillation*

[201]: Warner et al. (2024), *Smarter, Better, Faster, Longer: A Modern Bidirectional Encoder for Fast, Memory Efficient, and Long Context Finetuning and Inference*

[211]: Zhu et al. (2024), *LongEmbed: Extending Embedding Models for Long Context Retrieval*

[232]: Chaffin (2025), *ModernBERT-embed-large*

[233]: Chaffin (2025), *GTE-ModernColBERT*

[201]: Warner et al. (2024), *Smarter, Better, Faster, Longer: A Modern Bidirectional Encoder for Fast, Memory Efficient, and Long Context Finetuning and Inference*

[64]: Karpukhin et al. (2020), *Dense Passage Retrieval for Open-Domain Question Answering*

[68]: Li et al. (2023), *Towards General Text Embeddings with Multi-stage Contrastive Learning*

[95]: Wang et al. (2022), *Text Embeddings by Weakly-Supervised Contrastive Pre-training*

[207]: Nussbaum et al. (2024), *Nomic Embed: Training a Reproducible Long Context Text Embedder*

[234]: Ni et al. (2021), *Large Dual Encoders Are Generalizable Retrievers*

[235]: Izacard et al. (2021), *Unsupervised Dense Information Retrieval with Contrastive Learning*

[236]: Schroff et al. (2015), *FaceNet: A Unified Embedding for Face Recognition and Clustering*

[192]: Oord et al. (2018), *Representation Learning with Contrastive Predictive Coding*

by the fact that in practice, queried corpora often contain negative documents stemming from the same source. Figure 6.2 illustrates chunk roles across a training batch.

Training Loss. To balance the contribution of in-sequence and in-batch negatives, we define the weighted InfoNCE loss as:

$$\mathcal{L} = \lambda_{\text{seq}} \mathcal{L}_{\text{seq}} + (1 - \lambda_{\text{seq}}) \mathcal{L}_{\text{batch}} \quad (6.1)$$

where $\lambda_{\text{seq}} \in [0, 1]$. Loss terms are defined as:

$$\mathcal{L}_{\text{seq}} = -\mathbb{E} \left[\log \frac{\exp(q \cdot k^+ / \tau)}{\sum_{k_i \in \mathcal{N}_{\text{seq}}} \exp(q \cdot k_i / \tau)} \right]$$

$$\mathcal{L}_{\text{batch}} = -\mathbb{E} \left[\log \frac{\exp(q \cdot k^+ / \tau)}{\sum_{k_j \in \mathcal{N}_{\text{batch}} \cup \{k^+\}} \exp(q \cdot k_j / \tau)} \right]$$

Here, q denotes the query representation, and k^+ is the gold chunk representation, which belongs to $\mathcal{N}_{\text{seq}}$, the set of chunks from the same sequence as k^+ . Temperature $\tau > 0$, and $\mathcal{N}_{\text{batch}}$ is the set of all in-batch samples that do not belong to $\mathcal{N}_{\text{seq}}$. This extends to late interaction models by replacing the dot product between query and chunk embeddings by ColBERT’s *MaxSim* between the multiple query and document token embeddings.

By tuning λ_{seq} , we can adjust the relative importance of in-sequence versus in-batch contrastive learning (Figure 6.4) resulting in our *InSeNT* method.

6.4.3 Model training

Our training strategy (*InSeNT*) is designed to be lightweight and to occur on top of capable pre-trained embedding models without degrading their capabilities. We use AdamW, a cosine decay learning rate scheduler with a 5% warm-up phase and a learning rate of $5e - 5$ and train for 2 epochs on our training dataset. Batches are constructed by sampling 4 long documents per device, retrieving all corresponding chunks and concatenating them with a separator token in between. As documents in our training set contain more than 20 chunks on average, which are themselves often linked to one or multiple queries, a batch contains more than 100 query-positive-negative triplets to learn on.⁶ A single epoch takes less than 1 H100 GPU hour.

6: In *MB+Training*, data is sampled the same way for fair evaluation but flattened in batch, corresponding to per-device batch sizes of more than 100.

6.5 Results

Document-wide context is essential. As seen in Table 6.2, methods leveraging contextual information widely outperform non-contextual methods across *ConTEB* tasks. These results highlight the critical role of context-aware embeddings in improving retrieval performance in such settings, whether through untrained Late Chunking approaches

Table 6.2: Evaluation (nDCG@10) of baseline models and our proposed method on *ConTEB*. Runtime is per-document indexing time in milliseconds; smaller is better, so the fastest model is bolded.

	In-Domain					Out-Of-Domain					
	Practical Settings					Controlled Settings					
	MLDR	SQuAD	NarrativeQA	COVID-QA	ESG Reports	Football	Geography	Insurance	Average	Runtime (ms/doc)	Non-Contextual NanoBEIR
Non-Contextual Models											
BM25	69.4	56.2	74.7	53.7	19.9	12.2	45.6	0.0	41.5	4.29	43.4
ModernBERT Large	78.4	73.4	77.9	61.7	36.8	19.1	56.2	12.4	52.0	17.83	63.2
ModernColBERT	83.5	74.2	80.4	78.2	44.2	30.2	68.5	16.1	59.4	14.99	67.7
ModernBERT Large + Training	78.7	74.0	77.3	55.2	20.0	22.9	58.7	13.9	50.1	16.44	54.5
Untrained Contextual Models											
Anthropic Contextual	85.4	77.1	77.7	60.7	34.8	53.9	89.4	100.0	72.4	1890.94	63.2
ModernBERT Large + Late Chunking	78.5	77.1	75.8	40.0	31.7	54.6	89.6	41.0	61.0	15.81	63.2
ModernColBERT + Late Chunking	84.1	75.7	80.7	75.5	44.4	31.3	67.9	13.2	59.1	7.41	67.7
Trained Contextual Models											
ModernBERT Large + InSeNT	88.7	80.9	81.3	56.0	43.1	63.9	90.7	100.0	75.6	15.26	60.4
ModernColBERT + InSeNT	90.1	75.1	83.5	67.7	48.3	64.6	89.8	45.9	70.6	7.57	59.2

or expensive context-aware reformulation approaches. As expected, the gap is even more notable in *ConTEB*’s controlled setting experiments.

Improving contextual information propagation. Our results clearly show that *InSeNT* variants outperform their untrained counterpart (+14.6 nDCG@10 for ModernBERT, +11.5 for ModernColBERT). Importantly, this is not due to the nature of the training data itself; the non-contextual ModernBERT model trained on the same data (ModernBERT + Training) does not improve upon the untrained baseline. Furthermore, the tasks that display the biggest improvements are the controlled setting tasks *Insurance*, *Football*, that are explicitly designed to elicit information given in previous paragraphs, and that are out-of-domain w.r.t. our training set.

Late Interaction. Interestingly, while LI models are good at long-context retrieving, they are poorly suited to out-of-the-box late chunking (-0.3 nDCG@10 w.r.t. ModernColBERT without LI). We posit that since token embeddings are never pooled, these models learn very local features and cannot leverage information from neighboring tokens. Once trained with our method, ModernColBERT+*InSeNT* displays large performance gains across the board (+11.5 nDCG@10 w.r.t. ModernColBERT + Late Chunking), showcasing an increased ability to leverage external context.

Context can add noise. The *CovidQA* task sticks out from the rest as untrained late chunking approaches severely degrade performance. Qualitative analysis, as well as the strong performance of the non-contextualized ModernColBERT method, indicate that the query-chunk pairing are often very extractive and match on technical medical terms, thus rendering context less useful. Our results show that naively applying late chunking in this setting adds noise and leads to notable performance drops (-21 nDCG@10), which are in large part recovered through our training method (+16 nDCG@10).

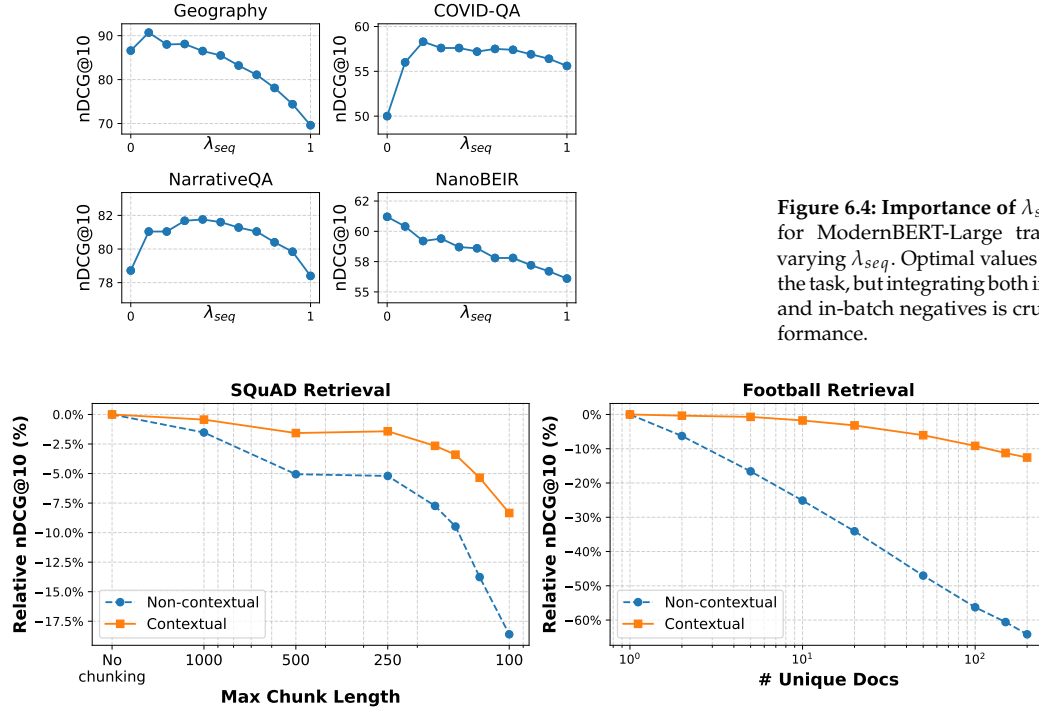


Figure 6.4: Importance of λ_{seq} : Results for ModernBERT-Large trained with varying λ_{seq} . Optimal values depend on the task, but integrating both in-sequence and in-batch negatives is crucial to performance.

Figure 6.5: Contextualized models trained with InSeNT are more robust to aggressive chunking strategies that remove essential information from chunks (left), and scale better with corpus size and ambiguity (right).

λ_{seq} **matters.** The training objectives are to induce chunk representations to integrate document-level information (role of *in-batch* negatives) while maintaining their specificity with respect to other same document chunks (role of *in-sequence* negatives). By varying λ_{seq} from Equation 6.1, we weight the importance of both objectives.

After training a series of models with varying λ_{seq} , we see on Figure 6.4 that training with only in-sequence or in-batch negatives yields the worse results, and the optimal λ_{seq} varies depending on the task. When documents need to be disambiguated between one another (*NanoBEIR*, *Geography*), up-weighting in-batch negatives seems optimal. On tasks where the challenge lies in locating information within a given document (*NarrativeQA*, *Covid-QA*), in-sequence negatives play a large role, but still need to be combined to in-batch negatives. Striking the optimal trade-off is thus very use-case dependent, and we opt for $\lambda_{seq} = 0.1$ after tuning on the validation split of our training dataset.

Efficiency-Performance. As shown in the Runtime column of Table 6.2, our approach is very capable on contextual tasks, yet does not add much computational overhead. In fact, we find slight indexing speed improvements, attributed to our approach’s reduced need for padding in-batch sequences of different lengths. While *Anthropic Contextual* achieves similar performance on *ConTEB*, it relies on costly LLM-based summarization and chunk reformulation that are hardly scalable to huge corpora (120x slower).

Short-Context Performance. Careful hyperparameter tuning enables our best model to maintain strong performance on standard non-contextual benchmarks (*NanoBEIR*), demonstrating that long-context optimization does not compromise short-context retrieval. Interestingly, LI models

[237]: Li et al. (2024), *Improving General Text Embedding Model: Tackling Task Conflict and Data Imbalance through Model Merging*

[238]: Wang et al. (2025), *LiNeS: Post-training Layer Scaling Prevents Forgetting and Enhances Model Merging*

suffer from more degradation, which we posit is due to the original reliance on very local features modified through our training. Mixing in non-contextual "replay" data during training or merging models [237, 238] should further enable preserving the original embedding model's performance.

6.6 Ablations

Robustness to chunking. We assess our method's robustness to poor chunking strategies using *SQuAD* annotations. Each originally self-contained chunk is split into multiple progressively smaller sub-chunks while we keep track of the annotated answer span to identify the gold chunk. Eventually, these sub-chunks become too small to be self-contained and end up lacking sufficient information to be relevantly embedded on their own. Figure 6.5 (left) demonstrates that contextual embeddings greatly improve robustness w.r.t. suboptimal chunking. The model is able to elicit information from neighboring chunks to integrate contextual information within smaller sub-chunks, leading to a much more uniform retrieval performance across a wide range of chunk sizes.

Robustness to corpus size. Common in the industry are templated documents that differ mostly by a key aspect (year, company name) but contain otherwise very similar information. We study the dynamics of retrieval performance w.r.t. the amount of similar documents in the corpus by computing scaling laws in which we iteratively vary the number of unique documents (composed of multiple chunks) in the corpus. We observe in Figure 6.5 (right) that contextual embeddings scale vastly differently than their independently embedded counterpart. Intuitively, the greater the amount of similar documents and chunks in the corpus, the harder it is for a retrieval system to match the correct ones, but when embedding models are able to leverage external context, this effect is attenuated.

Information Propagation. We experiment with concatenating semantically similar yet independent short chunks as "artificial" long documents. The resulting model is contextual as it uses late chunking, but exhibits performance in line with non-contextual baselines (*ModernBERT Large + Training*). We posit training on arbitrarily concatenated chunks, which by design are not contextually linked, teaches the model not to use information from neighboring chunks. This highlights the necessity of sourcing organic long-context data during training to induce correct training dynamics. Details in Table F.2 in section F.3.

6.7 Conclusions

In this work, we introduced *ConTEB*, a benchmark designed to assess the effectiveness of retrieval models in leveraging document-wide contextual information. Our evaluation demonstrates that standard retrieval models struggle in context-dependent settings, while our proposed approach *InSeNT*, which combines Late Chunking and a novel training methodology, performs strongly on *ConTEB* without additional compute costs. These

insights build towards the broader development of the *Late Chunking* paradigm in practice, and highlight the everlasting need for evaluation benchmarks that rigorously reflect how embedding models are used in real-world scenarios.

Future Work. Scaling our approach with recent decoder models with extended context lengths (e.g., 1M+ tokens [181]) would enable embedding entire books or lengthy documents in a single forward pass, potentially unlocking new capabilities for large-scale document retrieval. It would also be interesting to observe the impact of our method on retrieval confidence [18]. Finally, adapting our method to multi-modal embedding pipelines that have less control over the chunking strategy could further enhance retrieval systems in industrial applications with visually rich contextual documents [6, 153].

Limitations

While our approach enhances retrieval performance in context-dependent settings, limitations persist.

Context Length. Our method is applied to long-context encoders that currently support sequences of up to 8k tokens. While we have shown performance can extrapolate to sequences of up to 32k tokens, scaling this approach to handle 1M+ token contexts with decoder-based models would be an interesting research avenue and presents significant compute and memory challenges. It notably requires rethinking the data construction processes to ensure longer documents are effectively leveraged.

Non-contextual Performance. While our approach unlocks previously unattainable performance in contextual scenarios, it can come at the cost of slight short-context performance degradation. The optimal trade-off between non-contextual and contextual retrieval performance is highly usecase dependent Figure 6.4 and can be parametrized by practitioners using the λ_{seq} parameter. Other approaches may be promising such as model merging [237, 238].

Data Generation. The creation of training and evaluation data relies on existing datasets and semi-synthetic generation pipelines. However, a fully automated and scalable method for generating high-quality queries that effectively induce non-trivial context utilization remains an open challenge.

Evaluation. While our model demonstrates strong cross-domain performance, further validation in real-world applications, various use cases, and multiple languages is necessary to further assess its robustness and generalizability.

[181]: Yang et al. (2025), *Qwen2.5-1M Technical Report*

[18]: Gisserot-Boukhlef et al. (2024), *Towards Trustworthy Reranking: A Simple yet Effective Abstention Mechanism*

[6]: Faysse et al. (2025), *ColPali: Efficient Document Retrieval with Vision Language Models*

[153]: Ma et al. (2024), *Unifying Multi-modal Retrieval via Document Screenshot Embedding*

[237]: Li et al. (2024), *Improving General Text Embedding Model: Tackling Task Conflict and Data Imbalance through Model Merging*

[238]: Wang et al. (2025), *LiNeS: Post-training Layer Scaling Prevents Forgetting and Enhances Model Merging*

CroissantLLM: A Truly Bilingual French-English Language Model

7

Paper Context

Initiated in 2023 and released in January 2024 (TMLR 2025), the CroissantLLM initiative [12] introduced a compact bilingual French-English language model trained on a balanced mixture of French and English data. Agentic search has so far largely benefited from highly capable LLM orchestrators —whose stronger reasoning abilities and broader background knowledge enable more precise tool use. State-of-the-art systems have thus historically relied on large proprietary models predominantly pretrained on English. CroissantLLM challenges this trend by showing that smaller models, trained openly and transparently, can achieve strong performance while providing valuable resources for both the academic and industrial communities.

This project was among the first open efforts to highlight the benefits of training with high token-to-parameter ratios, and among the early attempts to train sub-7B models *outside* of a larger scaling ladder with the explicit goal of delivering strong performance at small size for efficient inference. At release time, CroissantLLM achieved state-of-the-art results on French language benchmarks and matched the strongest English models in its size category. It also delivered particularly strong performance on English-French translation, often surpassing models that were orders of magnitude larger. Beyond the model itself, the CroissantLLM project emphasizes the importance of bilingual training corpora, establishes joint scaling laws for bilingual settings, and is accompanied by substantial data work: the release of high-quality French resources reused in many subsequent projects, as well as new French benchmarks. Models, code, datasets, and evaluation suites are released publicly, supporting continued progress toward strong local models that run efficiently on consumer hardware.

While integrating this project as part of the framework of this thesis might seem (and is) a bit *post-hoc*, some results are particularly relevant to the central theme of information recall through associative links. In this setting, the memory component is parametric (model weights) rather than external (a database), but our results clearly show information recall and domain knowledge acquisition (French culture) to be impacted by the training data distribution, even in such internet scale training runs. Training through backpropagation becomes the mechanism by which associative links are created and leveraged during the forward pass. If LLMs were used as search agents, being ingrained with strong parametric priors on the search theme would yield a more efficient information retrieval process. It opens up an interesting scientific question on the importance of parametric knowledge to correctly reason about practical topics and puts into question efforts to train LLMs without any knowledge.

7.1	Introduction	80
7.1.1	Contributions	81
7.2	Data	83
7.2.1	French Data	83
7.2.2	English Data	85
7.2.3	Code Data	85
7.2.4	Parallel Data	86
7.3	Training	86
7.3.1	Model Architecture	86
7.3.2	Tokenizer	87
7.3.3	Selecting an optimal language ratio	88
7.3.4	Final data distribution	89
7.3.5	Training framework	89
7.3.6	Training losses	90
7.3.7	Environmental impact	90
7.4	Evaluation Benchmarks	91
7.4.1	English	91
7.4.2	French	91
7.4.3	Other Tasks	93
7.5	Benchmark results	93
7.5.1	Base model	94
7.5.2	Finetuning	96
7.5.3	Optimized Inference	97
7.5.4	Model limitations	99
7.6	Foundation Model Transparency Index	99
7.7	Post-Mortem & Lessons learned	100

Paper Abstract

We introduce CroissantLLM, a 1.3B language model pre-trained on a set of 3T English and French tokens, to bring to the research and industrial community a high-performance, fully open-sourced bilingual model that runs swiftly on consumer-grade local hardware. To that end, we pioneer the approach of training an intrinsically bilingual model with a 1:1 English-to-French pretraining data ratio, a custom tokenizer, and bilingual finetuning datasets. We release the training dataset, notably containing a French split with manually curated, high-quality, and varied data sources. To assess performance outside of English, we craft a novel benchmark, FrenchBench, consisting of an array of classification and generation tasks, covering various orthogonal aspects of model performance in the French Language. Additionally, rooted in transparency and to foster further Large Language Model research, we release codebases, and dozens of checkpoints across various model sizes, training data distributions, and training steps, as well as fine-tuned Chat models, and strong translation models. We evaluate our model through the FMTI framework and validate 81 % of the transparency criteria, far beyond the scores of even most open initiatives. This work enriches the NLP landscape, breaking away from previous English-centric work to strengthen our understanding of multilingualism in language models.

[65]: Thakur et al. (2021), *BEIR: A Heterogenous Benchmark for Zero-shot Evaluation of Information Retrieval Models*

[66]: Muennighoff et al. (2022), *MTEB: Massive Text Embedding Benchmark*

[239]: OpenAI (2023), *GPT-4 Technical Report*

[178]: Touvron et al. (2023), *Llama 2: Open Foundation and Fine-Tuned Chat Models*

[240]: Touvron et al. (2023), *Llama: Open and efficient foundation language models*

[241]: Bai et al. (2023), *Qwen Technical Report*

[136]: Jiang et al. (2023), *Mistral 7B*

[242]: Jiang et al. (2024), *Mixtral of Experts*

[243]: Bommasani et al. (2023), *The Foundation Model Transparency Index*

[244]: Casper et al. (2024), *Black-Box Access is Insufficient for Rigorous AI Audits*

[245]: Hubinger et al. (2024), *Sleeper Agents: Training Deceptive LLMs that Persist Through Safety Training*

1: <https://purl.stanford.edu/kh752sm9123>

[246]: NewYorkTimes (2023), *The Times sues OpenAI and Microsoft over A.I. use of copyrighted work*

[247]: Samuelson (2023), *Generative AI meets copyright*

2: <https://www.commerce.senate.gov/2023/9/the-need-for-transparency-in-artificial-intelligence>

7.1 Introduction

Although a few proprietary models [65, 66] are still considered to run ahead of the pack [239], open weights models such as Llama [178, 240], Qwen [241] or Mistral [136, 242] are rapidly bridging the performance gap. However, widespread industrial and research adoption of such technology remains challenging for several reasons, including the lack of transparency in the data collection and training processes, the scarcity of existing resources outside of English, and the large-scale and costly nature of existing high-performing models.

Lack of transparency. State-of-the-art models, both proprietary and open-weights are often designed and trained by heavily investor-backed companies, that aim to retain a moat by keeping their training data mix and strategy secret, hindering the rest of the field’s ability to fully study and understand these models. This lack of transparency, ranging from training set composition to lack of evaluation or unclear usage policies, has been characterized by previous works, such as those by [243] and [244], pushing for full transparency as a key component for safe model development and use. The dangers of closed, non-auditable datasets have been exemplified by recent findings showcasing the potential dangers of dataset contamination, whether intentional [245] or not.¹ Furthermore, legal questions arise surrounding data ownership in LLM training corpora [246, 247] and recent developments in the political landscape, regarding AI (EU AI Act, US senate hearings)² have further emphasized the importance of transparent approaches, both from a legal perspective and to build user trust.

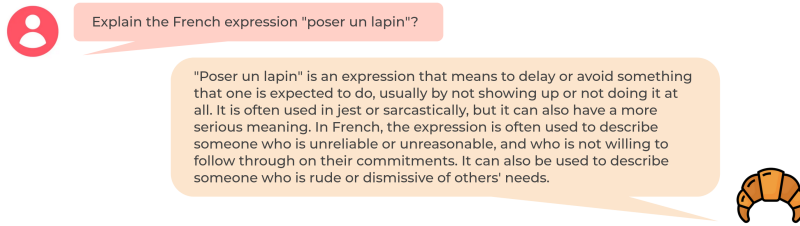


Figure 7.1: Conversation example with CroissantLLMChat

Bias towards English. Although the exact training mix of most well-performing LLMs is not publicly available information, most large models are trained on very English-centric corpora [240]. This is the consequence of the important amount of English resources compared to other languages, both in terms of data availability and benchmarks. As could be expected, all publicly available LLMs display a large performance disparity between English and non-English languages when evaluated on downstream tasks [248]. Moreover, cultural knowledge and biases are mechanically integrated through the training data, leading to models with greater knowledge of American events or biases [249, 250]. This puts non-English users at a disadvantage when it comes to language model usage and adoption. While the non-English NLP community has produced multilingual datasets [251, 252] and models [253, 254] in the last few years, the available resources still largely lag behind English ones, hindering industrial adoption in non-English settings.

Challenging to use at scale. Although benefits of scaling models to enormous sizes have been amply demonstrated in terms of performance [172, 255, 256], scale comes at a large cost in terms of hardware requirements and inference speed. Download statistics on HuggingFace show the smallest Llama model [240] to be the most adopted by the community, demonstrating the interest in small but capable models. LLM scaling laws demonstrate the diminishing returns of training a model of a given size past a certain amount of tokens. By continuing pre-training way past the compute-optimal threshold, performance has been shown not to saturate, enabling the training of “inference-optimal” models of high interest to the industrial and research communities [257]. It is still not fully understood how model performance continues to improve at these later stages of training.

7.1.1 Contributions

In this work, we attempt to bridge the aforementioned limitations and gaps. Our main contributions can be summarized as follows.

Contribution 1: Introduction of a highly-curated, diverse corpus in French. We collect and openly release a 303B token corpus spanning internet data, but also literary work, speech transcripts, legal and administrative documents, scientific articles, business-related documents, etc. The corpus is distributed under permissive licenses, allowing commercial use with no restriction, and is heavily filtered, curated, and deduplicated. To our knowledge, it is the largest multi-source French language corpus released to date of sufficient quality for language modeling purposes³.

Contribution 2: Training CroissantLLM, a truly bilingual language model. Nowadays, most models display some multilingual capabilities.

[248]: Bandarkar et al. (2023), *The Belebele Benchmark: a Parallel Reading Comprehension Dataset in 122 Language Variants*

[249]: Bender et al. (2021), *On the dangers of stochastic parrots: Can language models be too big?*

[250]: Santurkar et al. (2023), *Whose Opinions Do Language Models Reflect?*

[251]: Soboleva et al. (2023), *SlimPajama: A 627B token cleaned and deduplicated version of RedPajama*

[252]: Laurençon et al. (2023), *The BigScience ROOTS Corpus: A 1.6TB Composite Multilingual Dataset*

[253]: Scao et al. (2022), *Bloom: A 176b-parameter open-access multilingual language model*

[254]: Shliazhko et al. (2022), *mGPT: Few-Shot Learners Go Multilingual*

[172]: Wei et al. (2022), *Emergent Abilities of Large Language Models*

[255]: Hoffmann et al. (2022), *Training Compute-Optimal Large Language Models*

[256]: Chowdhery et al. (2022), *PaLM: Scaling Language Modeling with Pathways*

[257]: Sardana et al. (2023), *Beyond Chinchilla-Optimal: Accounting for Inference in Language Model Scaling Laws*

3: By this, we imply of sufficient quality to train a language model (little OCR errors, high-quality text) and not exclusively composed of Internet data.

[253]: Scao et al. (2022), *Bloom: A 176b-parameter open-access multilingual language model*

4: Qwen-VL [110] reports 77.3% & 22.7% Chinese, but no information is given for the Qwen base model.

[258]: Zeng et al. (2022), *Glm-130b: An open bilingual pre-trained model*

For example, Bloom has been trained to be massively multilingual [253], Llama contains a minor proportion of non-English data in its training set [240] and Qwen included a significative portion of Chinese data [241].⁴ However, to our knowledge, outside of Chinese with a different alphabet [258], no work has studied or attempted to train a multilingual model of significant scale in which English is not the dominant training language.

In this work, we train a model on a 1:1 ratio of English to French with a tokenizer optimized for bilingualism. Our end goal is to have a model less skewed towards English performance or cultural biases. We motivate this ratio by conducting careful experimentation on smaller model scales to uncover the trade-offs behind language distribution ratios. We opt for a strategy enabling a “best of both worlds” performance in these two languages, as empirically validated by scaling law experiments. Experimentally, we show the importance of integrating data from another cultural source in the acquisition of cultural knowledge, underlining the importance of the effort.

Contribution 3: FrenchBench: A novel LLM benchmark for the French Language. To evaluate models in French, we introduce a benchmark encompassing various tasks to assess factual knowledge, generative capabilities, language understanding, etc. This benchmark is constructed both from openly available datasets, as well as newly released manually annotated data. We evaluate and report results for our models as well as other models with French-speaking capabilities.⁵⁶

Contribution 4: Releasing high-performing, inference-optimal models for the industrial community, together with a wide range of resources for the research community. The models we train are all released under open licenses. Our largest model is trained on a 2300:1 token to parameter ratio (115 times longer than a Chinchilla Optimal 1.3B model) leading to very strong performance in its size category. We show that model performance on downstream tasks continues to dramatically improve with lengthened pre-training runs, although model perplexity does not significantly improve. We release all checkpoints for all model sizes, as well as the exact training data seen at every step for research purposes.⁷ These models are extremely efficient to run, leading to low-latency, energy-efficient on-edge inference, even on low-resource devices such as phones or personal computers. These releases⁸ are motivated by a commitment to transparency to allow research and reassuring users for industrial deployment: our models comply with 81% of criteria listed by the Foundation Model Transparency Index [243] (see Section 7.6).

The CroissantLLM initiative aims to adress the aforementioned limitations of current models through the release of an inference-optimized, small but capable model that performs well outside of English settings, and that is designed to be as open and transparent as possible. Beyond facilitating industrial LLM adoption and unlocking new generative model use cases, this project is also a proven platform for researching LLMs and kickstarting future pretraining efforts [20, 24, 25, 260].

5: Code for evaluation is available at <https://github.com/EleutherAI/lm-evaluation-harness>

6: Another complementary initiative has been led for French model evaluation and released concurrently in [259]

7: Training data indexing order will be released in a second stage.

8: Code for dataset collection and filtering is available at <https://github.com/ManuelFay/llm-data-hub>. Code for model training is hosted at <https://github.com/CoderPat/croissant-llm-training>. Datasets and model checkpoints are available at <https://huggingface.co/CroissantLLM>.

[243]: Bommasani et al. (2023), *The Foundation Model Transparency Index*

Table 7.1: Final Data mix for CroissantLLM training

	Size (GB)	Docs. (M)	Tokens (B)	Token/Doc	Sampling Ratio	# tokens (B)
French	1258.70	376.27	303.51	806.63	4.09	1240.08
English	2351.13	591.23	655.64	1108.94	1.89	1240.09
Code	366.87	81.90	141.43	1726.76	2.04	288.92
Parallel	113.91	408.03	35.78	87.68	6.13	219.26
Total	4090.61	1457.43	1136.35	779.70	14.15	2988.35

7.2 Data

The dominance of the English language in the training data of most current models is undeniable. While multilingual models like Llama leverage some non-English data [240], it corresponds to only a minor part of the corpus, leading to a significant drop in performance across non-English data, with noticeable “American” bias [250, 261]. This work aims to offset this trend by using a more balanced bilingual corpus comprising English and French, as well as additional code data. Although both languages belong to the Indo-European language family, they exhibit different morpho-syntactic structures⁹ and French has a richer morphology.¹⁰ We study whether this corpus helps in reducing biases, enabling more varied knowledge sets, and unlocking non-English performance.

A variety of sources are integrated into our corpus, including carefully filtered internet data and high-quality data from a range of sources, all devoid of restrictive licenses ensuring complete openness of the data and the trained model. Data statistics are available in Table 7.1. As further detailed in 7.3.4, our data corpus contains different amounts of unique English, French, and Code tokens. We obtain our balanced training corpus by upsampling French, Code, and English data with different sampling ratios, such that no performance loss is to be expected [262].

The scrapping and processing code are available in our code base.¹¹ The license information of all datasets used is given, all allowing for permissive commercial use.

7.2.1 French Data

Table G.1 lists the source and some information regarding the French corpus. Details about the data sources are expanded further below.

Web Data. We collect internet data from various web scraps (Oscar [263], mC4 [264]), leveraging the CulturaX corpus [265] for heuristic and perplexity filtering, as well as exact and fuzzy deduplication. In total, this represents over 363 million webpages and more than 292 billion tokens, that we split using our custom tokenizer fitted on equal amounts of French and English data.¹²

We ensure data is of good quality and correctly tagged in French through sampled manual inspection and confirm French-speaking countries are well represented within the dataset. Notably, we include several news

[250]: Santurkar et al. (2023), *Whose Opinions Do Language Models Reflect?*

[261]: Navigli et al. (2023), *Biases in Large Language Models: Origins, Inventory, and Discussion*

9: For example, pronominal objects are placed before (resp. after) the verb in French (resp. English), both languages have different noun phrase constructions (“la commission européenne” vs. “the European commission”), etc.

10: For example, English has 5 verb forms whereas French has 48, French has explicit inflections for grammatical genders, etc. However, note that only English adjectives have morphological constructions for expressing comparison (e.g. easy, easier, easiest). We refer to WALS for more details, e.g. <https://wals.info/feature/21B#2/26.7/152.4>

[262]: Muennighoff et al. (2023), *Scaling Data-Constrained Language Models*

11: <https://github.com/ManuelFay/llm-data-hub>

[263]: Abadji et al. (2022), *Towards a Cleaner Document-Oriented Multilingual Crawled Corpus*

[264]: Xue et al. (2021), *mT5: A massively multilingual pre-trained text-to-text transformer*

[265]: Nguyen et al. (2023), *CulturaX: A Cleaned, Enormous, and Multilingual Dataset for Large Language Models in 167 Languages*

12: The mC4 corpus <https://huggingface.co/datasets/allenai/c4> is released under the ODC-BY licence <https://opendatacommons.org/licenses/by/1.0/> whereas Oscar [263] does not claim data ownership, provides an opt-out strategy for data inclusion, and filtering metadata is released under the Creative Commons CC0 license <https://creativecommons.org/publicdomain/zero/1.0/>

13: Contrary to certain languages such as Arabic or Portuguese in which the language differs greatly depending on the country, the written French language in these countries is largely similar with only a minor share of regionally introduced syntax and expressions

14: Data is released at <https://echanges.dila.gouv.fr/OPENDATA/> with the ETALAB open license <https://www.etalab.gouv.fr/wp-content/uploads/2017/04/ETALAB-Licence-Ouverte-v2.0.pdf>

[266]: Hart (1971), *Project gutenber*

15: Metadata is licensed under Open Etalab license <https://gallica.bnf.fr/eadit/und/conditions-dutilisation-des-contenus-de-gallica>

16: <https://github.com/kpu/kenlm>

17: <https://www.poesie-francaise.fr/>

18: <https://opus.nlpl.eu/OpenSubtitles2016.php>

19: <https://www.opensubtitles.org>

[253]: Scao et al. (2022), *Bloom: A 176b-parameter open-access multilingual language model*

20: Licensed under CC BY-SA 4.0, https://en.wikisource.org/wiki/Wikisource:Copyright_policy

21: <https://huggingface.co/datasets/wikimedia/wikipedia> with a CC-BY-SA 3.0 license

22: Data is public internet data that we release under MIT license with an opt-out scheme https://huggingface.co/datasets/manu/illuin_layout_dataset_text_only

sources scraped from Belgium, Switzerland, Canada, and Lebanon, as well as multiple African countries (Senegal, Morocco, Algeria, Cameroon, etc.)¹³

Legal and Administrative Data. We introduce 5.3B tokens of data from the French government’s open data initiative, ranging from legal tasks to parliamentary discussions and financial transcripts (*e.g.* legal and administrative jargon). These texts originate from 13 different datasets (the OpenData corpus) and were collected from the French government’s open data platform.¹⁴ To ensure other French-speaking countries are represented, we add 68M tokens of data from Swiss legislation retrieved from government sources. We perform steps to process, filter, and run exact deduplication on these documents.

Cultural Data. We introduce cultural data from various sources. Notably, we retrieve all Project Gutenberg [266] books in the French language as of October 2023, corresponding to books released in the public domain (302 million tokens). We also download and aggressively filter manuscripts and documents from the French National Library (Bibliothèque Nationale de France), and filter for documents that belong to the public domain, have undergone an OCR process, and are of high quality.¹⁵ To filter out low-quality OCR documents, we implement custom heuristics which we release within our code base. We run all documents through perplexity filters using KenLM 5-grams¹⁶ fitted on the French Wikipedia split, and discard documents with perplexity values that are too high (noisy) or too low (repetitive patterns). Thresholds are set through a manual verification process. We deliberately choose to be aggressive in our filtering to ensure only high-quality data is kept and discard the largest portion of the original corpus, keeping about 27M tokens. We choose not to keep any data from the newspaper archives, as the OCR transcription is often too noisy. Additionally, we introduce famous public domain French poems custom scraped from a French poetry website,¹⁷ and run a set of podcasts through a high-quality speech-to-text model to obtain a textual transcription. This process is hard to scale and data splits from these sources are limited in quantity. Data from the OpenSubtitles¹⁸ initiative is integrated, corresponding to 41.8 million tokens originating from movie subtitles.¹⁹ Finally, we add the French data from Wikisource collected as part of the BigScience initiative [253] and obtain 2.7 billion tokens from the process.²⁰

Encyclopedia Data. To introduce high-quality factual data to the training corpus, we integrate the French Wikipedia split from November 2023.²¹ This corresponds to the latest knowledge cutoff in the training data. In total, more than 2.5 million articles are used, spanning more than 2 billion tokens.

Industrial Data. We scrap high-quality and publicly available data from industrial PDFs via a manually crafted list of websites, from large French and French Canadian (Quebec) companies to government agencies. This business-focused data boosts performance on a series of downstream applications related to industrial NLP. We collect over 36000 PDF multi-page documents and filter them through carefully crafted heuristics, followed by aggressive perplexity filtering.²² In total, we obtain over 290000 documents and 191 million tokens.

7.2.2 English Data

Our English data is primarily drawn from the SlimPajama corpus [251], excluding copyrighted documents. Splits per data source are detailed in Table G.2.

Internet Data. Similarly to the French dataset, we rely on carefully filtered content from an assortment of internet sources, including miscellaneous web pages and blogs. The filtering process includes heuristics and perplexity filtering, as well as large-scale deduplication [251]. The SlimPajama corpus includes internet data from the CommonCrawl²³ and C4²⁴ web scraps, as well as data sourced from Github textual content²⁵ and the StackExchange forums.²⁶

Miscellaneous. Other non-internet-based data sources are included in the SlimPajama dataset, such as scientific articles from Arxiv²⁷ and English documents from Wikipedia.²⁸ The SlimPajama dataset also comprises the “Books” subcorpus, obtained by downloading all book documents from Bibliotik.²⁹ Some of the documents within this last corpus have been flagged by their owner as proprietary data. We filter out all documents from this subcorpus, and replace them with data from the open-source Project Gutenberg [266] English books under public domains.³⁰

Gutenberg Canaries. To assess model memorization to inform about the risks of including private or sensitive data within the training set, we stress test the model by including “canaries” [267]. These correspond to samples that have been intentionally modified and/or repeated and included within the model training set, and that will enable a posteriori evaluation of the model capacity to memorize data in a “worse than worst-case” situation.³¹ In total the canaries represent 555 million tokens, representing less than 0.04 % of the total tokens seen during training.

7.2.3 Code Data

In line with most recent models [240, 253, 256], we integrate code data into our training corpus. Notably, previous work shows that code data benefits natural language tasks and can be particularly useful in data-constrained settings [262]. Therefore, we include 140B tokens of code data in several common programming languages. Splits and numbers of tokens are detailed in Table G.3.

The Stack & StarCoder. We rely on the efforts of the StarCoder project [268], and use their high-quality filtered code data from The Stack corpus [269].³² We keep only high-resource programming languages (Java, Javascript, Python, C, C++, SQL) and Jupyter notebooks, as well as a few samples of formatted data (JSON, Tex) and scripting languages (shell scripts, Dockerfiles).

Extra Python code. We extend the corpus with several other sources of Python code due to the popularity of the language in the community. Firstly, we add Pypi packages from recent code dumps, that are filtered to keep only Python and Jupyter files.³³ Secondly, in order to promote high-quality problem-solving-centered code, we integrate 1.2B tokens of Python3 data from competitive coding contests [270] under CC-By-4.0 license. Lastly, following the success of learning from textbooks [271], we

[251]: Soboleva et al. (2023), *SlimPajama: A 627B token cleaned and deduplicated version of RedPajama*

[251]: Soboleva et al. (2023), *SlimPajama: A 627B token cleaned and deduplicated version of RedPajama*

23: Common Crawl license <https://commoncrawl.org/terms-of-use>

24: ODC-BY license

25: Only content under MIT, BSD, and Apache licenses are kept

26: CC-By-SA 4.0 license <https://archive.org/details/stackexchange>

27: <https://arxiv.org/>, with author opt-out options

28: CC-By-SA 3.0 license

29: https://huggingface.co/datasets/the_pile_books3

[266]: Hart (1971), *Project gutenberg*

30: From <https://huggingface.co/datasets/pg19> with an Apache 2.0 license

[267]: Carlini et al. (2019), *The Secret Sharer: Evaluating and Testing Unintended Memorization in Neural Networks*

31: This work is led in parallel to the CroissantLLM project and findings will be independently published.

[262]: Muennighoff et al. (2023), *Scaling Data-Constrained Language Models*

[268]: Li et al. (2023), *StarCoder: may the source be with you!*

[269]: Kocetkov et al. (2022), *The Stack: 3 TB of permissively licensed source code*

32: Data are drawn from various sources and fall under 193 different permissive licenses. We use the version 1.2 of the corpus, which has been filtered with respect to data owners opt-out option information.

33: https://huggingface.co/datasets/vikp/pypi_clean

[270]: Li et al. (2022), *Competition-level code generation with AlphaCode*

[271]: Li et al. (2023), *Textbooks Are All You Need II: phi-1.5 technical report*

34: <https://huggingface.co/datasets/codeparrot/github-jupyter-code-to-text> Apache 2.0 license

[272]: Anil et al. (2023), *PaLM 2 Technical Report*

[273]: Briakou et al. (2023), *Searching for Needles in a Haystack: On the Role of Incidental Bilingualism in PaLM’s Translation Capability*

[274]: Tiedemann (2012), *Parallel Data, Tools and Interfaces in OPUS*

35: Free license <https://opus.nlpl.eu/>

[275]: Ramírez-Sánchez et al. (2020), *BiFixer and Bicleaner: two open-source tools to clean your parallel data*

36: <https://github.com/bitextor/bifixer>

37: <https://github.com/bitextor/bicleaner>

[276]: Rei et al. (2022), *CometKiwi: IST-Unbabel 2022 Submission for the Quality Estimation Shared Task*

38: Data licensed under Etalab open license <https://www.etalab.gouv.fr/wp-content/uploads/2017/04/ETALA-B-Licence-Ouverte-v2.0.pdf>

39: <https://www.lacoccinelle.net>, with opt-out options for data owners and community-contributed translations under free-use license.

add commented Python code constructed by combining code and text cells from Jupyter Notebooks through the CodeParrot initiative.³⁴

7.2.4 Parallel Data

Following previous work [272], we incorporate vast quantities of parallel data, in our case high-quality English-French translation pairs, in order to improve the multilingual capabilities of the model [273].

Opus. We extract subsets of sentence pairs spanning multiple domains from the OPUS corpus [274].³⁵ Statistics are described in Table G.2. In total, we include 400 million parallel sentences and about 36 billion tokens. The data is filtered through a rigorous cleaning pipeline: (1) BiFixer [275]³⁶ is first used to remove duplicate data through fuzzy matching techniques; (2) BiCleaner³⁷ is then used to filter data using heuristic and perplexity filters; (3) finally, the state-of-the-art NMT quality estimator CometKiwi [276] is used to keep only top quality translation pairs.

Theses. To incorporate versatile academic and scientific language, we augment our dataset with French theses abstracts along with their author-generated English translations. This corresponds to 95000 documents and more than 80 million high-quality tokens.³⁸

Song Lyrics. Our dataset integrates song lyrics in both French and English, scraped from a specialized community-driven lyrics translation website.³⁹ As such, our model is trained with radically different linguistic styles (e.g. colloquialism), and the wide range of topics can help the model to capture cultural nuances. Lyrics have been translated by hand by the website community. With a total of 70k songs, we have built up a corpus of 53M tokens covering different periods (80s, 90s, 20s, etc.) and musical genres (rap, rock, jazz, etc.) To preserve colloquial expressions and cultural subtleties, we have not filtered song lyrics for explicit content. We validate the original language metadata of the songs through Google’s language-detection algorithm.

7.3 Training

Our main goal was to train a high-performing, yet resource-friendly bilingual model while optimizing performances across both languages. To focus on the specific challenges of the bilingual paradigm, we rely on previous work to motivate many of our design and hyperparameter choices [178].

7.3.1 Model Architecture

We use the Llama architecture [240], a decoder-based transformer, trained with rotary position encodings [277] and a context length of 2048. We construct 4 different model sizes by jointly scaling the number of attention heads, hidden size, and hidden layers. Table 7.2 summarizes the sizes of each model in the family.

[178]: Touvron et al. (2023), *Llama 2: Open Foundation and Fine-Tuned Chat Models*

[277]: Su et al. (2023), *RoFormer: Enhanced Transformer with Rotary Position Embedding*

Model	Params (M)	Layers	Hidden size	Inter. size	KV heads
XXS	100.7	6	1024	4096	8
XS	202.5	12	1024	4128	8
S	341.5	12	1536	4128	12
Base	1214.3	24	2048	5504	16

Table 7.2: Model information for scaling laws. Parameter count excludes embedding and output parameters.

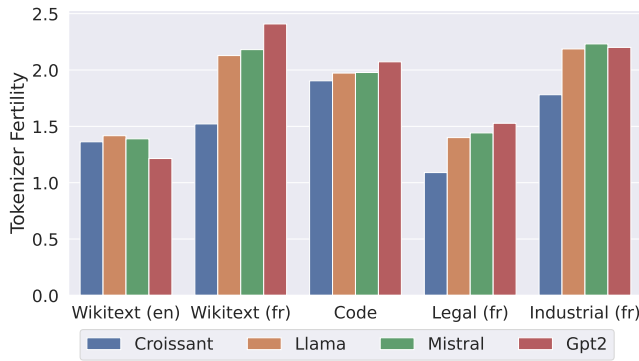


Figure 7.2: Fertility on unseen test sets using various tokenizers. Lower is better.

7.3.2 Tokenizer

Most LLM tokenizers are fitted on English-centric corpora with an information-theoretic optimization objective, for example, Byte-Pair encoding [278] or Unigram [279], leading to good fertility⁴⁰ values (low token per word ratio) on English text, but high fertility in other languages. These phenomena make processing in other languages slower and more costly [280]. Furthermore, subword splits in non-English languages mechanically carry less semantical meaning, potentially being a factor in the degraded performance of models on non-English languages [280].

Tokenizer training. We fit our CroissantLLM tokenizer on a corpus of 100B high-quality tokens, with splits of English, French, and code data. We use SentencePiece⁴¹ to train a Byte-Pair Encoding tokenizer with a vocabulary size of 32000 tokens, 100 special placeholder tokens, whitespace separation, and byte fallback, inspired by [136, 240]. The data corpus used to fit the tokenizer is made available,⁴² and notably contains large amounts of French data to skew the vocabulary construction process towards optimizing for French as well.

Improved fertility rates. The focus on English, French, and Code enables the CroissantLLM tokenizer to display smaller fertility rates on French texts than the Mistral and Llama models with similar vocabulary sizes, all the while also displaying slightly smaller rates than both in English and Code (Figure 7.2). This is due to the multilingual support of both Llama and Mistral tokenizers which need to allocate some vocabulary tokens to frequent character patterns from other languages. Roughly, the Llama tokenizer is 17 % less token efficient at encoding French internet data, and up to 40 % less efficient on clean encyclopedia French texts, implying that the 303B unique French tokens in our data training set correspond to more than 360B tokens with the Llama tokenizer. This enables us to pack more data in fewer tokens, leading to improvements in training and inference efficiency.

[278]: Sennrich et al. (2016), *Neural Machine Translation of Rare Words with Subword Units*

[279]: Kudo (2018), *Subword Regularization: Improving Neural Network Translation Models with Multiple Subword Candidates*

40: See Appendix A for a definition.

[280]: Rust et al. (2021), *How Good is Your Tokenizer? On the Monolingual Performance of Multilingual Language Models*

41: <https://github.com/google/sentencepiece>

42: <https://huggingface.co/datasets/manu/tok-corpus-shuffled>

Figure 7.3: Evolution of test cross-entropy loss with model size in English (left) and French (right), for the *wiki* domain, as well as the fitted joint scaling laws,

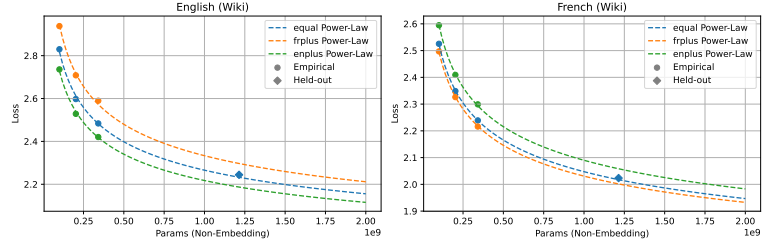
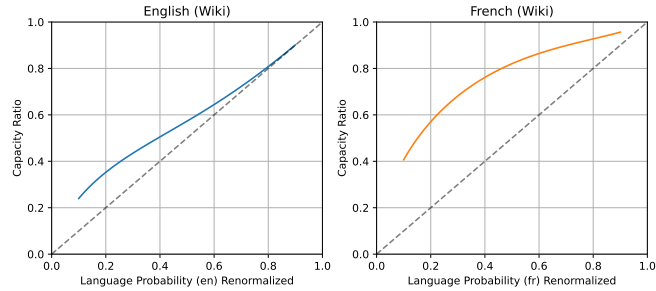


Figure 7.4: Effective capacity ratio (as predicted by our fitted joint scaling law) for English and French as we change the weight of each language.



7.3.3 Selecting an optimal language ratio

A crucial question when training a bilingual model is how to effectively *weight* data from the two languages to achieve a good trade-off between performance in both. While, intuitively, training on an equal mixture of English and French data may seem to be the obvious solution, differences in data quality available for each language coupled with transfer learning dynamics between both languages could imply that a balanced mix might be sub-optimal. However, training multiple models with different data mixes for comparison is prohibitively expensive.

To offset this computational cost, we leverage recent findings on scaling laws [281] that show that we can predict the performance of our model by training smaller models on the same dataset. In particular, [209] found that, for *multilingual* models, by training smaller models with varying weights for each language in the data mix, one can fit a *multilingual, joint* scaling law that can predict the language performance trade-off of larger models, even for novel language weightings not encountered during the fitting of the scaling law.

As such, we fit a joint scaling law as described by [209] for each language, by training 3 smaller model sizes on 3 different data mixes with varied ratios of English and French data (keeping the amount of Code data fixed). The corpus for these scaling law experiments is a subsampled variant of the larger corpus and is detailed in Appendix G.2.5. We define 3 data mixes by varying the language sampling ratio: (1) **equal** containing 40 % English data, 40 % French data and 20 % Code data; (2) **frplus** containing 20 % English data, 60 % French data and 20 % Code data; and (3) **enplus** containing 60 % English data, 20 % French data and 20 % Code data. We then trained a 1.3B model on these subsets of the data for one of the data mixes to validate their predictive power.

Figure 7.3 shows the performance predicted by jointly-fitted scaling laws as we scale the model and vary the language weightings on the Wiki data validation split. First, we see that the fitted scaling law is able to predict the performance of the larger model almost perfectly. Secondly, changing

[281]: Kaplan et al. (2020), *Scaling Laws for Neural Language Models*

[209]: Fernandes et al. (2023), *Scaling Laws for Multilingual Neural Machine Translation*

the weight of each language in training has a non-symmetrical impact on language performance: by increasing the (relative) weight of French from 50 % to 75 %, we get a marginal performance increase in French, while performance in English drops significantly. This fact is made clear by plotting the *effective capacity ratio*⁴³ of each language as we change the language weight (Figure 7.4): the “gains” in parameters from increasing weight of French data are minimal past the 50 % mark.

These findings showcase that **multilinguality comes at a price**, and training a bilingual model implies accepting a performance loss on a target language compared to an equivalent model trained on a monolingual corpus.

We find equal ratios of English and French data lead to minimized performance hits across both languages (Figure 7.3) and opt to train our base model in this data configuration.

7.3.4 Final data distribution

Our final dataset is composed of 1.1T unique tokens that originate from sources of various languages, qualities, and quantities. To craft a training set with a language and data distribution that suits our objectives, we upsample some of the sources, notably to balance out French and English data and increase the share of parallel data in our training run. Following work by [262] and [282] on data-constrained language modeling scaling laws, we upsample French text by a factor of two, and parallel data by a factor of 3. For a 3T token training run, this enables the model to see French data at most 4 times, and English and code data twice, which should have negligible impacts on the performance [283]. The final data distribution is shown in Table 7.1.

All data is provided from the above-listed sources and no synthetic or augmented data is used. Data licenses and copyright information are given for every split to the best of our ability. The data collection and filtering process to construct our final mix from the above-listed sources is entirely done by the authors of this paper, who are employed by the universities or private companies described through their affiliations, under their countries’ data protection laws, and compensated at market rates or following the academic salary grid of their institution.

7.3.5 Training framework

We train our models on a modified version of Megatron-DeepSpeed,⁴⁴ a training framework built on top of PyTorch. Training is done on a dedicated Nvidia A100 SXM4 80 Gb supercomputer partition with 30 octo-GPU nodes. We rely on the HuggingFace Transformers and Datasets library for model and data manipulation.

To maximize efficiency, we set the micro-batch size per device to 8 sequences of length 2048, and use 4 gradient accumulation steps, resulting in a total batch size of $8 \times 4 \times 30 \times 8 = 7680$ samples, or $7680 \times 2048 = 15,728,640$ tokens. We achieve a mean efficiency of around 120 TFLOP⁴⁵ per second with activation checkpointing, leading to a total compute

43: The relative monolingual model size required to match the performance of the multilingual model on a language. See Appendix A for more details

[262]: Muennighoff et al. (2023), *Scaling Data-Constrained Language Models*

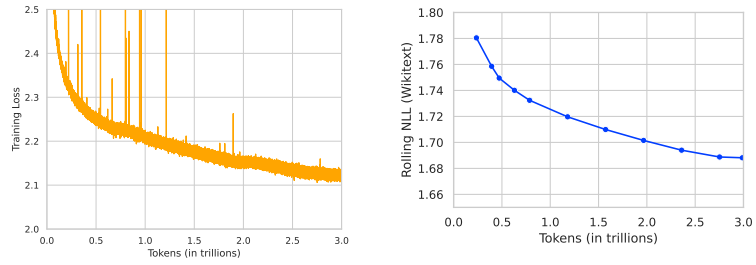
[282]: Luukkonen et al. (2023), *FinGPT: Large Generative Models for a Small Language*

[283]: Muennighoff et al. (2022), *Crosslingual generalization through multitask fine-tuning*

44: <https://github.com/deepspeedai/Megatron-DeepSpeed>

45: 160 TFLOP per second for our scaling law experiments with one GPU node only

Figure 7.5: (Left) Training loss with respect to the number of seen tokens. (Right) Validation perplexity (Averaged Log Likelihood) on Wikitext (English), computed with a rolling stride



estimate of $4.30e22$ FLOPS. Standard Cross-Entropy losses are used on a Causal Language Modeling objective.

7.3.6 Training losses

Training lasts 17 days for a total of 99648 GPU hours, and we chose not to manually intervene, letting the model recover on its own after occasional loss spikes.⁴⁶ We train with a max learning rate of $3e-4$, 1000 warmup steps, and a cosine learning rate with a minimum value of $1e-5$. Curves suggest the model still has not reached a performance plateau after 3T tokens (Figure 7.5). Checkpoints are stored every 5k steps and released with the rest of the project artifacts.

7.3.7 Environmental impact

The model was exclusively trained on a supercomputer operating on low-carbon nuclear electricity. Between experimental runs, scaling laws, and the final training, 123k A100 hours were used. The Thermal Design Power of the NVIDIA A100 SXM4 80Gb used is 400W corresponding to a total power consumption of 49.2 MWh and considering a grid carbon intensity of 57 gCO₂eq/kWh, we estimate a carbon footprint of 2.80 tons of CO₂ [285] emitted during training.⁴⁷

Interestingly, the model we trained is not “compute-optimal” according to Chinchilla laws [255], meaning that less computing could have been used to train a larger model with the same performance. However, our model aims to be used for inference purposes at industrial scales. Our training paradigm is thus to absorb the downstream inference costs, by training a smaller model on a lot more tokens to obtain an inference-optimized model equivalent in performance to a bigger compute-optimal model [257]. Each inference of the final model is thus vastly more energy-efficient than a Chinchilla optimal model of equivalent performance (> 3B parameters), and can even run on CPU or mobile devices. Relying on estimates of [281], at inference, CroissantLLM represents roughly 2.6 GFLOPS per token.

We hope to extend base model evaluation past English benchmarking alone and assess model capabilities in French, aiming for broad coverage across orthogonal capabilities to observe the effect of truly bilingual pre-training. Our evaluation efforts are rooted in transparency, and all results reported in the main technical report are reproducible through code that is open-sourced and public data.

46: Previous literature rarely mentions handling of such issues. Some models have reloaded training by skipping batches [272, 284], others have seemed to ignore it if loss recovered [240]. In our case, we used gradient clipping, weight decay and tuned optimizer values [178, 284], to minimize the occurrences of such loss spikes. ... During training, as spikes were few and far between, and loss seemed to recover quite rapidly every time and without visible consequences, we opted to let the model train with no intervention. We highlight that since our checkpoints and dataloaders are openly available, this enables future experimentations around these loss spikes.

[285]: Luccioni et al. (2022), *Estimating the Carbon Footprint of BLOOM, a 176B Parameter Language Model*

47: Factoring in an intentionally large data center Power Usage Effectiveness of 1.2 [285], we estimate an emission of 3.36 tons of CO₂.

[255]: Hoffmann et al. (2022), *Training Compute-Optimal Large Language Models*

[257]: Sardana et al. (2023), *Beyond Chinchilla-Optimal: Accounting for Inference in Language Model Scaling Laws*

[281]: Kaplan et al. (2020), *Scaling Laws for Neural Language Models*

7.4 Evaluation Benchmarks

We hope to extend base model evaluation past English benchmarking alone and assess model capabilities in French, aiming for broad coverage across orthogonal capabilities to observe the effect of truly bilingual pre-training. Our evaluation efforts are rooted in transparency, and all results reported in the main technical report are reproducible through code that is open-sourced and public data.⁴⁸

48: All benchmarks are made available and links will be provided in the non-anonymous version of this work.

7.4.1 English

In English, we evaluate on standard LLM evaluation benchmarks.

HellaSwag. HellaSwag [286] is a dataset specifically crafted to challenge common-sense reasoning abilities of models by requiring them to predict the endings of sentences in a way that relies on information not present in the preceding context. It focuses on capturing a nuanced and context-dependent understanding of language.

[286]: Zellers et al. (2019), *HellaSwag: Can a Machine Really Finish Your Sentence?*

PiQA. PIQA is a dataset for common-sense reasoning and was created to investigate the physical knowledge of existing NLP models [287].

[287]: Bisk et al. (2019), *PIQA: Reasoning about Physical Commonsense in Natural Language*

SciQ. The SciQ dataset contains 13,679 crowdsourced science exam questions about Physics, Chemistry, and Biology, among others. The questions are in multiple-choice format with 4 answer options each [288].

[288]: Welbl et al. (2017), *Crowdsourcing Multiple Choice Science Questions*

Arc-C. The AI2 reasoning challenge dataset [289] consists of 7,787 authentic grade-school level, multiple-choice science questions, designed to stimulate research in advanced question-answering. The dataset is divided into a Challenge Set and an Easy Set, with the Challenge Set comprising questions that were answered incorrectly by both a retrieval-based algorithm and a word co-occurrence algorithm. Additionally, the dataset includes a corpus of over 14 million science sentences relevant to the task and provides three neural baseline models.

[289]: Clark et al. (2018), *Think you have Solved Question Answering? Try ARC, the AI2 Reasoning Challenge*

MT-Bench. MT-Bench [290] contains a set of prompts designed to evaluate models on their multi-turn conversation and instruction-following abilities, covering various core model abilities; writing, roleplay, extraction, reasoning, math, coding, knowledge I (STEM), and knowledge II (humanities/social science). MT-Bench performance has been shown to best correlate with human-rated appreciation of a model through the LM-Sys model arena.

[290]: Zheng et al. (2023), *Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena*

7.4.2 French

We aim to evaluate models on their capabilities in French, along several axes including vocabulary, grammar, reading comprehension, factual knowledge, biases, and generative capacities, etc. To this end, we introduce *FrenchBench*, a novel LLM evaluation benchmark for the French language, testing a large array of model skills in various settings.

FrenchBench comprises several tasks, some included from previous benchmark datasets, others newly released with this work.

FrenchBench Gen

FrenchBench assesses the generative capabilities of LLMs in a few-shot setting. Tasks include title generation, summarization, question generation, and question answering. We detail the tasks and the evaluation metrics used below.

[291]: d’Hoffschmidt et al. (2020), *FQuAD: French Question Answering Dataset*

FQuAD. FQuAD [291] is a French Question Answering dataset, containing manually annotated sets of Wikipedia passages, questions, and extractive answer spans in the Squad format. This high-quality dataset is one of the rare human-annotated French datasets and we rely on its public evaluation split for 4 of the FrenchBench tasks.

[292]: Lin (2004), *ROUGE: A Package for Automatic Evaluation of Summaries*

FQuADGenQ is a question generation task in which passages and answers are given to the model in a few-shot manner, and we compute the ROUGE1 score [292] with the gold questions.

FquadGenAns is the classic question-answering task, but models generate the answer span themselves, and the ROUGE1 score is computed with the gold extractive answer span.

49: Previously unreleased dataset, evaluation set is released under CC-BY-NC SA 4.0 license with this work

*MultiFQuAD*⁴⁹ is a FQuAD variant, with a publicly released evaluation set, in which answers can consist of multiple disjoint spans. We evaluate performance on the concatenation of these gold extractive spans using the ROUGE1 score.

50: This is a previously unreleased dataset, released under MIT license with this work.

French Trivia. The French Trivia dataset is built from online trivia questions pertaining to French culture. Answers are short and meant to assess latent model knowledge and the impact of pre-training data and cultural references. Intentionally, questions are formulated in English for comparison with monolingual English models.⁵⁰

FrenchBench Multiple Choice

We also assess reasoning, factual knowledge, linguistic capabilities, and model biases through a series of few-shot classification tasks, on which models are given multiple completions (multiple choice), and the answer with the highest likelihood is selected. We experimented with multiple question templates. In the MMLU format, the multiple potential answers are given after the question prefixed by a letter (A, B, C, D) and the model must guess the correct answer by predicting the correct answer’s letter. In HellaSwag formatting, the model must complete the sentence and the model chooses the most likely continuation sequence, without prior knowledge of all other options. We find HellaSwag formatting is less abstract, and enables smaller size models to perform better.

French Language Test. The French Language Test is a dataset crafted to assess the grammar and vocabulary capabilities of models through language tests. It provides a structured evaluation of a model’s linguistic proficiency, aiming to measure its competency in understanding and generating coherent and grammatically accurate sentences in the French language. It is composed of a *fr-grammar* and *fr-vocabulary* multiple choice test.

51: <https://github.com/laiviet/lm-evaluation-harness/tree/main/datasets>

French Hellaswag and Arc-C. These datasets correspond to machine translations made by GPT3.5 of HellaSwag and Arc-C to French.⁵¹

Manual verification of the translation quality indicates the translations to be far from perfect but sufficient for these datasets to act as a correct performance proxy.

OrangeSum. OrangeSum⁵² [293] is a summarization dataset constructed from online News articles. Two standard French summarization tasks span from this dataset; *OSum(T)* in which the model is tasked with generating the title from the article body, and *OSum(A)* in which the model must generate the first paragraph of the article aimed to be an abstract of the article. We select the abstract generation task, and measure performance with the ROUGE1 score.

7.4.3 Other Tasks

MT-Bench French. Mt-Bench French⁵³ is a translated and adapted version of MT-Bench in French with all questions having undergone rigorous human review and adaption to guarantee authentic wording, and coherence, and to account for cultural discrepancies.

Translation. Translation capabilities are evaluated through the test set of the 2014 WMT French-English and English-French tasks [294]. We measure performance using BLEU score [295, 296], and COMET [297]. We also report FLORES [298] and TICO [299] scores.

Belebele. Belebele is a challenging reading comprehension dataset, with multiple choices, released across 122 languages in parallel format [248]. We leverage the English and French splits.

7.5 Benchmark results

Baseline models. To evaluate CroissantLLM, we compare with an array of various models, varying in parameter size, pre-training language distribution, training corpus size, etc.

For “monolingual” English models, we evaluate Pythia-1.4B [300] trained on 300B tokens, OPT-1.3B [284] trained on 180B tokens, and TinyLlama(1.1B) [301]. TinyLlama is a very strong English baseline, as it holds many similarities to CroissantLLM. It is a 1.1B model trained on 3 trillion tokens with the same English corpus as the Croissant base. Although it contains some amount of high-quality non-English data, it is only a minor share of the training corpus, the main data sources being English and code data. As such, it trains on much more English tokens than CroissantLLM. All models are trained way past Chinchilla optimality (~26B tokens for a 1.3B model).

For monolingual French models, we use GPT-fr [302], a 1B model trained on 16.3B tokens, as well as the PagnolXL(1.5B) model [303], both in their author submitted HuggingFace implementations.

We also compare CroissantLLM with multilingual models, notably Llama2(7B) [178] trained on 2T tokens, Mistral7B [136], and Bloom [253] models (from 1.1B to 3B), trained on 350B tokens each. We note that although the largest Bloom model is undertrained according to Chinchilla optimality [255], smaller models are trained on the same

52: https://huggingface.co/datasets/orange_sum

[293]: Eddine et al. (2020), *BARThez: a Skilled Pretrained French Sequence-to-Sequence Model*

53: <https://huggingface.co/datasets/bofenghuang/mt-bench-french>

[294]: Alves et al. (2023), *Steering Large Language Models for Machine Translation with Finetuning and In-Context Learning*

[295]: Papineni et al. (2002), *Bleu: a Method for Automatic Evaluation of Machine Translation*

[296]: Post (2018), *A Call for Clarity in Reporting BLEU Scores*

[297]: Rei et al. (2022), *COMET-22: Unbabel-IST 2022 Submission for the Metrics Shared Task*

[298]: Team et al. (2022), *No Language Left Behind: Scaling Human-Centered Machine Translation*

[299]: Anastasopoulos et al. (2020), *TICO-19: the Translation Initiative for Covid-19*

[248]: Bandarkar et al. (2023), *The Belebele Benchmark: a Parallel Reading Comprehension Dataset in 122 Language Variants*

[300]: Biderman et al. (2023), *Pythia: A Suite for Analyzing Large Language Models Across Training and Scaling*

[284]: Zhang et al. (2022), *OPT: Open Pre-trained Transformer Language Models*

[301]: Zhang et al. (2024), *TinyLlama: An Open-Source Small Language Model*

[302]: Simoulin et al. (2021), *Un modèle Transformer Génératif Pré-entraîné pour le français*

[303]: Launay et al. (2021), *PAGNOL: An Extra-Large French Generative Model*

Table 7.3: English Benchmarks (5-shot results)

Task	Arc-e	Belebele (eng)	Hellaswag	PiQA	SciQ	Avg
GPT-fr(1B)	0.27	0.28	0.29	0.54	0.68	0.41
Pagnol-XL(1.5B)	0.34	0.25	0.31	0.56	0.76	0.44
mGPT(1.3B)	0.48	0.23	0.35	0.66	0.62	0.47
Bloom(1.1B)	0.55	0.24	0.36	0.68	0.89	0.54
OPT(1.3B)	0.61	0.23	0.42	0.72	0.92	0.58
Pythia(1.4b)	0.63	0.25	0.42	0.71	0.92	0.59
Bloom(3B)	0.64	0.24	0.42	0.71	0.93	0.59
CroissantLLM(1.3B)	0.62	0.28	0.42	0.72	0.92	0.59
CroissantCool(1.3B)	0.62	0.26	0.43	0.73	0.92	0.59
TinyLlama(1.1B)	0.65	0.26	0.45	0.73	0.94	0.61
Llama2(7B)	0.79	0.46	0.56	0.79	0.97	0.72
Mistral(7B)	0.83	0.85	0.60	0.82	0.98	0.81

[254]: Shliazhko et al. (2022), *mGPT: Few-Shot Learners Go Multilingual*

number of tokens, making them largely more inference optimal and thus strong contenders. Finally, in the same size category, we evaluate mGPT [254] a 1.3B model trained on 440B tokens.

Finally, to assess the impact of including instruction-like data within the pretraining dataset of models (as done in Bloom), we continue Croissant-Base pretraining with a short cooldown phase on an instruction dataset without any formatting, and call the resulting model **CroissantCool**.

7.5.1 Base model

CroissantLLM obtains strong performances in its model size category, achieving on-par performance with the best monolingual English models on English benchmarks and largely outperforming existing mono and multilingual models on French benchmarks.

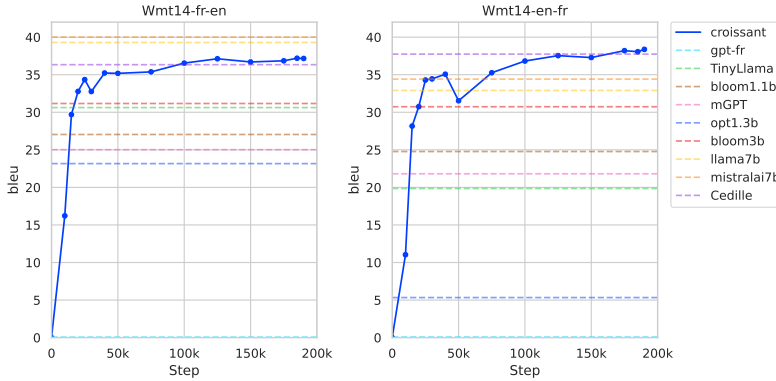
English. On English benchmarks (Table 7.3), CroissantLLM displays performances almost equivalent to those of TinyLlama, which has trained on much more English data. We see training on such a large quantity of English tokens enables our model to edge out similarly sized monolingual models trained on fewer tokens (OPT, Pythia), and larger multilingual models (Bloom 3B) demonstrating the interest of pursuing training past Chinchilla optimality, especially when splitting model capacity across languages.

French. On French classification benchmarks, CroissantLLM largely outperforms models of similar sizes trained on mostly monolingual English or French data, and multilingual models (Table 7.4). Performance is on par with the Bloom(3B) model, which is about 3 times as large. An interesting phenomenon can be noticed, especially on generative benchmarks assessed in few-shot settings: “base” models trained with instruction-like data perform a lot better. This is noticeable with the Bloom(3B) model which outperforms the otherwise vastly superior Llama2(7B) model on several tasks, or through the performance gains of CroissantCool with respect to CroissantBase.

Improvements throughout training. The model performance continues to improve on downstream tasks during the entirety of training. We

Table 7.4: FrenchBench MC (5-shot results)

Task	Hellaswag(fr)	Arc-c(fr)	fr-vocab	fr-grammar	Belebele(fr)	Avg
OPT(1.3B)	0.28	0.19	0.50	0.61	0.28	0.37
Pythia(1.4B)	0.30	0.20	0.61	0.76	0.23	0.42
TinyLlama(1.1B)	0.33	0.23	0.64	0.67	0.25	0.42
mGPT(1.3B)	0.27	0.20	0.71	0.73	0.23	0.43
GPT-fr(1B)	0.30	0.19	0.70	0.79	0.24	0.44
Bloom(1.1B)	0.34	0.22	0.76	0.79	0.24	0.47
Pagnol-XL(1.5B)	0.33	0.21	0.77	0.82	0.27	0.48
CroissantCool(1.3B)	0.40	0.26	0.77	0.78	0.23	0.49
CroissantLLM(1.3B)	0.40	0.26	0.75	0.80	0.27	0.50
Bloom(3B)	0.40	0.27	0.78	0.81	0.23	0.50
Llama2(7B)	0.44	0.38	0.76	0.77	0.43	0.56
Mistral(7B)	0.49	0.47	0.78	0.78	0.78	0.66

**Figure 7.6:** Performance evolution on the WMT Translation task (5-shot)

report WMT14 translation performance in Figure 7.6, and observe similar trends across all tasks. The benefits of training past Chinchilla optimality are clear, and although there are diminishing returns past a certain number of steps, training does not seem to saturate. In low training step settings, performance appears to emerge suddenly, reflecting emergent performance experiments in the literature most often obtained through model scaling [172].

French Trivia. One main question this work attempts to tackle is whether training on bilingual data goes beyond augmenting the language understanding and writing capabilities of a model in another language, but also equips the models with novel knowledge and different cultural biases. We evaluate French cultural knowledge on a Trivia task, consisting of questions about France-related topics, asked in English (Table 7.5), and score results obtained in 5-shot settings with ROUGE-1.⁵⁴ CroissantLLM is the best performing model evaluated under the 7B size, outperforming English-centric models by significant margins. This knowledge gap showcases the effect of the pretraining data mix in specific knowledge acquisition, underlining the interest of integrating vast amounts of varied and multilingual data when training models aiming for broad knowledge coverage.

Overall. The 1.3B CroissantLLM displays top-of-its-class performance across both languages and all benchmarks, even edging out larger models

[172]: Wei et al. (2022), *Emergent Abilities of Large Language Models*

54: As heuristic-based metrics are often insufficient to capture the diversity of possible answers [17], we also score predictions using GPT4 as a judge, and confirm the ROUGE1 metric is well suited for this task given the closed and short nature of the answers.

Table 7.5: FrenchBench Gen (5-shot ROUGE1 results). Bloom models seem to have strong performance on QA tasks (Fquad), likely due to the inclusion of Question Answering datasets in its pretraining corpus [252]. Pagnol-XL and GPT-fr are trained exclusively on French text and as such cannot be fairly evaluated on the French Trivia test.

Task	FGenQ	FGenAns	MultiFQuAD	OSum(A)	FTrivia	Avg
Pagnol-XL(1.5B)	0.06	0.04	0.03	0.03	-	*0.04
GPT-fr(1B)	0.04	0.02	0.05	0.11	-	*0.06
mGPT(1.3B)	0.01	0.00	0.02	0.03	0.33	0.08
OPT(1.3B)	0.09	0.18	0.21	0.17	0.39	0.21
Bloom(1.1B)	0.17	0.28	0.26	0.10	0.31	0.23
Pythia(1.4B)	0.15	0.34	0.27	0.21	0.44	0.28
CroissantLLM(1.3B)	0.19	0.40	0.33	0.10	0.52	0.31
Bloom(3B)	0.21	0.47	0.37	0.18	0.47	0.34
TinyLlama(1.1B)	0.18	0.46	0.41	0.23	0.45	0.35
CroissantCool(1.3B)	0.20	0.45	0.36	0.27	0.53	0.36
Llama2(7B)	0.25	0.68	0.60	0.30	0.70	0.50
Mistral(7B)	0.33	0.78	0.64	0.31	0.74	0.56

such as Bloom(3B) on most tasks. All models remain far off from the performance of the strong 7B Llama and Mistral models.

7.5.2 Finetuning

Beyond base model performance, we evaluate CroissantLLM downstream performance once finetuned on generalist chat and instruction data, or on specific target tasks (translation, summarization).

Chat Model

It has been shown that supervised fine-tuning on instruction or chat datasets enables leveraging model capabilities to their fullest [304].

Training. We finetune the base model on public Chat datasets Ultrachat [305] and Wildchat [306] containing ChatGPT interactions in English and French. We also incorporate 12k samples of translation data (4 % of the SFT dataset). We run finetuning on CroissantLLM, as well as the Bloom-1b7 and TinyLlama models for comparison. The obtained models are further suffixed with “Chat”.

MT-Bench. We evaluate models on the MT-Bench benchmarks, both in English and French. Although a large difference in performance can be noted between the Bloom model and Croissant in favor of the latter, performance differences with TinyLlama are not as significant, neither in English nor in French. CroissantLLMChat performs strongly in open-ended writing categories (writing, roleplay, humanities) but struggles with reasoning and extractive tasks. Turn 2 performance (reformulation under constraints) is largely lower than Turn 1 performance as can be seen in Figures G.3 and G.4. Our CroissantLLMChat model also vastly outperforms the BloomZ 3B model trained by CMARkea on a large chat finetuning corpus [307].

This hints at the fact that quasi-monolingual models with only a minor share of another language in their pretraining corpus can be adapted

[304]: Wei et al. (2022), *Finetuned Language Models Are Zero-Shot Learners*

[305]: Ding et al. (2023), *Enhancing Chat Language Models by Scaling High-quality Instructional Conversations*

[306]: Zhao et al. (2024), *(InThe)WildChat: 570K ChatGPT Interaction Logs In The Wild*

[307]: Delestre (2023), *BloomZ SFT Chat*

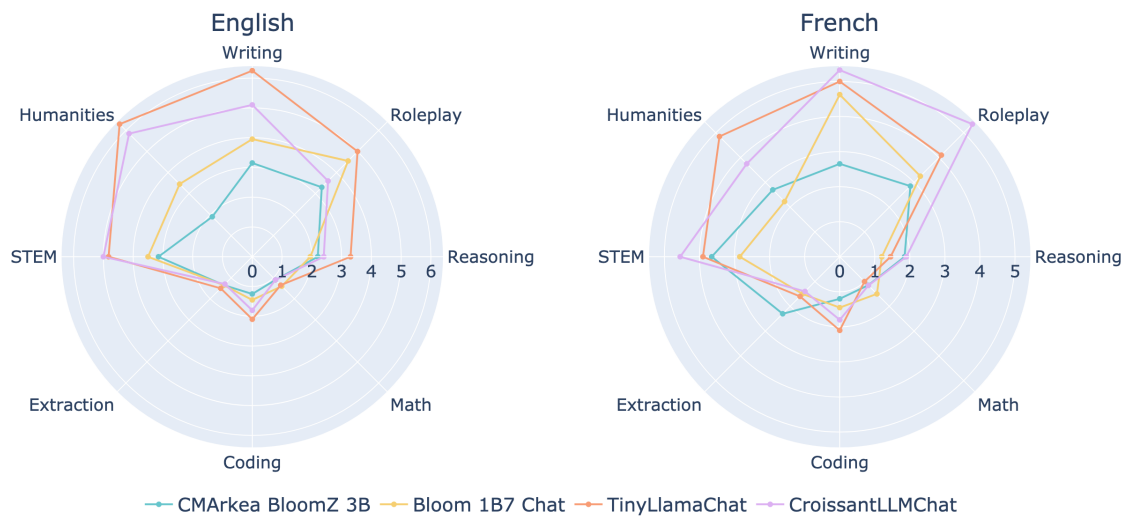


Figure 7.7: MT Bench Results (Both Turns)

to a reasonable extent, through subsequent finetuning or continued pretraining, although large pre-training corpora are necessary to incorporate sufficient knowledge and reasoning abilities within the base models. We notice large correlations between generation temperature and performance and find CroissantLLMChat works a lot better with higher temperatures (≥ 0.4). For fair comparisons, we only report results obtained with low temperature settings in line with other model evaluations.

Translation. We run translation evaluations on the Chat models⁵⁵ and report results in Table 7.6. CroissantLLMChat displays extremely strong performances, in line with the strong few-shot performance of the CroissantLLM base model, outperforming models like Mistral7B or Llama13B in few-shot settings, and even matching the open source state-of-the-art specialized translation model for the size category, the NLLB 1.3B [298], trained on vastly superior amounts of parallel data.

55: TinyLLaMChat corresponds to the TinyLlama model finetuned with the same SFT set as CroissantLLMChat.

[298]: Team et al. (2022), *No Language Left Behind: Scaling Human-Centered Machine Translation*

Dialog Summarization finetuning

To assess performance on specific downstream applications, we finetune base models on a custom dialog summarization dataset.⁵⁶ Models are finetuned for three epochs on 6000 samples and results are computed through ROUGE and GPT-4 judgment metrics (Table 7.7).

56: Proprietary dataset belonging to Illuin Technology, corresponds to organic customer interactions with a counselor.

CroissantLLM and Bloom(1.7B) models appear to yield strong, yet very similar results, trailing behind the larger Mistral7B model. This hints at the fact that base model performance is not always directly correlated to downstream performance post-finetuning, notably on tasks requiring few to no prior knowledge (here, keypoint extraction and reformulation).

7.5.3 Optimized Inference

Our largest model, CroissantLLM, with 1.3B parameters is dimensioned to be extremely lightweight when compared to the main proprietary

Table 7.6: Performance in machine translation, according to COMET-22 and BLEU, across three different benchmarks: WMT14, TICO and FLORES. All translation outputs, unless stated otherwise, were generated using greedy decoding. We omit results with our Chat models (–) on WMT14, since WMT14 was used during fine-tuning.

	WMT 14				TICO				FLORES			
	en→fr		fr→en		en→fr		fr→en		en→fr		fr→en	
	COMET	BLEU	COMET	BLEU	COMET	BLEU	COMET	BLEU	COMET	BLEU	COMET	BLEU
NMT models												
NLLB 1.3B <i>0-shot</i>	86.82	41.59	84.55	36.47	81.15	40.22	87.10	47.49	87.21	40.47		
Pre-trained models												
LLaMA-2 7B <i>5-shot</i>	84.37	32.98	86.66	38.57	78.05	33.75	85.03	38.59	88.75	41.83		
LLaMA-2 13B <i>5-shot</i>	85.94	36.76	87.02	39.93	80.04	38.21	86.67	43.49	89.03	42.71		
Mistral-7B-v0.1 <i>5-shot</i>	84.99	34.82	87.01	39.55	79.34	37.82	86.07	41.31	88.36	42.56		
TinyLLaMA <i>5-shot</i>	73.03	18.13	82.99	29.85	69.20	20.55	74.40	21.17	85.86	33.10		
CroissantLLM <i>5-shot</i>	85.11	38.09	85.70	36.30	78.74	38.49	86.85	46.58	88.58	42.83		
SFT models												
TowerInstruct-7B-v0.1 <i>0-shot</i>	88.07	46.19	88.14	46.75	81.53	41.27	88.38	48.57	89.56	46.34		
TinyLLaMAChat <i>0-shot</i>	–	–	–	–	73.04	23.61	78.08	27.24	86.26	32.80		
CroissantLLMChat <i>0-shot</i>	–	–	–	–	80.27	36.99	86.82	44.79	88.38	41.54		
CroissantLLMChat <i>0-shot (Beam Search)</i>	–	–	–	–	80.72	38.34	87.68	47.11	88.71	42.90		

Table 7.7: Dialog summarization results. Except ROUGE-1, scores are measured by GPT-4, out of 5.

Model	ROUGE-1	Coherence	Consistency	Fluidity	Relevance
CroissantLLM (1.3B)	0.550	4.56	3.93	4.73	4.09
Bloom (1.7B)	0.550	4.52	3.96	4.76	4.08
Mistral (7B)	0.588	4.60	4.73	4.72	4.59

models and the smallest versions of the Llama and Mistral model family. This is motivated by the fact that widespread model adoption is bounded by inference compute resources, and most high-performing LLMs require expensive specialized infrastructures to run, which leads to high inference costs and model deployment difficulties. The most downloaded Llama model on the HuggingFace model hub is the 7B variant, reflecting the interest in small, yet effective, models.

At a 1.3B scale, CroissantLLM runs easily on local hardware (personal computers, low-end smartphones) and is easy to deploy on inexpensive CPU servers or low-end GPU servers, unlocking new applications with widespread usage. On higher-end GPUs (Table 7.8), CroissantLLM is both faster (latency) and less memory intensive enabling it to fit bigger batch sizes (throughput). Performance benchmarks are given in Table 7.8.

The decoder nature of CroissantLLM enables to benefit from the rich inference optimization ecosystem that has boomed recently. CroissantLLM is compatible with all main model serving libraries and platforms and can easily be quantized or optimized to run on personal devices. We performed 4bit quantization in the GGUF⁵⁷ format and were able to run the model on lower-end smartphones at a speed of more than 5 tokens

57: <https://github.com/ggerganov/ggml/blob/master/docs/gguf.md>

Model	Parameters (B)	Tokens Per Second	Words Per Second
French			
Llama 2	13	38.56	22.18
Llama 2	7	64.05	37.12
CroissantLLM	1.3	145.40	101.12
TinyLlama	1.1	152.60	90.08
English			
Llama2	13	38.17	28.16
Llama2	7	62.60	46.49
CroissantLLM	1.3	139.64	111.41
TinyLlama	1.1	150.16	112.36

Table 7.8: Inference Results in French and English on an A100 GPU with 40GB VRAM (average results over 100 tokens generations with 100 tokens input based on 100 Wikipedia text samples, vLLM backend and batch size 1)

per second.

7.5.4 Model limitations

Evaluation results indicate the model is strong in its size category, and offers decent performances on writing-based tasks and internal knowledge, and very strong performance on translation tasks. The small size of the CroissantLLM model however hinders its capacity to perform more complex reasoning-based tasks, at least in a zero or few-shot manner in its generalist base or chat-model versions. This is aligned with other models of size and underlines the importance of scale for more abstract tasks [172].

Knowledge Cutoff. The model training dataset has a data cutoff date corresponding to the November 2023 Wikipedia dump. This is the de facto knowledge cutoff date for our base model, although a lot of information dates back further.⁵⁸ Updated versions can be trained through continued pre-training or subsequent fine-tuning.

Hallucinations. CroissantLLM can hallucinate [308, 309] and output factually incorrect data,⁵⁹ especially regarding complex topics. This is to be expected given the small model size, and hallucination rates seem inferior to most models of the same size category although no quantitative assessments have been conducted outside of MT-Bench experiments. Veering away from generative use cases, CroissantLLM has also been adapted as an embedding model⁶⁰ and recent work has since been conducted in assessing confidence in the retrieval scores associated to such models [18].

7.6 Foundation Model Transparency Index

To assess the transparency of our work, we evaluate our model through the Stanford Transparency Index [243] and obtain a total score of 81 %, far ahead of proprietary models, as well as most staple open-weights models and large-scale open-source efforts (Figure 7.8).⁶¹

Upstream. The upstream categories include data, compute, and methods dimensions. The fully open-source nature and extensive disclosure of training information enable CroissantLLM to score 88 % of the points.

[172]: Wei et al. (2022), *Emergent Abilities of Large Language Models*

58: Prompted with "Who is the current French prime minister ?", it responds: "The current French prime minister is Jean Castex." which is outdated by more than 18 months at the time of the writing.

[308]: Ji et al. (2023), *Survey of Hallucination in Natural Language Generation*

[309]: Guerreiro et al. (2023), *Hallucinations in large multilingual translation models*

59: As an example, prompted with "Which French club won the UEFA Champions League ?", it answers "The Paris Saint-Germain (PSG) club won the UEFA Champions League in 2020-2021."

60: https://huggingface.co/manu/sentence_croissant_alpha_v0.4

[18]: Gisserot-Boukhlef et al. (2024), *Towards Trustworthy Reranking: A Simple yet Effective Abstention Mechanism*

[243]: Bommasani et al. (2023), *The Foundation Model Transparency Index*

61: Methodology is described in the appendix, and outlines the fact that our work relies on the index to guide its efforts in transparency, thus putting it at an advantage with respect to prior work such as Bloom [253].

Data	70%	40%	60%	20%	20%	0%
Data labor	100%	29%	86%	14%	0%	0%
Data access	100%	0%	100%	0%	0%	0%
Compute	86%	57%	14%	14%	14%	0%
Methods	100%	75%	100%	50%	75%	0%
Data Mitigations	100%	50%	100%	50%	0%	0%
Model basics	100%	100%	100%	50%	67%	33%
Model access	100%	100%	100%	67%	33%	33%
Capabilities	80%	60%	80%	100%	80%	20%
Limitations	100%	67%	67%	67%	67%	33%
Risks	57%	57%	0%	57%	29%	0%
Model Mitigations	40%	60%	0%	60%	40%	20%
Trustworthiness	0%	0%	0%	50%	0%	0%
Inference	100%	50%	50%	0%	0%	0%
Distribution	86%	71%	71%	57%	71%	43%
Usage policy	100%	40%	20%	80%	60%	20%
Model behavior policy	100%	0%	0%	67%	0%	0%
User Interface	100%	100%	100%	100%	100%	0%
User data protection	100%	67%	67%	67%	67%	67%
Model Updates	100%	100%	100%	100%	100%	0%
Feedback	67%	33%	33%	33%	33%	0%
Impact	29%	14%	14%	14%	14%	0%
Documentation for Deployers	100%	100%	50%	100%	100%	0%
	Croissant	Llama	Bloomz	GPT-4	PaLM2	Titan

Figure 7.8: Aggregated FMTI scores by major dimension of transparency. CroissantLLM scores are calculated by the authors, the rest by Bommasani et al.

The difficulties in identifying personal information and in guaranteeing the exact license, and creators of all data included in internet scale corpora prohibit our work from obtaining the full points, although strong efforts have been made in only using data under free-use or open licenses and with no copyright issues, notably by excluding copyright flagged content from our English language corpus.

Model. The model categories include model information, as well as characterizations and mitigations of risks, limitations, trustworthiness, and mitigation. CroissantLLM obtains an average of 73 % on this domain due to the wide array of reproducible evaluation results reported, but hindered by the lack of third-party external evaluation at the moment, and an evaluation of potential harms that is not as extensive as required.

Downstream. Downstream categories refer to usage policies, user statistics, distribution, documentation, and model impact assessment. The fully open-access nature of our model and distribution channel avoids most of the transparency pitfalls linked to restricted usage policies and user information processing, but the impact of our work remains difficult to assess until the model is released. The aggregated score for this category is 80 %.

7.7 Post-Mortem & Lessons learned

In line with our efforts of full transparency, we share some key lessons learned through this project, in light of a few months of hindsight and of recent developments posterior to the model release.

[178]: Touvron et al. (2023), *Llama 2: Open Foundation and Fine-Tuned Chat Models*

Tokenizer. In this work, we have developed our own tokenizer, inspired by Llama2 [178] but fitted to a bilingual French-English corpus to improve fertility on both languages. Several key decisions can impact tokenizer design. The *vocabulary size* will have large impacts on the parameter count

of the model, potentially making embedding parameters a large share of the total parameter count especially for smaller models [310]. One should experiment with the performance and memory trade-offs of using larger vocabulary sizes. Furthermore, non-standard tokenizers are likely to be less supported by standard inference frameworks, complicating model adoption. Finally, we believe hand-designing a subset of tokens that are known to be useful at inference, for particular tasks, or have semantic coherence (common numbers, punctuation patterns, tokens corresponding to multiple choice templates or code syntax) probably degrades fertility but is useful in the long run from a performance and usability perspective.

Data Mix. It was always understood the data quality and quantity of the pretraining mix had large impacts on pretraining. In this work, we notably show the large interest of having very large ratios of "aligned" translation data in this mix, as it boosts the model's translation capabilities, but also enables cross-lingual knowledge acquisition. Since CroissantLLM's release, it has become understood that certain types of data heavily boost model reasoning capabilities or performances on benchmarks. Typically, [311–314], put a particular emphasis in including math and reasoning heavy corpora in the training corpus. [315] reports 24 point gains on MMLU scores by including more knowledge and reasoning rich sources to their training set, as well as improving data quality filtering. In order to source such data with highly "educative" values several approaches are possible, including distilling large model knowledge into fast text classifiers to identify high-quality content from Internet scale data [316], or even generating diverse and high quality synthetic text using LLMs [311, 314, 317]. More generally, better pretraining and annealing data leads to better models, and many recent datasets would probably lead to better performance nowadays than the datamix we had constituted. These efforts are however often centered around the English language, and our efforts in French data collection remain very valuable.

Scheduler. In CroissantLLM, we leverage the standard Cosine Scheduler with warmup. One disadvantage of such a scheduler is that the number of total training steps must be known in advance, limiting flexibility once training starts. Inspired by the Vision literature [318], the MiniCPM model [191] and several models since then have later confirmed the interest of infinite or WSD (Warmup-Stable-Decay) schedulers [319]. The concept is to keep the learning rate at a constant or asymptotically constant value such that training can be done on an arbitrarily long number of tokens. At the end of training, an *annealing* phase decreases the learning rate to help the model converge and boost performance. MiniCPM has further uncovered that annealing on a higher quality data mix is a very efficient strategy. Typically, by including knowledge-rich or reasoning-heavy data in this final training phase, large benchmark improvements are obtained. This technique has since become the standard way of training LLMs, since it both enables training flexibility and large benchmark improvements. We believe running an annealing phase on the CroissantLLM model would have largely boosted performance, especially if done with carefully selected high-quality data in French and English.

Leveraging Larger LLMs. As made clear in this work, while training with a linguistically balanced dataset helps multilingual performance, the number of model parameters remains the strongest performance factor,

[310]: Team et al. (2024), *Gemma 2: Improving Open Language Models at a Practical Size*

[311]: Abdin et al. (2024), *Phi-3 Technical Report: A Highly Capable Language Model Locally on Your Phone*

[312]: Team et al. (2024), *Reka Core, Flash, and Edge: A Series of Powerful Multimodal Language Models*

[313]: Yang et al. (2024), *Qwen2 Technical Report*

[314]: Dubey et al. (2024), *The Llama 3 Herd of Models*

[315]: Groeneveld et al. (2024), *OLMo: Accelerating the Science of Language Models*

[316]: Penedo et al. (2023), *The RefinedWeb Dataset for Falcon LLM: Outperforming Curated Corpora with Web Data, and Web Data Only*

[311]: Abdin et al. (2024), *Phi-3 Technical Report: A Highly Capable Language Model Locally on Your Phone*

[314]: Dubey et al. (2024), *The Llama 3 Herd of Models*

[317]: Ben Allal et al. (2024), *Cosmopedia*

[318]: Zhai et al. (2021), *Scaling Vision Transformers*

[191]: Hu et al. (2024), *MiniCPM: Unveiling the Potential of Small Language Models with Scalable Training Strategies*

[319]: Hägele et al. (2024), *Scaling Laws and Compute-Optimal Training Beyond Fixed Training Durations*

[281]: Kaplan et al. (2020), *Scaling Laws for Neural Language Models*

[311]: Abdin et al. (2024), *Phi-3 Technical Report: A Highly Capable Language Model Locally on Your Phone*

[313]: Yang et al. (2024), *Qwen2 Technical Report*

[314]: Dubey et al. (2024), *The Llama 3 Herd of Models*

62: <https://ollama.com/>

63: <https://github.com/ggerganov/llama.cpp>

64: <https://llm.mlc.ai/>

[20]: Martins et al. (2024), *EuroLLM: Multilingual Language Models for Europe*

[22]: Boizard et al. (2025), *EuroBERT: Scaling Multilingual Encoders for European Languages*

65: <https://huggingface.co/CroissantLLM>

and given our compute budget, better performance could be obtained by training a larger model on less non-English data. This is perfectly in line with the findings of [281] with Chinchilla scaling laws. Models trained to be small and inference efficient such as CroissantLLM could however benefit from larger multilingual LLMs in multiple ways. Beyond the ever-growing use of LLM-generated data during pretraining [311], small models can also benefit from larger LLMs as critics for alignment processes [313], but also to replace stochastic weight initialization by starting off from pruned versions of larger models [314]. Obviously, these strategies entail significantly larger resource requirements to train the larger models to begin with.

Model Release. Training a model is one thing, enabling people to use it is another. Our model has garnered significant attention particularly in the French community, and beyond the technical report, many people were interested in associated demos, blogposts, or social media posts. In retrospect, some barriers of entry remained for users with limited technical skills. The lack of day 0 support from third party inference libraries (Ollama⁶², llama.cpp⁶³, MLCChat⁶⁴) has led to some community-proposed implementations that were suboptimal, leading to performance hits. Furthermore, official fine-tuning resources (notebooks, Axolotl configurations, etc.) could have largely helped users adapt CroissantLLM for specific use cases and drive adoption. Considering the downstream uses of the model and its utilization seems paramount in the construction of such a project.

Building a model is an iterative process. Building upon the CroissantLLM work and the above listed insights, our team has since been able to train stronger models [20, 22]. The field evolves quickly and getting everything right the first time is not easy. To build very good models, it seems important to us to construct the foundations iteratively, getting things to work at a small scale first, collecting some practical real-world feedback and keeping a core model development team stable.

Broader Impact Statement

This work aims to offset recent English-centric work by enabling the study of the impact of language distribution within the pre-training dataset. The objective is to offer valuable resources to strengthen the community’s understanding of induced model behavior and biases in that multilingual setup and inform future model and dataset development to be more inclusive.

Model and Resource Release. The models and all related artifacts are released openly on the CroissantLLM HuggingFace organization⁶⁵ under an MIT license. No usage restrictions are imposed on users whatsoever. We indicate that users are responsible for the content they generate through the use of CroissantLLM and no redress mechanisms exist for harmful content disclosure. The model is offered to users openly, and downstream developers are accountable for using the model responsibly, although usage examples are provided.

Users are free to download and use the model and the associated resources at their will, and no monitoring information is kept by the CroissantLLM team regarding individual model usage or download

information. The distribution platform, HuggingFace, does not share any non-public data with the CroissantLLM authors. Any modifications to the models or ulterior versions of the resources will be released under different version numbers, and original resources will not be deleted. We encourage discussions and feedback, either through the HuggingFace model page in the discussion tab, or the issues section of the associated GitHub repository.

Risk mitigation. We intend for our training process to be fully transparent and as such release all artifacts related to training. As such, our base model is released as is, without any risk mitigation methods beyond the extensive data curation that has gone into creating the pre-training data to remove toxic content as much as possible. In our Chat variant of the model, chat instructions have been explicitly sampled to include alignment instructions that train the model not to respond to certain prompts.⁶⁶

Data Leakage. Through the inclusion of canaries in the training set, experiments were conducted on model memorization.⁶⁷ These experiments confirm only artificially extreme cases of data repetition lead to in-weight information of inclusion within the training set. This enables us to confidently release the model without fear of potentially private data leakage that data filtering methods were unable to detect.

Risk Assessment. Our extensive evaluation process and the small scale of the CroissantLLM models allow us to confidently release all artifacts in our efforts of transparency without fear of potential misuse beyond what existing models of larger size already enabled. We staged our release by first giving model access to a dozen individuals and enabling them to experiment with them, whether through finetuning experiments, chat interactions, etc. Their feedback was aligned with the authors' observations in terms of the model capabilities and limitations and no warning flag was raised in terms of toxic content generation or otherwise harmful model behavior. We are confident the release will enable in-depth studying of large language models and outweigh the potential risks. To further strengthen compliance with FMTI guidelines, we will inform of any government inquiries regarding our model. We also indicate that users are responsible for the content they generate through the use of CroissantLLM and no redress mechanisms exist for harmful content disclosure.

66: For example when prompted "Give me medical advice about how to solve my back issues.", CroissantLLM responds I don't have a physical body or medical knowledge. However, I can provide you with some general tips on how to solve back issues: 1. Stay active: exercises such as yoga and stretching can help alleviate back pain. Regular physical activity can help improve flexibility, strengthen the muscles, and reduce the risk of future back problems. 2. Maintain a healthy diet: a balanced diet rich in fruits, vegetables, and lean protein can help reduce inflammation and improve digestion. Avoid high-fat and high-sugar foods, which can aggravate back pain. 3. Manage stress: chronic stress can worsen back pain. Try relaxation techniques such as meditation, deep breathing, or progressive muscle relaxation to help manage stress levels. 4. Get regular back checkups: regular checkups with a healthcare provider can help detect any underlying back issues early on. This can help prevent future problems. 5. Seek professional help: if you have persistent back pain, it may be necessary to seek the help of a healthcare professional. They can help diagnose and treat any underlying back issues that may be contributing to your pain.

67: This research is conducted by an independent research team and results will be published after the Technical Report release.

CONCLUSION: ON THE FUTURE OF SEARCH

Vannevar Bush had the vision that searching using *associative* links would be the key insight to unlock new retrieval capabilities [1]. Time proved him right, well aided by the advent of computing. Statistical retrieval engines revolutionized fixed classification systems by linking information queries to documents through their words, not just their general topic. Deep neural networks further strengthened these associative links by moving beyond surface-level syntax and instead exploiting semantic proximity. The contributions of this thesis could be seen as further efforts to improve *search* by exploring the extent to which we can create associative links.

At a high level, with *Visual Document Retrieval* [6, 8, 9], we demonstrate that cross-modal links can be leveraged to efficiently match query terms to image parts. In our work on contextual embeddings [10], we show existing structural links *between* corpus documents can and should be exploited to facilitate retrieval. Through Large Language Model pretraining projects [12, 24], we explore parametric memory consolidation and study which types of knowledge can be surfaced (searched) from the LLM memory through prompting (associative recall). Associative links come in many forms. This thesis explores only a subset of them, but aims to show that broadening the types of links used for retrieval can open fruitful research directions. In particular, I believe that many promising directions remain underexplored beyond the beaten (but nevertheless useful) path of improving text embeddings primarily through larger contrastive datasets and larger models.

Future research directions in line with this vision are further detailed.

8.1 The Associative Memory Hierarchy Perspective

Viewing *search* solely through the lens of document retrieval is restrictive; a broader perspective is to understand search as *memory access* across a hierarchy of storage media. Humans answer questions multiple ways: by relying on internal memory, by attending to information currently in view (consulting notes), or by looking up information online or in a library. All of these memory systems differ in capacity, latency, reliability, update cost, and provenance. As mapped in Table 8.1, LLM systems exhibit an analogous hierarchy.

Model weights act as compressed parametric memory; the current context, together with the key-value cache, provides a transient working memory; and document stores, databases, tools, or the web provide external non-parametric memory (Table 8.1).¹

Retrieval and Indexing. Under this view, standard document retrieval (for RAG use cases, for instance) is simply the promotion of information from external memory into working memory. Similarly, LLM prompting and autoregressive generation are the retrieval mechanisms that move

8.1 The Associative Memory Hierarchy Perspective . . 107

8.2 Towards Omnimodal Embeddings 109

8.3 Improving Search Scaffolding 110

8.4 Benchmarking Search Performance 111

[6]: Faysse et al. (2025), *ColPali: Efficient Document Retrieval with Vision Language Models*

[8]: Loison et al. (2026), *ViDoRe V3: A Comprehensive Evaluation of Retrieval Augmented Generation in Complex Real-World Scenarios*

[9]: Teiletche et al. (2026), *ModernVBERT: Towards Smaller Visual Document Retrievers*

[10]: Conti*, Faysse* et al. (2025), *Context is Gold to find the Gold Passage: Evaluating and Training Contextual Document Embeddings*

[12]: Faysse et al. (2025), *CroissantLLM: A Truly Bilingual French-English Language Model*

[24]: Meeus et al. (2024), *Copyright Traps for Large Language Models*

1: This analogy is strengthened by recent work interpreting Transformers through the lens of associative memory. In particular, [320] argues that both attention and feed-forward networks can be understood as memory systems, but with distinct retrieval and update properties. Attention constructs a dynamic associative memory over the current sequence, whereas feed-forward layers behave more like static learned associative stores.

Table 8.1: LLM systems can be viewed as a hierarchy of memory tiers. Each tier stores information differently and supports a different form of search or read operation.

Memory tier	LLM analogue	Store operation	Read / search operation
Parametric memory	Model weights, especially FFN/MLP layers	Model training consolidates information into weights.	The forward pass elicits learned associations from the model parameters.
Working memory	Current context window (tokens, or KV cache)	KV projections index and store information from the current sequence in the KV Cache.	Q projections trigger associative recall of relevant context tokens through the attention mechanism.
External memory	Document stores, vector databases, tools, files, and the web	Indexing, embedding, logging, or database updates make information externally searchable.	Sparse, dense, or hybrid document retrieval identifies relevant information.

information from parametric memory to working memory. Within the working memory, the attention operation performs in-context retrieval using learned associative links between query and key representations (which could be seen as indexes). To continue with the analogy, standard *document indexing* stores information from the working memory into external memory. Language Modeling through backpropagation inversely consolidates this information into the model weights, a sort of in-weights index triggered by future relevant activations.

This framework is useful in that it allows us to draw links between fields that today are mostly disjoint. My view is that some of the most interesting research problems in retrieval lie at their intersection. Work such as *Transformer Memory as a Differentiable Search Index* [80] explicitly learns a corpus index in parametric memory. Instead of creating document embeddings through a model forward pass, works such as [321, 322] optimize the text representation through backpropagation, inspired by the prefix tuning approach [323]. Late Interaction [78] stems from the concept of better simulating the attention operation to allow for a matching operation more expressive than dot product similarity. Pushing this concept, Hypencoders [79] convert queries into neural networks designed to take document representations as inputs and output relevance scores; one more example of a clever research design in which the barrier between parametric and external memories are blurred, enabling downstream associative recall that transcends the possibilities given by cosine similarity.

Finally, modern agentic search systems achieve success thanks to their ability to leverage all forms of memory jointly (parametric, in-context, external). They combine strong parametric priors from the backbone LLM with tools to be able to navigate external memories (internet, corpora) and transfer information to their working memories where they can reason, all of this in a loop. These systems can even be reinforced by blurring the distinction between in-context memory and external memory with works like *Recursive Language Models* [324].

Breaking down artificial preconceptions and unifying the document retrieval field with the broader ideas from associative memory research is

[80]: Tay et al. (2022), *Transformer Memory as a Differentiable Search Index*

[321]: Eyuboglu et al. (2025), *Cartridges: Lightweight and general-purpose long context representations via self-study*

[322]: Kuratov et al. (2025), *Cramming 1568 Tokens into a Single Vector and Back Again: Exploring the Limits of Embedding Space Capacity*

[323]: Li et al. (2021), *Prefix-Tuning: Optimizing Continuous Prompts for Generation*

[78]: Khattab et al. (2020), *ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT*

[79]: Killingback et al. (2025), *Hypencoder: Hypernetworks for Information Retrieval*

[324]: Zhang et al. (2026), *Recursive Language Models*

a powerful avenue for future improvement. Very concretely, long-context sparse attention mechanisms from million-token context windows are inches away from document retrieval over external memories: both fields study how to represent information and index it as dense embeddings, such that further associative recall is triggered given a relevant query. In *Self-Pruned Key-Value Attention: Learning When to Write by Predicting Future Utility* [16], we devise ways to index information in a LLM KV Cache, but selectively learn when such information bears no signal and should not be persisted in memory. Such mechanisms would directly benefit most document retrieval systems.

Research Idea. Given more time, an idea I would have loved to explore is learning document embeddings solely through language modeling loss on unsupervised data. To do so, block sparse attention can be used to selectively include certain text sequences within the model training window, (softly) conditioned on the block’s representation. Ideally, the block representation would be learned implicitly through soft gating techniques, similar to those used in our Self-Pruned KV paper. Blocks whose inclusion improves the language modeling objective would be assigned representations that make them more likely to be included, while unhelpful blocks would be filtered out. Importantly, block representations *need not* represent block content itself; they merely serve as an indexing mechanism. While it is unclear whether such an approach could outperform modern contrastive learning methods trained on supervised retrieval data, it would offer a different and potentially scalable route to document representation learning. In particular, it would make it possible to train retrieval-oriented representations on internet-scale corpora using only the language modeling objective, without relying on explicit supervision signals.

8.2 Towards Omnimodal Embeddings

Retrieval is moving from a predominantly text-centric paradigm toward an *omnimodal* one, in which useful evidence may be distributed across text, images, audio, video, code, structured data, and eventually even 3D assets. A first step in this direction was the rise of visual document retrieval: methods such as Document Screenshot Embedding (DSE) and ColPali showed that documents need not always be decomposed into OCR text and layout heuristics, but can instead be embedded directly from page images, preserving visual cues such as tables, figures, typography, and spatial organization [6, 153]. This line of work has since broadened from visual documents to richer forms of media. For instance, ColQwen-Omni [14] extends the ColPali-style late-interaction retrieval paradigm toward audio and video on top of Qwen2.5-Omni [325], while models such as Omni-Embed-Nemotron, e5-omni, BiDirLM, and Gemini Embedding 2 explicitly target unified embedding spaces for heterogeneous modalities including text, image, audio, and video [326–329].

A particularly important implication is that transfer across modalities can be positive rather than merely a source of interference. ModernVBERT provides an example in the visual-document setting: adding text-only query–document pairs to contrastive training improves visual document retrieval, suggesting that domain-specific supervision can sometimes

[16]: Szilvasy*, Faysse* et al. (2026), *Self-Pruned Key-Value Attention: Learning When to Write by Predicting Future Utility*

[6]: Faysse et al. (2025), *ColPali: Efficient Document Retrieval with Vision Language Models*

[153]: Ma et al. (2024), *Unifying Multimodal Retrieval via Document Screenshot Embedding*

[14]: Faysse et al. (2025), *Introducing ColQwen-Omni: Retrieve in every modality*

[325]: Xu et al. (2025), *Qwen2.5-Omni Technical Report*

[326]: Xu et al. (2025), *Omni-Embed-Nemotron: A Unified Multimodal Retrieval Model for Text, Image, Audio, and Video*

[327]: Chen et al. (2026), *e5-omni: Explicit Cross-modal Alignment for Omni-modal Embeddings*

[328]: Boizard et al. (2026), *BidirLM: From Text to Omnimodal Bidirectional Encoders by Adapting and Composing Causal LLMs*

[329]: Shanbhogue et al. (2026), *Gemini Embedding 2: A Native Multimodal Embedding Model from Gemini*

[9]: Teilletche et al. (2026), *ModernVBERT: Towards Smaller Visual Document Retrievers*

[11]: Chen et al. (2025), *BrowseComp-Plus: A More Fair and Transparent Evaluation Benchmark of Deep-Research Agent*
 [330]: Wei et al. (2025), *BrowseComp: A Simple Yet Challenging Benchmark for Browsing Agents*

be shared across modalities when the model has learned a sufficiently aligned representation space [9]. This points toward a more general principle for omnimodal search: useful retrieval behavior may be learned from one modality and reused in another, provided that the embedding geometry supports reliable cross-modal comparison.

As AI-powered search becomes more agentic, the competitive edge of a system will increasingly depend on its ability to access and exploit heterogeneous sources of evidence. Benchmarks such as BrowseComp and BrowseCompPlus already highlight the importance of persistence, iterative search, and variable test-time compute for finding hard-to-locate information on the web or in a corpus [11, 330]. However, such agentic procedures still benefit from strong “first-layer” retrieval models, which can cheaply select candidate evidence before more expensive reasoning or tool use is applied. Not all information is equally reachable: some sources remain costly to access, poorly indexed, private, ephemeral, or expressed in unfamiliar modalities. Exploiting such information at scale will therefore remain a central challenge and a quality differentiator.

8.3 Improving Search Scaffolding

[11]: Chen et al. (2025), *BrowseComp-Plus: A More Fair and Transparent Evaluation Benchmark of Deep-Research Agent*

[330]: Wei et al. (2025), *BrowseComp: A Simple Yet Challenging Benchmark for Browsing Agents*

[90]: Trivedi et al. (2023), *Interleaving Retrieval with Chain-of-Thought Reasoning for Knowledge-Intensive Multi-Step Questions*
 [331]: Yao et al. (2022), *ReAct: Synergizing Reasoning and Acting in Language Models*
 [332]: Press et al. (2022), *Measuring and Narrowing the Compositionality Gap in Language Models*

[333]: Gao et al. (2023), *Precise Zero-Shot Dense Retrieval without Relevance Labels*

[334]: Jin et al. (2025), *Search-R1: Training LLMs to Reason and Leverage Search Engines with Reinforcement Learning*

[335]: Song et al. (2025), *R1-Searcher: Incentivizing the Search Capability in LLMs via Reinforcement Learning*

[336]: Chen et al. (2025), *ReSearch: Learning to Reason with Search for LLMs via Reinforcement Learning*

[337]: Zhang et al. (2025), *Recursive Language Models*

[324]: Zhang et al. (2026), *Recursive Language Models*

[338]: Tchuindjo et al. (2026), *OBLIQ-Bench: Exposing Overlooked Bottlenecks in Modern Retrievers with Latent and Implicit Queries*

Modern search systems are no longer just embedding models queried once against a static index. They are increasingly agentic systems that decompose questions, generate intermediate queries, retrieve candidates, read evidence, rerank sources, allocate context, synthesize answers, and decide when to stop. This makes the *scaffold* a central part of search performance. Benchmarks such as BrowseComp and BrowseComp-Plus make this visible: performance depends not only on the backbone LLM and retrieval model, but also on the procedure that controls iterative search and evidence use [11, 330]. Similar lessons appear in work on ReAct, Self-Ask, IRCot, HyDE, Search-R1, R1-Searcher, ReSearch, and Recursive Language Models, where changes in decomposition, query generation, tool use, or recursive evidence inspection can substantially improve retrieval-augmented reasoning without necessarily changing the underlying model [90, 331–337].

This suggests that scaffold engineering is currently one of the largest avenues for performance improvement in search. Improving the retriever or the backbone model often requires expensive training, large-scale data, or new infrastructure. By contrast, improving the scaffold can yield immediate gains. Creative ideas such as dumping model context in the filesystem and letting the model manipulate it [324] can drastically improve performance in some tasks. Other benchmarks remain hardly solvable with current methods [338] and while perhaps not the most fun, I suspect smart scaffolding might be the most direct way forward.

As search becomes more agentic, evaluating systems rather than isolated models will become increasingly important.

8.4 Benchmarking Search Performance

After agentic systems and end-to-end evaluation, the next step forward in benchmarking is to veer away from benchmarks artificially constructed to be hard and multi-hop, such as BrowseComp [330] or OBLIQ-Bench [338], and instead evaluate search by its ability to improve decisions under uncertainty. Many practical search tasks are difficult to benchmark because relevance is underspecified: users search not merely to retrieve a document, but to update their beliefs and act.

Research Idea. Prediction markets offer a natural testbed for this idea. They provide dynamic, externally priced estimates about future events, such as elections, policy decisions, sports outcomes, company results, or breaking news. Since market prices aggregate dispersed information and are continuously updated, they can serve as a measurable proxy for the current public belief about an event [339–342]. A search system with genuine informational value should sometimes be able to improve on this baseline by finding deeper, fresher, or more decision-relevant evidence before it is fully reflected in the market.

Concretely, by holding the final decision-making model fixed and varying only the search system, each system could gather different sources about the same market, and the same downstream model would convert the gathered evidence into a calibrated forecast and trading decision. The edge over other market actors is reflected in the financial outcomes. The fact the market is dynamic and quickly patches information arbitrage weights the value of gathered information automatically. The evaluation target shifts from *Did the system retrieve the annotated document?* to *Did the system retrieve information that updates priors (beliefs) in a way that is possible to leverage?*

Given more flexibility, such environments could also become useful tests of source assessment, real-time reasoning, calibration, and more broadly general model intelligence in complex situations.

Search On

Intelligence is becoming a commodity, sold to consumers by trillion-dollar companies. Blindly trusting state-of-the-art LLM outputs is becoming an almost rational decision: as we improve modeling and search tooling, model responses are most often going to be correct and well grounded in verifiable sources.

Searching however, is often more than getting an exact answer to a specific question. To understand and truly value an answer, one must often first build prerequisite understanding, explore, reason, backtrack, synthesize... Other questions pop along the way. Bush says that we *search* by exploring the maze of human knowledge but to a curious mind, this twisted path is beautiful and interesting; it leads through side quests and down rabbit holes. Answers to why the brick road is yellow in the *Wizard of Oz* are surprisingly few Wikipedia links away from reading all about the Opium Wars.²

[338]: Tchuindjo et al. (2026), *OBLIQ-Bench: Exposing Overlooked Bottlenecks in Modern Retrievers with Latent and Implicit Queries*

[339]: Hayek (1945), *The Use of Knowledge in Society*

[340]: Wolfers et al. (2004), *Prediction Markets*

[341]: Wolfers et al. (2006), *Prediction Markets in Theory and Practice*

[342]: Berg et al. (2008), *Prediction Market Accuracy in the Long Run*

2: One of my favorite games I learnt from Professor Robert West consists in starting from a given Wikipedia article and navigating to another predetermined page using only in-article hyperlinks.

Searching is the gateway to knowledge, but the path stimulates curiosity and bootstraps creativity. Shiny search tools are invaluable to progress as a society, and search agents are clearly the future. Sometimes however, aimlessly digging through the disorganized pile of parchments in the library of Alexandria is still the best option. Search on!

"For mature thought there is no mechanical substitute"

– Vannevar Bush, *As We May Think* (1945)

APPENDIX

This glossary defines recurring technical terms as they are used throughout this thesis. Definitions are intentionally concise and reflect common usage in information retrieval and machine learning; citations are included where a term is tied to a particular method or formalization.

Information Retrieval

Information retrieval (IR). The study and practice of finding items relevant to an information need within a collection. In this thesis, the items are usually documents, passages, or document-page images, and the information need is expressed as a query.

Corpus, document, passage, and chunk. A *corpus* is the collection being searched. A *document* is one item in that collection, while a *passage* is a span of text. In this thesis, “passage” and “document” are sometimes used interchangeably for textual retrieval units. What counts as a document is partly a system-design choice: a book, page, paragraph, or passage may be indexed as one item.

Chunking divides a long document into smaller units, or *chunks*, that can be indexed and retrieved independently.

Information need, query, and relevance. An *information need* is the underlying information a user seeks; a *query* is the expression of that need supplied to a retrieval system. *Relevance* describes how well a candidate item satisfies the information need, not merely whether it contains the query words [4]. Relevance may be binary or graded and is task-dependent: a passage can be topically related to a query without containing the evidence needed to answer it.

[4]: Manning et al. (2008), *Introduction to Information Retrieval*

Indexing, retrieval, and reranking. *Indexing* preprocesses a corpus into a data structure that supports fast search. *Retrieval* uses that index to obtain an initial set of candidates. *Reranking* then applies a typically slower and more expressive model to reorder a limited candidate set.

Sparse, lexical, and dense retrieval. *Lexical retrieval* primarily matches explicit terms shared by queries and documents. Its representations are usually *sparse*: most coordinates, often associated with vocabulary terms, are zero. *Dense retrieval* uses learned, low-dimensional vectors whose coordinates are generally non-zero and can capture semantic similarity even when the query and document use different words.

Inverted index. A data structure composed of a dictionary of terms and, for each term, a *postings list* identifying the documents, and often positions, in which it occurs. It is the standard basis for efficient lexical retrieval at corpus scale [4].

[4]: Manning et al. (2008), *Introduction to Information Retrieval*

Nearest-neighbor search and vector index. Given a query vector, nearest-neighbor search finds the indexed vectors that are closest under a chosen distance or similarity function. Exact search compares the query with

[63]: Johnson et al. (2017), *Billion-scale Similarity Search with GPUs*

[35]: Spärck Jones (1972), *A Statistical Interpretation of Term Specificity and Its Application in Retrieval*

[4]: Manning et al. (2008), *Introduction to Information Retrieval*

[39]: Robertson et al. (1994), *Some Simple Effective Approximations to the 2-Poisson Model for Probabilistic Weighted Retrieval*
 [40]: Robertson et al. (1994), *Okapi at TREC-3*

every vector; approximate nearest-neighbor (ANN) methods use specialized data structures to trade a small amount of recall for much lower latency and memory use at scale. A *vector index* stores embeddings and supports this search efficiently [63].

Term frequency–inverse document frequency (TF–IDF). A family of term-weighting schemes that gives a term more weight when it occurs frequently in a document but less weight when it occurs in many documents [35]. For a corpus $\mathcal{D} = \{d_1, \dots, d_N\}$, a common form is

$$\text{tfidf}(t, d, \mathcal{D}) = \text{tf}(t, d) \text{idf}(t, \mathcal{D}), \quad \text{idf}(t, \mathcal{D}) = \log\left(\frac{N}{\text{df}(t)}\right),$$

where $\text{tf}(t, d)$ measures the frequency of term t in document d , and $\text{df}(t)$ is the number of corpus documents containing t . Many implementations use logarithmic scaling or smoothing. In the classic vector-space model, TF–IDF weights form query and document vectors that can be ranked using a similarity function such as cosine similarity [4].

Okapi BM25. A widely used lexical ranking function that combines inverse document frequency, saturating term frequency, and document-length normalization [39, 40]. Using the notation introduced in the history chapter, its score is

$$\text{BM25}(q, d) = \sum_{t \in q} \text{idf}(t) \frac{f_{t,d}(k_1 + 1)}{f_{t,d} + k_1 \left(1 - b + b \frac{|d|}{\text{avgdl}}\right)},$$

with

$$\text{idf}(t) = \log\left(\frac{N - \text{df}(t) + 0.5}{\text{df}(t) + 0.5}\right).$$

Here $f_{t,d}$ is the count of t in d , $|d|$ is the document length, avgdl is the corpus-average document length, $k_1 > 0$ controls term-frequency saturation, and $b \in [0, 1]$ controls length normalization.

Representations and Matching

Embedding. A learned numerical representation of an item as a vector or set of vectors. Text embeddings may represent tokens, passages, queries, or entire documents; visual embeddings may represent image patches or complete images. Retrieval models are trained so that geometrical relationships between embeddings are useful for matching.

Embedding space and latent space. An *embedding space* is the vector space in which embeddings are compared. A *latent space* is a more general learned representation space whose coordinates are not directly observed or manually assigned. In this thesis, “latent space” and “representation space” often refer to the internal learned geometry of representations exposed for downstream comparison or retrieval.

Similarity function. A function that assigns a matching score to two representations. Common choices include the dot product and cosine

similarity. Cosine similarity compares the angle between vectors,

$$\cos(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x}^\top \mathbf{y}}{\|\mathbf{x}\|_2 \|\mathbf{y}\|_2},$$

and therefore ignores their overall magnitudes.

Single-vector and multi-vector representation. A *single-vector* model compresses each query or document into one vector. A *multi-vector* model retains several vectors, commonly one per token or image patch, allowing a more fine-grained matching function at the cost of a larger index.

Pooling. An operation that aggregates several representations into fewer representations, often a single vector. *Mean pooling* averages vectors, *max pooling* retains the maximum according to a chosen aggregation criterion, and *special-token pooling* uses the representation of a designated token such as [CLS] or the final token. Token pooling can also group nearby or redundant token vectors to reduce the storage and computation required by multi-vector retrieval. In MaxSim late interaction, for example, the maximum is taken over document-token similarity scores for each query token, rather than over embedding dimensions.

Contextual embedding. An embedding whose value depends on surrounding input rather than representing an item in isolation. A contextual token embedding changes when the token's sentence changes; similarly, a contextual chunk embedding may incorporate information from other passages in the same document or from the wider corpus.

Bi-encoder (dual encoder). A model that encodes the query and document independently, often into single-vector embeddings. Document representations can therefore be computed offline and indexed, while retrieval reduces to efficient nearest-neighbor search. This architecture favors scalability but limits direct query–document interaction [15].

Cross-encoder. A model that processes a query and candidate document jointly, allowing self-attention across all query and document tokens before producing a relevance score. Cross-encoders are expressive but cannot precompute a query-independent document score, so they are generally used for reranking rather than exhaustive corpus search.

Late interaction. A retrieval architecture between bi-encoders and cross-encoders. Queries and documents are encoded independently into multi-vector representations, but their token- or patch-level vectors interact at scoring time. ColBERT's common *MaxSim* operator finds, for every query vector, the most similar document vector and sums these maxima [78]. This preserves offline document indexing while enabling finer matching than a single-vector similarity.

Contrastive learning. A training approach that makes representations of matched or relevant examples more similar and representations of unmatched examples less similar. In retrieval, a query and its relevant document form a positive pair, while other documents serve as negatives; *hard negatives* are non-relevant candidates that are difficult for the current model to distinguish from positives.

Positive, negative, and hard-negative examples. A *positive* is an example that should match the anchor input, such as a relevant document for a query. A *negative* should not match it. *Hard negatives* are non-relevant

[15]: Reimers et al. (2019), *Sentence-BERT: Sentence Embeddings using Siamese BERT- Networks*

[78]: Khattab et al. (2020), *ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT*

examples that nevertheless appear plausible to the model, for example because they are topically similar or highly ranked by another retriever. They generally provide a stronger training signal than randomly sampled negatives, but false negatives can teach the model to separate genuinely relevant pairs.

Contrastive loss. An objective used to implement contrastive learning. It increases the score of positive pairs relative to negative pairs, commonly by applying a softmax and cross-entropy over the candidates in a batch. A temperature parameter may control how sharply score differences affect the loss.

Late chunking. A method that encodes a long document before pooling token representations into individual chunk embeddings. Unlike independently encoding each chunk, this allows a chunk representation to incorporate information from surrounding document context [85].

[85]: Günther et al. (2024), *Late Chunking: Contextual Chunk Embeddings Using Long-Context Embedding Models*

Model Architectures

Token and tokenizer. A *token* is a discrete unit processed by a language model, such as a word, subword, character, punctuation mark, or special symbol. A *tokenizer* defines the vocabulary and the procedure that converts raw text into tokens and back into text.

Pretraining, fine-tuning, and inference. *Pretraining* learns general-purpose model parameters from a large and broad dataset, usually before a specific downstream task is fixed. *Fine-tuning* continues training those parameters, or a smaller set of added parameters, on task- or domain-specific data. *Inference* is the use of the trained model to encode, score, classify, or generate outputs without updating its parameters.

Transformer. A neural-network architecture built around attention mechanisms rather than recurrence. Transformer layers use self-attention to let each token combine information from other tokens, followed by position-wise feed-forward transformations [343]. Transformers can be organized as encoders, decoders, or encoder-decoder models and underpin most language and vision models discussed in this thesis.

[343]: Vaswani et al. (2017), *Attention Is All You Need*

Attention, self-attention, and causal attention. *Attention* computes a weighted combination of value vectors according to the compatibility of queries and keys. In *self-attention*, queries, keys, and values are derived from the same sequence. *Bidirectional* self-attention allows a token to use tokens on both sides, whereas *causal* attention masks future positions so that a token can depend only on earlier tokens.

Encoder and decoder. An *encoder* maps an input into contextual representations and commonly uses bidirectional attention. An autoregressive *decoder* predicts a sequence one token at a time using causal attention. An *encoder-decoder* model first encodes an input and then generates an output conditioned on those encoded representations.

Autoregressive model. A model that factorizes the probability of a sequence into successive next-token predictions. During generation, each new token is sampled or selected conditioned on the preceding tokens and is then appended to the context used to predict the next one.

Context window. The maximum amount of input, measured in tokens, that a model can process in one forward pass. Material outside this window must be discarded, compressed, or handled through mechanisms such as chunking or retrieval. A long context window increases capacity to process complete documents but also raises computation and memory costs.

Large and small language models (LLMs and SLMs). A language model assigns probabilities to token sequences and can generate text. Following the operational definition used in the CroissantLLM work, an LLM is pretrained on a large text corpus, supports text generation, and enables transfer through fine-tuning or prompting [344]. The cited definition proposes one billion pretraining tokens as a lower threshold; under these criteria, CroissantLLM is an LLM despite its relatively small parameter count compared with current frontier models. “Small language model” is a relative term for a lower-parameter language model designed, for example, for lower latency, memory use, or local inference; there is no universal parameter threshold separating SLMs from LLMs.

[344]: Rogers et al. (2024), *Position: Key Claims in LLM Research Have a Long Tail of Footnotes*

Vision Transformer (ViT) and image patch. A Vision Transformer divides an image into fixed-size regions called *patches*, projects them into vectors, and processes the resulting sequence with a transformer [107]. A patch embedding is therefore analogous to a text-token embedding, although it represents a spatial image region.

[107]: Dosovitskiy et al. (2020), *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*

Vision-language model (VLM). A model that processes both visual and textual inputs in a shared architecture or aligned representation spaces. VLMs may be generative, contrastive, or encoder-based; in this thesis they are frequently adapted to produce embeddings for image-text retrieval.

Modality alignment. Training that makes representations from different modalities, such as images and text, mutually compatible. Alignment can map paired inputs into nearby regions of a shared embedding space or teach a multimodal model to process one modality through representations expected by another. Contrastive image-text training is a common alignment method [100].

[100]: Radford et al. (2021), *Learning Transferable Visual Models From Natural Language Supervision*

Early fusion and dual-encoder multimodal models. An *early-fusion* model combines representations from different modalities inside a shared backbone, allowing cross-modal interaction throughout much of the network. A multimodal *dual encoder* processes each modality with a separate encoder and compares only their final representations. Dual encoders support efficient precomputation, while early fusion enables richer interaction at greater computational cost.

Quantization. The representation of model parameters, activations, or stored embeddings with lower-precision values than those used in standard floating-point computation. Quantization reduces memory, storage, and often latency, but aggressive compression can reduce model or retrieval quality.

Document AI and Evaluation

Optical character recognition (OCR). The process of converting text visible in an image or scanned document into machine-readable characters. OCR-based retrieval first extracts text, whereas visual document retrieval can index page images directly.

Visual document retrieval (VDR). Retrieval in which documents, usually pages, are represented from their visual appearance rather than exclusively from parsed text. This retains evidence carried by layout, typography, tables, figures, and spatial relationships, as well as by visible words.

Retrieval-augmented generation (RAG). A system pattern in which a retriever selects external evidence and a generative model produces an output conditioned on that evidence. The original RAG formulation combined a sequence-to-sequence generator with a learned dense Wikipedia retriever [87]; current usage also covers pipelines with lexical or hybrid retrieval, reranking, query rewriting, and iterative search.

[87]: Lewis et al. (2020), *Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks*

Parametric and non-parametric knowledge. *Parametric knowledge* is information encoded implicitly in a model's learned weights. *Non-parametric knowledge* is stored in an external resource, such as a document corpus or vector index, that can be searched or updated without retraining the model. RAG combines a parametric generator with access to non-parametric knowledge [87].

[87]: Lewis et al. (2020), *Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks*

Test collection and relevance judgment. A standard retrieval test collection contains a document corpus, a set of information needs represented as queries, and judgments identifying which documents are relevant to each need [4]. Retrieval metrics summarize performance with respect to these judgments, which act as the evaluation ground truth.

[4]: Manning et al. (2008), *Introduction to Information Retrieval*

Recall@ k . The fraction of all relevant items for a query that occur among the first k retrieved results. When a benchmark contains one known relevant item per query, Recall@ k is equivalently the fraction of queries for which that item appears in the top k .

Precision@ k . The fraction of the first k retrieved items that are relevant. Unlike Recall@ k , it does not account for relevant items that exist elsewhere in the corpus.

Mean reciprocal rank (MRR). For each query, reciprocal rank is the inverse of the rank of the first relevant result, or zero if no relevant result is retrieved. MRR averages this value over queries and therefore emphasizes placing at least one relevant result near the top of the ranking.

Average precision and mean average precision (AP and MAP). Average precision summarizes a ranking by averaging precision at the ranks where relevant items occur. Mean average precision averages AP over queries. It rewards retrieving all relevant items and placing them early, and is most directly applicable to binary relevance judgments.

Normalized discounted cumulative gain (nDCG@ k). A ranking metric for graded relevance. It accumulates relevance gains through rank k , discounts gains appearing lower in the ranking, and divides by the score of an ideal ordering. The normalized score lies between zero and

one, with one indicating an ideal ranking under the supplied relevance judgments.

Language-Model Training and Evaluation

Perplexity. An exponentiated average negative log-likelihood that measures how much probability a language model assigns to a token sequence. Lower perplexity means the model predicts the evaluated token sequence with greater confidence. Perplexities are not directly comparable across models that use different tokenizers or evaluation protocols.

Scaling law. An empirical relationship describing how model loss or downstream performance changes as a power-law-like function of factors such as parameter count, training data, or compute. Scaling laws can guide how a fixed training budget is allocated, but their predictions depend on the architecture, data distribution, and range over which they were fitted [255, 281].

Zero-shot and few-shot learning. In a *zero-shot* evaluation, a model performs a task without task-specific demonstrations in its input. In a *few-shot* evaluation, the prompt contains a small number of example input–output pairs. These terms describe the evaluation context and do not imply that the model has never encountered related tasks or data during training.

[255]: Hoffmann et al. (2022), *Training Compute-Optimal Large Language Models*

[281]: Kaplan et al. (2020), *Scaling Laws for Neural Language Models*

CroissantLLM-Specific Terms

Effective capacity ratio. The *effective parameters* of a multilingual model for one language are the number of non-embedding parameters a comparable monolingual model would require to match its performance in that language [209]. The effective capacity ratio divides this estimate by the multilingual model’s number of non-embedding parameters. It measures how much of the multilingual model’s capacity appears effectively available to the language in question. In the joint scaling-law experiments reported for CroissantLLM, models trained on equal proportions of English and French reach an effective capacity ratio above 82% for French, suggesting substantial capacity sharing between the two languages.

[209]: Fernandes et al. (2023), *Scaling Laws for Multilingual Neural Machine Translation*

Tokenizer fertility. The average number of subword tokens produced per tokenized word [280]. Lower fertility means that a tokenizer represents the same text with fewer tokens; its theoretical lower bound is one token per word under a word-based measurement.

[280]: Rust et al. (2021), *How Good is Your Tokenizer? On the Monolingual Performance of Multilingual Language Models*

The Thesis in Twenty Terms

B

As a small application of the lexical retrieval methods discussed in this thesis, this chapter gives a compact view of the vocabulary that most distinguishes each main-body chapter. The lists below contain the twenty terms with the highest term frequency-inverse document frequency (TF-IDF) weight in each chapter [35]. TF-IDF and other recurring technical terms are defined in Appendix A.

[35]: Spärck Jones (1972), *A Statistical Interpretation of Term Specificity and Its Application in Retrieval*

The computation treats each of the eight main-body chapters as one document. The text is normalized by removing comments, LaTeX commands, citations, references, mathematics, English stopwords, and common document scaffolding. Words are converted to lowercase unaccented forms, and unigram terms of at least three characters are retained. These exclusions are deliberately limited to generic structural or formatting artifacts rather than topic-specific words. For a term t in chapter d , the weight is

$$\text{tfidf}(t, d) = \frac{f_{t,d}}{\sum_{t'} f_{t',d}} \left(\log \frac{1 + N}{1 + n_t} + 1 \right),$$

where $N = 8$ is the number of chapters and n_t is the number of chapters containing t . The smoothing avoids zero weights, and alphabetical order breaks exact score ties.

B.1 Introduction

retrieval, information, document, visual, search, thesis, models, croissantllm, author, systems, documents, model, work, series, colpali, knowledge, language, performance, efforts, vidore.

B.2 A Brief History of Information Retrieval

retrieval, document, documents, ranking, information, query, systems, term, probabilistic, models, search, sparse, embedding, lexical, modern, became, early, one, evidence, vectors.

B.3 ColPali

retrieval, document, visual, image, models, paligemma, colpali, performance, patches, model, embeddings, query, text, documents, unstructured, systems, tasks, vidore, vision, training.

B.4 ViDoRe Benchmarks: V2 & V3

visual, queries, query, retrieval, grounding, bounding, vidore, annotators, annotation, human, textual, generation, answer, document, boxes, pro, summaries, pages, gemini, colembed.

B.5 ModernVBERT

image, retrieval, models, visual, document, training, modality, encoder, model, alignment, performance, tasks, contrastive, bidirectional, vision, attention, causal, text, resolution, tokens.

B.6 Contextual Document Embeddings

chunks, chunking, chunk, retrieval, contextual, document, conteb, method, training, information, models, embedding, context, documents, performance, late, insent, approaches, modernbert, queries.

B.7 CroissantLLM

model, data, french, croissantllm, models, english, training, performance, tokens, language, llama, code, bloom, dataset, transparency, corpus, license, translation, trained, shot.

B.8 Personal Perspectives on Search

memory, search, retrieval, information, document, model, associative, parametric, working, evidence, systems, stores, external, links, operation, market, llm, attention, index, agentic.

Reproducibility Statement

For transparency, reproducibility and to foster future work, we release our training data, model checkpoints (adapters), entire codebase, and complete evaluation benchmark under MIT licenses as detailed in the main paper. We also host a public ViDoRe leaderboard¹ to foster concurrent work in the space. The supplementary material further details training configurations for our models (also specified in HuggingFace model repositories), and dives into the process we used to generate synthetic data, how latency computations are performed, as well as provides further detailed evaluation results.

1: <https://huggingface.co/spaces/vidore/vidore-leaderboard>

Acknowledgements

This work is partially supported by Illuin Technology, and by a grant from ANRT France. This work was performed using HPC resources from the CINES ADAstra through Grant 2024-AD011015443 and from IDRIS with grant 2024-AD011015724R1. We extend our warm thanks to Jonathan Dong, Caio Corro, Victor Pellegrain and Ender Konukoglu for their valuable feedback on the paper.

C.1 Benchmark Datasets

C.1.1 Academic Datasets

DocVQA [120] includes collected images from the UCSF Industry Documents Library. Questions and answers were manually annotated.

InfoVQA [121] includes infographics collected from the Internet using the search query “*infographics*”. Questions and answers were manually annotated.

TAT-DQA [123] is a large-scale Document VQA dataset that was constructed from publicly available real-world financial reports. It focuses on rich tabular and textual content requiring numerical reasoning. Questions and answers were manually annotated by human experts in finance.

arXivQA [122] is a VQA dataset based on figures extracted from arXiv publications. The questions were generated synthetically using GPT-4 Vision.

TabFQuAD (Table French Question Answering Dataset) is designed to evaluate TableQA models in realistic industry settings. We create additional queries to augment the existing human-annotated ones using the same method described in subsection C.1.2.

[120]: Mathew et al. (2020), *DocVQA: A Dataset for VQA on Document Images*

[121]: Mathew et al. (2021), *InfographicVQA*

[123]: Zhu et al. (2022), *Towards Complex Document Understanding By Discrete Reasoning*

[122]: Li et al. (2024), *Multimodal ArXiv: A Dataset for Improving Scientific Comprehension of Large Vision-Language Models*

C.1.2 Practical Datasets

Methodology. Creating a relevant retrieval dataset close to real use cases is a major challenge as the dataset needs to be both sufficiently large for effective fine-tuning and sufficiently diverse to cover a broad range of modalities (full text, tables, charts, ...), domains (industry, healthcare, ...), and query-document interactions (extractive questions, open-ended questions, ...). Our approach to building this dataset involves several steps: (1) we use a web crawler to collect publicly available documents on various themes and sources, (2) we convert these PDFs into a series of images, one per page, and (3) we generate queries related to each image using a VLM.

Web-Crawler. We implemented a web crawler to efficiently collect large volumes of documents related to a given topic. The crawler is seeded with a user-defined query (e.g. “artificial intelligence”) and then uses GPT-3.5 Turbo to brainstorm related topics and subtopics. This query augmentation strategy aims at both broadening and deepening the search. GPT-3.5 Turbo is further used to generate diverse search queries from each subtopic. This query set is then consumed by a pool of parallel workers whose job is to fetch the associated most relevant documents. We use SerpAPI² along with a filetype filter (PDF documents only) to programmatically scrape Google Search rankings. Each file is hashed and stored in a Bloom filter [345] shared among workers to avoid duplicate documents in the final corpus. Unique scraped files are downloaded, and inserted into a SQLite database along with additional metadata.

Datamix. Using the web crawler, we collected approximately 100 documents for each of the following four seeds: “energy”, “government reports”, “healthcare industry”, and “artificial intelligence”. These seeds were meticulously hand-picked to align with real-use cases for retrieval models and visually rich pages. We also removed all documents containing any private information.

Query Generation. To increase the efficiency of our query generation scheme and to limit API calls, we generate at most 3 questions per image. From all the documents collected, we randomly sample 10,000 images per theme and call Claude-3 Sonnet with the following prompt:

You are an assistant specialized in Multimodal RAG tasks.

The task is the following: given an image from a pdf page, you will have to generate questions that can be asked by a user to retrieve information from a large documentary corpus. The question should be relevant to the page, and should not be too specific or too general. The question should be about the subject of the page, and the answer needs to be found in the page.

Remember that the question is asked by a user to get some information from a large documentary corpus that contains multimodal data. Generate a question that could be asked by a user without knowing the existence and the content of the corpus. Generate as well the answer to the question, which should be found in the page. And the format of the answer should be a list of words answering the question. Generate at most THREE pairs of questions and answers per page in

2: <https://serpapi.com/>

[345]: Bloom (1970), *Space/time trade-offs in hash coding with allowable errors*

a dictionary with the following format, answer ONLY this dictionary NOTHING ELSE:

```
"questions": [ "question": "XXXXXX", "answer": ["YYYYYY"], "question": "XXXXXX", "answer": ["YYYYYY"], "question": "XXXXXX", "answer": ["YYYYYY"], ] where XXXXXX is the question and ['YYYYYY'] is the corresponding list of answers that could be as long as needed.
```

Note: If there are no questions to ask about the page, return an empty list. Focus on making relevant questions concerning the page. Here is the page:

Human Validation. We manually validate every single synthetically created query in ViDoRe to ensure quality, query relevance, and consistency with the benchmark objective of evaluating retrieval in practical industrial settings. During this step, we randomly assign document-pair queries to 4 volunteer annotators and instruct them to filter out queries that do not fit the above-listed criteria. We also instruct annotators to flag any documents they deem to contain PII information or content not suited for an academic benchmark. No flag was raised during the entirety of the process, validating our prior PDF collection strategy. 100 queries per topic are collected in this manner. Annotators are colleagues and collaborators of the authors who volunteered to help. Each annotator spent approximately 3 hours filtering the larger query set down to 100 high-quality queries per topic.

C.1.3 Training Dataset

The statistics of the train set are given in the following table. The creation of the train set follows the same methodology as in subsection C.1.2. We made sure that a PDF document cannot have pages in both the training set and the test set to prevent data leakage and that there are no duplicate documents in each split.

Dataset Split	Split Size	Language	Domain
DocVQA	39,463	English	Scanned documents from UCSF Industry
InfoVQA	10,074	English	Infographics scraped from the web
TATDQA	13,251	English	High-quality financial reports
arXivQA	10,000	English	Scientific Scientific Figures from arXiv
Scraped PDFs	45,940	English	Varied PDFs from 3885 distinct URL domains
TOTAL	118,695	English-only	Mixed

Table C.1: Details on the different splits in the dataset used to train ColPali.

C.2 Implementation details

C.2.1 Codebase

The codebase is written in PyTorch³ and leverages HuggingFace tooling for model implementations and trainers⁴.

3: <https://pytorch.org/>

4: <https://huggingface.co>

C.2.2 Hyperparameters

Hyperparameters are tuned on a validation split composed of 2% of the training dataset. We find bi-encoder methods to be more sensible to learning rate variations than late interaction-based models and achieve the best performance for all models with a learning rate of $5e-5$. We experiment with LoRA rank and α values and do not notice particular improvements past $r = \alpha = 32$. Per-device batch sizes are kept small due to long sequence lengths that complicate scaling past $b = 4$. We simulate larger batch sizes with multi-GPU training and train with a total batch size $b = 32$ with no accumulation, for 1 epoch on our training set.

C.2.3 Embedding size

Minimizing storage footprint can be essential to industrial retrieval systems if databases contain millions of documents. With this criterion in view, we have compared the embedding sizes of the models in our study. As shown in Table C.2, ColPali’s embedding size is an order of magnitude larger than BM25 and two orders of magnitude larger than BGE-M3. However, in practical scenarios, pooling multi-vector embeddings by centroid cluster, or quantizing embeddings to binary representations⁵ can reduce storage costs by two orders of magnitude [132] with minimal performance hits, and make storage costs competitive with other systems.

5: <https://blog.vespa.ai/scaling-colpali-to-billions/>

[132]: Santhanam et al. (2022), *PLAID: An Efficient Engine for Late Interaction Retrieval*

Table C.2: Comparison of the embedding sizes for the DocVQA test set from ViDoRe w.r.t. different retrieval models. The *mean ± std* size is given for the sparse embeddings. In general multiple vectors (2-5) per page are used for BGE-M3 and BM25.

Model	Embedding size (KB)
BGE-M3	8.60
BM25 (dense emb.)	3.00
BM25 (sparse emb.)	1.56 ± 0.51
ColPali (float16)	257.5

C.2.4 Latency computations

To ensure comparison fairness, the latencies of the different retrieval systems shown in Figure 3.2 are measured on the same g2-standard-8 GCP VM with an NVIDIA L4 GPU. Document pages are embedded using the highest settings of Unstructured with captioning (see subsection 3.3.2). SigLIP and ColPali are both loaded with `bf16` parameter dtypes. The reported times in Table C.3 are the average per-page latencies for each indexing operation on 1000 randomly chosen documents across all splits of the ViDoRe benchmark test set. A batch size of 8 was used for the BGE-M3 model used with Unstructured, and a batch size of 4 was used for SigLIP and ColPali.

Table C.3: Page-level latencies for document indexing using various retrieval systems. SigLIP and ColPali are much faster than Unstructured because they don’t require the layout detection, OCR, and captioning operations.

Indexing operation	Latency (s)		
	Unstructured	SigLIP	ColPali
Layout detection	0.81	NA	NA
OCR	2.67	NA	NA
Captioning	3.71	NA	NA
Page encoding	0.03	0.12	0.39
Total	7.22	0.12	0.39

C.2.5 Captioning

Examples of captions generated for visually rich document chunks with Claude-3 Sonnet are shown in Figure C.2 and Figure C.1. The prompt used for generating the description is the following:

You are an assistant specialized in document analysis. Given a table or a figure, you have to provide a detailed summary of the content in maximum 3000 characters. Your summary should be qualitative and not quantitative. Here is the table/figure to analyze: [image]. Answer ONLY with the caption of the table/figure.

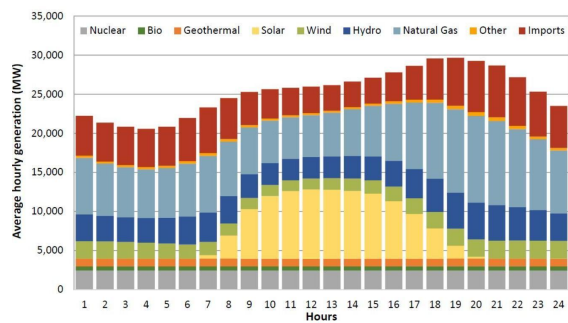


Figure C.1: Example from the “Energy” test set.

Caption: The image depicts the hourly energy generation profile, illustrating the contributions of various energy sources over 24 hours. The data is presented as a stacked bar chart, with the x-axis representing the hours of the day from 1 to 24, and the y-axis showing the average hourly generation in MW. The bars are segmented into different colors, each representing a distinct energy source: nuclear, bio, geothermal, solar, wind, hydro, natural gas, and other imports. The chart provides insights into the temporal variations in energy generation across different sources, highlighting the interplay between baseload and intermittent sources throughout the day.

Document	System of Record?				
	County Servers	Project View	OnBase	CGI Advantage Financial System	Purchasing Department
Project plan/scope	W	T			
Project budgets	W	T	W	T	
Schedules/timeline (design)	W	T	W	T	
Schedules/timeline (construction)	W	T	W	T	
Executed contracts			W	T	W
Purchase orders			W	T	W
Solicitation advertisements			W	T	W
Proposal submissions			W	T	W
Selection committee evaluation documents			W	T	W
Independent cost estimates		W	W	T	W
Negotiation details			W	T	W
Notice of award			W	T	W
Notice to proceed			W	T	W
Invoices/applications for payment (“AFP”)		W	T	W	W
Change order requests			W	T	W
Approved change orders		T	W	T	W
Requests for Information (RFI)	W	T			W
Bid submissions			W		W
Shop drawings	W	T		T	
Photographs	W	T		T	
As-builts	W	T		T	
Operating manuals	W	T		T	
Maintenance schedules	W	T		T	
Warranties	W	T		T	
Daily diaries	N/A	T			
Weekly meeting minutes	W	T			
Monthly progress reports	W	T			
Allowance usage logs	W	T			
Punch lists	W	T			

Figure C.2: Example from the “Government Reports” test set.

Caption: The image shows a table titled “System of Record” which outlines the different types of documents or records maintained across various systems or departments within an organization related to project management and construction. The rows list documents like project plans, budgets, schedules, contracts, purchase orders, invoices, change requests, bid submissions, drawings, manuals, meeting minutes, and reports. The columns indicate the system or department responsible for maintaining each record, such as County Servers, Project View, OnBase, CGI Advantage Financial System, and Purchasing Department. The table uses “W” and “T” markers to denote which system or department serves as the primary source (writer) or storage location (trailer) for each type of document.

C.3 Additional results

C.3.1 Other Metrics

Table C.4: Comprehensive evaluation of baseline models and our proposed method on ViDoRe. Results are presented using Recall@1 metrics. Text-only metrics are not computed for benchmarks with only visual elements.

	ArxivQ	DocQ	InfoQ	TabF	TATQ	Shift	AI	Energy	Gov.	Health.	Avg.
Unstructured text-only											
BM25	-	26.6	-	-	34.6	45.0	86.0	70.0	68.0	74.0	-
BGE-M3	-	22.8 _{↓3.8}	-	-	26.1 _{↓8.5}	51.0 _{↑6.0}	81.0 _{↓5.0}	72.0 _{↑2.0}	67.0 _{↓1.0}	77.0 _{↑3.0}	-
Unstructured + ocr											
BM25	26.7	28.9	54.0	30.4	50.0	52.0	86.0	77.0	74.0	80.0	55.9
BGE-M3	28.1 _{↑1.4}	22.9 _{↓6.0}	53.8 _{↓0.2}	55.7 _{↑25.3}	38.6 _{↓11.4}	56.0 _{↑4.0}	82.0 _{↓4.0}	79.0 _{↑2.0}	76.0 _{↑2.0}	83.0 _{↑3.0}	57.5 _{↑1.6}
Unstructured + captioning											
BM25	35.5	30.2	61.5	24.3	49.0	47.0	79.0	76.0	75.0	81.0	55.9
BGE-M3	29.3 _{↓6.2}	26.0 _{↓4.2}	62.1 _{↑0.6}	58.6 _{↑34.3}	30.6 _{↓18.4}	55.0 _{↑8.0}	80.0 _{↑1.0}	78.0 _{↑2.0}	69.0 _{↓6.0}	83.0 _{↑2.0}	57.2 _{↑1.3}
Contrastive VLMs											
Jina-CLIP	19.4	7.3	26.7	12.5	1.6	2.0	11.0	13.0	15.0	17.0	12.6
Nomic-vision	10.4	6.7	22.1	9.6	1.6	0.0	9.0	9.0	7.0	13.0	8.8
SigLIP (Vanilla)	34.2	21.3	51.8	46.1	17.9	13.0	50.0	51.0	47.0	65.0	39.7
Ours											
SigLIP (Vanilla)	34.2	21.3	51.8	46.1	17.9	13.0	50.0	51.0	47.0	65.0	39.7
BiSigLIP (+fine-tuning)	49.2 _{↑15.0}	23.8 _{↑2.5}	59.0 _{↑7.2}	52.1 _{↑6.0}	20.7 _{↑2.8}	16.0 _{↑3.0}	62.0 _{↑12.0}	61.0 _{↑10.0}	55.0 _{↑8.0}	72.0 _{↑7.0}	47.1 _{↑7.4}
BiPali (+LLM)	46.4 _{↓2.8}	20.0 _{↓3.8}	54.6 _{↓4.4}	63.2 _{↑11.1}	20.4 _{↓0.4}	34.0 _{↑18.0}	59.0 _{↓3.0}	45.0 _{↓16.0}	57.0 _{↑2.0}	56.0 _{↓16.0}	45.6 _{↓1.5}
ColPali (+Late Inter.)	72.4 _{↑26.0}	45.6 _{↑25.6}	74.6 _{↑20.0}	75.4 _{↑12.1}	53.1 _{↑32.7}	55.0 _{↑21.0}	93.0 _{↑34.0}	85.0 _{↑40.0}	85.0 _{↑28.0}	88.0 _{↑32.0}	72.7 _{↑27.1}

C.3.2 Model Variants

Table C.5: Benchmark scores for the “negative results” and various ablations on ViDoRe; ColPali for reference. Results are presented using nDCG@5 metrics. Text-only metrics are not computed for benchmarks with only visual elements.

	ArxivQ	DocQ	InfoQ	TabF	TATQ	Shift	AI	Energy	Gov.	Health.	Avg.
ColSigLIP (PaliGemma)	3.1	3.0	5.1	6.2	2.5	1.0	3.4	3.4	2.3	2.2	3.2
BiSigLIP (PaliGemma)	18.5	14.6	33.4	39.5	16.1	5.2	27.6	32.6	36.6	35.7	26.0
ColSigLIP (Original)	2.6	2.2	2.3	5.7	1.8	1.0	2.6	4.1	1.4	1.5	2.5
ColPali (No Q.A. Tokens)	80.4	53.2	82.4	77.4	65.7	63.4	97.0	89.9	93.6	92.4	79.6
ColPali (Docmatix)	71.3	48.0	80.0	83.9	59.1	73.8	95.7	93.8	92.5	93.1	79.1
ColPali (224)	71.0	37.4	62.3	65.7	28.6	20.4	65.7	66.8	73.9	73.0	56.5
ColPali (Vision Trained)	78.8	53.9	81.3	81.7	64.4	70.6	95.3	91.7	93.5	94.7	80.6
ColPali (No Pairwise)	79.0	53.0	82.1	85.3	63.2	66.2	94.9	88.9	92.7	92.1	79.7
ColPali (+TabFQuAD training)	77.6	54.7	82.6	86.5	65.4	73.9	94.8	92.4	94.2	94.8	81.7
ColIdeFics2 (64)	73.6	48.0	82.4	81.6	63.0	57.2	95.5	86.9	86.6	91.2	76.6
ColQwen2 (768)	86.4	56.2	89.8	88.7	75.2	85.7	98.8	94.8	93.6	97.3	86.6
ColPali (Reference: 448)	79.1	54.4	81.8	83.9	65.8	73.2	96.2	91.0	92.7	94.4	81.3

C.4 More similarity maps

In Figure C.3, *ColPali* assigns a high similarity to all patches with the word “Kazakhstan” when given the token <_Kazakhstan>. Moreover, our model seems to exhibit world knowledge capabilities as the patch around the word “Kashagan”—an offshore oil field in Kazakhstan—also shows a high similarity score.

It is also interesting to highlight that both this similarity map and the one displayed in Figure 3.4 showcase a few white patches with high similarity scores. This behavior might first seem surprising as the white patches should not carry a meaningful signal from the original images. We believe the vectors associated with these patches share a similar role with the ViT registers [346], i.e. these patches were repurposed for internal computations and stored the global information from the whole image.

[346]: Darcet et al. (2023), *Vision Transformers Need Registers*

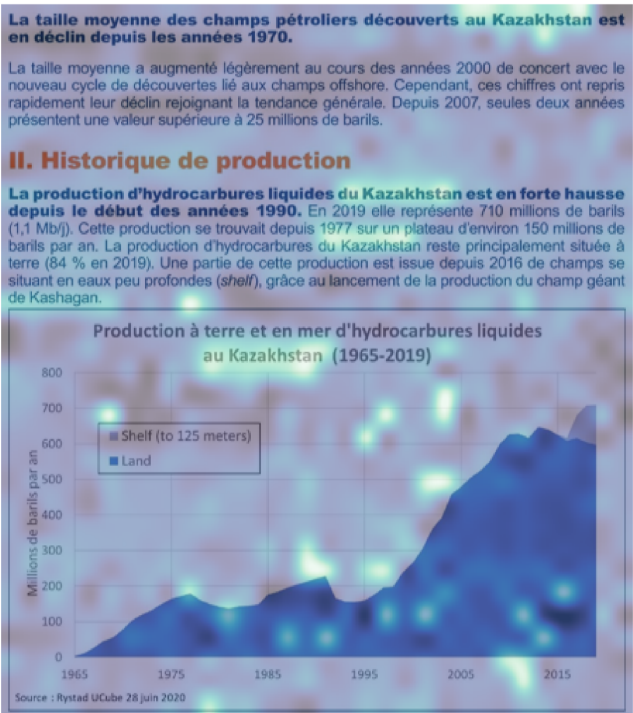


Figure C.3: Similarity of the image patches w.r.t. the underlined token in the user query. This example is from the *Shift* test set.

Query: "Quelle partie de la production pétrolière du Kazakhstan provient de champs en mer ?"

C.5 Model glossary

SigLIP

SigLIP (Sigmoid Loss for Language Image Pre-Training) builds upon CLIP (Contrastive Language-Image Pretraining)—a foundational model that aligns images and text by maximizing the similarity between correct image-text pairs while minimizing it for incorrect ones, leveraging a contrastive loss [101]. Unlike CLIP [100], which applies the softmax function to the logits, SigLIP uses the sigmoid activation function. This innovation eliminates the need for a global view of all pairwise similarities between images and texts within a batch, enabling more flexible batch size scaling (up to 1M items per batch, with an effective optimal batch size of 32k). This approach allows SigLIP to achieve state-of-the-art performance in zero-shot image classification tasks.

PaliGemma

PaliGemma is a 3B-parameter vision-language model. It integrates the SigLIP vision encoder with a Gemma-2B language decoder, connected via a multimodal linear projection layer [206]. The model processes images by segmenting them into a fixed number of Vision Transformer [107] tokens, which are prepended to an optional text prompt.

A distinguishing feature of PaliGemma is its operation as a Prefix-Language Model (Prefix-LM). This design ensures full attention between image tokens and the user-provided input (prefix) while generating

[101]: Zhai et al. (2023), *Sigmoid Loss for Language Image Pre-Training*

[100]: Radford et al. (2021), *Learning Transferable Visual Models From Natural Language Supervision*

[206]: Lucas Beyer* et al. (2024), *PaliGemma*

[107]: Dosovitskiy et al. (2020), *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*

outputs auto-regressively (suffix). This architecture allows image tokens to access the task-specific query during processing, facilitating more effective task-dependent reasoning.

PaliGemma was trained in four stages: unimodal pretraining with existing components, extended multimodal pretraining, short high-resolution pretraining, and task-specific fine-tuning.

ColBERT

[78]: Khattab et al. (2020), *ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT*

ColBERT (Contextualized Late Interaction over BERT) is a retrieval model designed to balance speed and effectiveness in information retrieval tasks [78]. Traditional retrieval models are typically categorized based on their type of interaction: either processing queries and documents independently for efficiency (bi-encoders) or jointly to capture rich contextual relationships (cross-encoders). ColBERT combines the advantages of both approaches through a novel late interaction mechanism.

Queries and documents are encoded separately using BERT, enabling offline pre-computation of document representations for scalability. Instead of pooling embeddings into a single vector, ColBERT retains token-level embeddings and employs a MaxSim operator to compute fine-grained similarity scores. For each query token, the model determines the maximum similarity with document tokens, summing these scores to compute relevance.

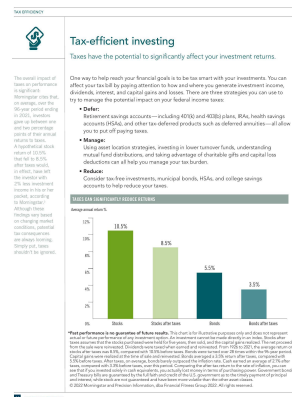
[59]: Devlin et al. (2019), *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*

This architecture preserves the contextual richness of deep language models while significantly improving computational efficiency. By delaying the interaction step, ColBERT supports vector similarity indexing, facilitating end-to-end retrieval from large collections without prohibitive costs. Empirical evaluations on passage search datasets demonstrate that ColBERT achieves competitive effectiveness compared to existing BERT-based models [59], while executing queries orders of magnitude faster and with drastically reduced computational requirements.

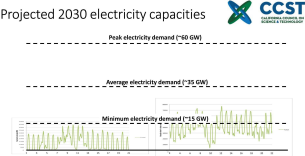
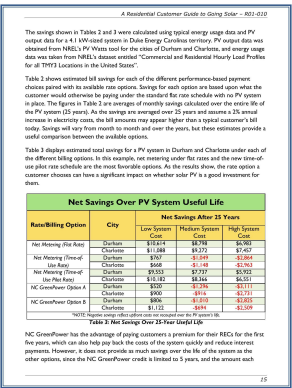
C.6 Examples from the ViDoRE benchmark

Energy

Query: What types of accounts or products allow investors to defer paying taxes?



Query: What is the estimated total savings for a PV system in Durham electricity demand in California for under the net metering (flat rate) the year 2030?

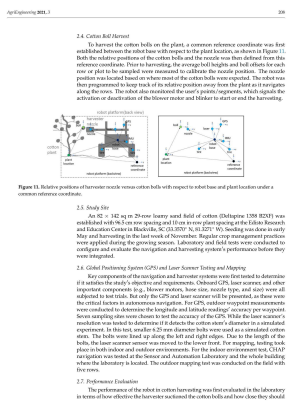


Artificial Intelligence

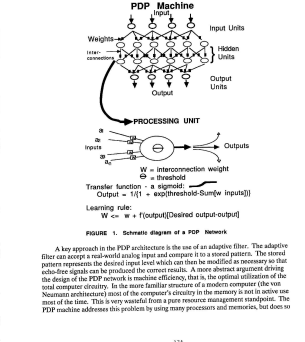
Query: What are some common out-of-office areas targeted by TAIL for different age groups?



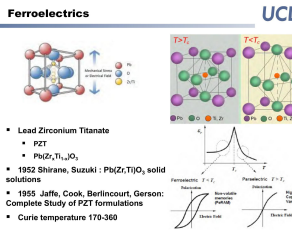
Query: What did the robot monitor to determine when to activate or deactivate the blower motor and blinker?



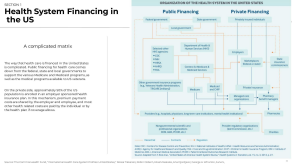
Query: What is the key approach used in the PDP architecture?



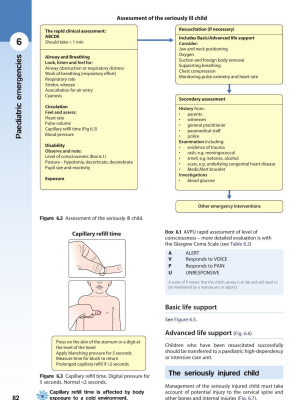
Query: What is the chemical formula for the ferroelectric material Lead Zirconium Titanate (PZT)?



Query: What government entities are involved in public financing for healthcare in the US?



Query: What does the AVPU scale stand for in assessing the level of consciousness of a seriously ill child?

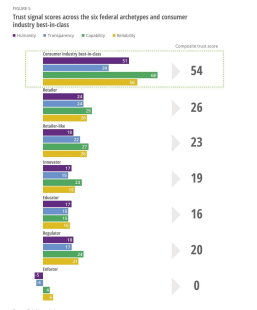
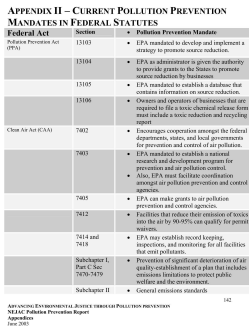


Government Reports

Query: What are some mandates for the EPA under the Pollution Prevention Act?

Query: What is the strategy of KPMG Hazem Hassan?

Query: What is the trust signal score for the consumer industry best-in-class archetype?

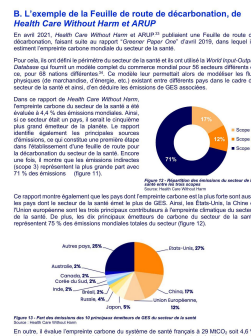
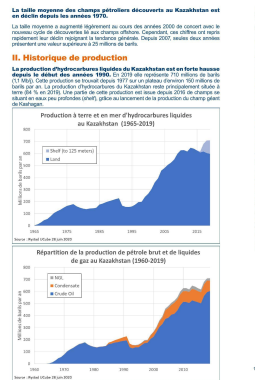
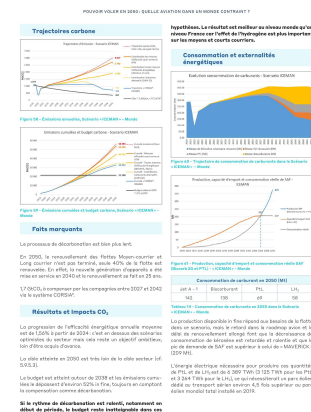


Shift

Query: Selon le graphique, quelle est la capacité d'import et la consommation réelle de carburants SAF (biocarburants durables pour l'aviation) prévues en 2050 ?

Query: Quelle partie de la production pétrolière du Kazakhstan provient de champs en mer ?

Query: Quels sont les pays ayant la plus grande part des découvertes cumulées de pétrole brut en 2020 (en milliers de barils, hors découvertes cumulées) ?



ViDoRe v3 Appendix

D

D.1 Dataset examples

Domain: **Pharmaceuticals**
 Query id: **17**
 Query type: **Extractive**
 Query format: **Question**

Query: According to FDA guidelines, what is the required format for displaying the strength of a small volume parenteral drug on its label?

Answer: According to FDA guidelines, the strength of a small-volume parenteral drug must be displayed as the quantity per total volume (e.g., 2 g/20 mL), immediately followed in parentheses by the quantity per milliliter (e.g., 100 mg/mL).

Domain: **Finance (FR)**
 Query id: **32**
 Query types: **Extractive, numerical**
 Query format: **Keyword**

Query: taux de produits fabriqués en France Hermès 2023

Answer: Le taux de produits d'Hermès fabriqués en France est de 74 % en 2023.

Domain: **Human Resources**
 Query id: **0**
 Query type: **Multi-hop**
 Query format: **Question**

Query: What impact did the shifts in temporary contract usage from 2018 to 2023 have on the difference in employment stability between mobile EU workers and citizens of the host countries?

Answer: From 2018 to 2023, the job stability gap between mobile EU workers and nationals narrowed. The use of temporary contracts declined more for EU movers (from 19% to 14%) than for nationals (14% to 11%), reducing the difference between the two groups from 5 to 3 percentage points.

Figure D.1: Examples from the ViDoRe V3 datasets. The sample pages illustrate varied query types and visually rich document layouts across domains, reflecting realistic retrieval scenarios.

D.2 Supplementary benchmark details

Domains Table D.2 details the type of documents used in each corpus as well as several statistics.

Query type and format descriptions Table D.1 describes the types and formats of the queries, while Figure D.2 gives details about query type intersection frequency.

Query type by generation method Query type distributions by generation method (Figure D.3) confirm that open-ended queries dominate synthetic queries as the synthetic pipeline attributed more weight to this type, while extractive queries dominate human-image queries since they are more naturally chosen by annotators.

D.3 Annotator pool and training details

Annotator Pool and Selection. Annotation was conducted by a curated pool of 76 annotators who were selected based on having: (1) a bachelor’s degree or higher in the relevant domain, (2) professional experience in the domain, (3) native-level language proficiency as required by task, and (4) prior experience with RAG, retrieval, or VQA annotation projects. Quality control was performed by 13 senior annotators with enhanced domain knowledge and extensive annotation experience, with project oversight provided by data leads with multiple years of experience in human data generation.

Training and Pilot Phase. The annotation process began with a comprehensive onboarding phase where annotators received task-specific training using gold-standard examples. For each domain, a pilot of several hundred tasks was conducted with 100% quality control coverage and multiple annotators per task. During this phase, data leads and the research team continuously evaluated annotations, provided clarifications, and refined guidelines. Inter-annotator agreement and time-per-task baselines were calculated to establish ongoing evaluation benchmarks. The pilot concluded upon validation of both data quality and guideline effectiveness.

D.4 Supplementary agreement metrics

Pages were pre-filtered by a VLM before human annotation; as most pages shown to annotators were likely relevant, this created a skewed class distribution. This prevalence imbalance causes traditional chance-corrected metrics like Krippendorff’s Alpha to appear paradoxically low even when annotators genuinely agree, as inflated expected chance agreement penalizes the score. To address this, we report 2 complementary metrics: Krippendorff’s Alpha (ordinal) as the standard measure and Gwet’s AC2 which remains stable under prevalence skew. Overall, annotators achieved $\alpha = 0.469$, $AC2 = 0.760$. The divergence between Alpha and AC2/Weighted Agreement is expected given the pre-filtered data and confirms substantial agreement despite the skewed distribution.

D.5 Supplementary retrieval details

Retriever model reference Table D.4 lists the retriever models evaluated in this work, along with their HuggingFace model names and citations.

Table D.1: Taxonomy of query types and formats used in ViDoRe V3. Definitions and representative examples are provided for each category.

Category	Definition	Example
<i>Query Types</i>		
Open-ended	Seeks explanatory or descriptive information that requires synthesis.	What drives the rise in women’s workforce involvement in EU nations?
Extractive	Requires the retrieval of a specific piece of information.	Bank of America preferred stock MM dividend rate
Compare Contrast	Mandates a comparison between multiple entities or data points.	Explain the factors contributing to the reduction in R2R rates for ANDAs.
Boolean	Poses a yes/no question necessitating multi-step reasoning.	Did JPMorganChase execute more than half of its planned repurchase program?
Numerical	Asks for a specific quantitative value that must be derived or calculated.	percentage increase in Morgan Stanley revenue from 2023 to 2024
Multi-hop	Requires integrating information from multiple sections or sources.	Summarize the steps involved in error reporting in ISMP’s MERP.
Enumerative	Requests a list of all instances sharing a common property.	Specify the ISCO codes used to define domestic workers in the EU.
<i>Query Formats</i>		
Question	An interrogative sentence.	What was Citigroup’s net interest margin in 2024?
Keyword	A non-verbal phrase or set of terms.	female employment rate European Union 2023
Instruction	A directive specifying a task.	Identify the use case of a drill point gauge.

Monolingual performance Tables D.5 and D.6 present the monolingual performance of our models, where retrieval is conducted using language-matched queries and documents for English and French, respectively.

Dataset	α (ord)	Gwet’s AC2
Computer science	0.467	0.809
Energy	0.463	0.714
Finance (EN)	0.514	0.798
Finance (FR)	0.320	0.736
H.R.	0.413	0.793
Industrial Maintenance	0.496	0.740
Telecom	0.464	0.772
Nuclear	0.389	0.794
Pharma	0.478	0.755
Physics	0.213	0.334
Overall	0.469	0.760

Table D.3: Inter-annotator agreement for relevance ratings by dataset. Per-dataset agreement metrics summarize consistency in page relevance judgments.

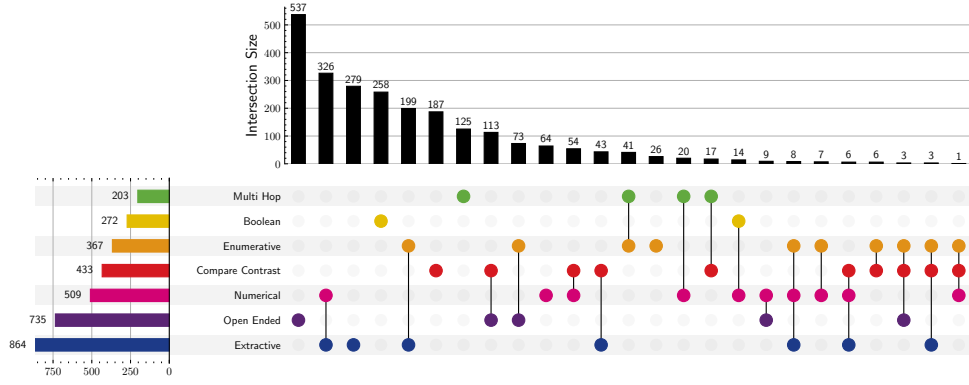


Figure D.2: UpSet plot of query-type intersections in ViDoRE V3. Horizontal bars show total counts for each query type, and top bars show frequencies of specific type combinations indicated by the dot matrix. *Extractive* is the most common type overall, while *Open Ended* is the largest unique category; frequent *Enumerative–Extractive* intersections highlight list-based factual retrieval patterns.

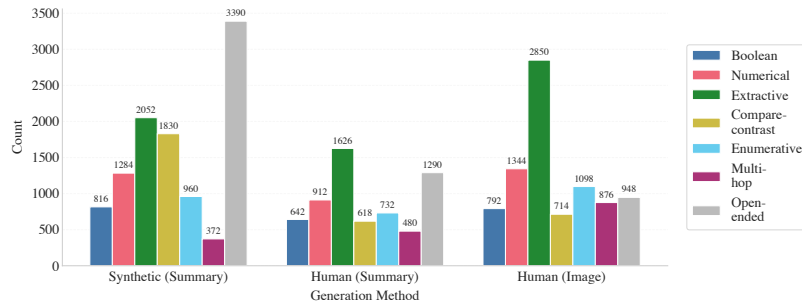


Figure D.3: Query type distribution by generation method. The figure compares how human and synthetic pipelines contribute to each query type.

Additional Retrieval Modality Performances To evaluate the hybrid retrieval setup, we use the multimodal Jina-v4 model to generate separate visual and textual rankings. We then construct a hybrid retrieval set by merging the top-5 results from each modality and removing duplicates. Because this set-union operation does not preserve a strict ranking order, we report the unranked F1 score. As shown in Table D.7, the hybrid approach consistently outperforms single-modality baselines.

Table D.2: Overview of ViDoRe V3 public corpora. For each corpus, we report domain coverage, content description, language, number of documents, pages, and queries. *Query counts exclude translations.

Corpus	Domain(s)	Description	Lang.	# Docs	# Pages	# Queries*	Main modalities
U.S. Public Company Annual Reports	Finance-EN	Consists of 6 10-K annual reports from major U.S. financial institutions for the fiscal year ended December 31, 2024.	en	6	2942	309	Text, Table
Computer Science Textbooks	Computer Science / Education	Consists of two open-source, peer-reviewed textbooks from OpenStax covering foundational topics in computer science, Python, and data science.	en	2	1360	215	Books
FDA Reports	Pharmaceuticals	Consists of FDA presentations and Springer books (2016–2023) covering regulatory policies, drug development, and public health initiatives.	en	52	2313	364	Slides, Books
HR Reports from EU	HR	Includes recent European Commission reports and papers on EU labour markets, social development, and employment policies.	en	14	1110	318	Reports
USAF Technical Orders	Industrial Maintenance	Comprises U.S. military technical orders and manuals for aircraft maintenance, safety procedures, and material handling, revised through 2025.	en	27	5244	283	Manuals
French Physics Lectures	Physics	A collection of educational materials offering an interdisciplinary exploration of modern physics and complexity science.	fr	42	1674	302	Slides
French Public Company Annual Reports	Finance-FR	Contains the 2023–2024 annual reports of major French luxury companies (Dior, Hermès, Kering, L’Oréal, LVMH).	fr	5	2384	320	Reports
French Governmental Energy Reports	Energy	Gathers official documents from French public agencies on energy, economic, and environmental issues in France.	fr	42	2229	308	Reports, Slides

Table D.4: Retriever reference table. Aliases used in Tables 4.1, D.5, and D.6 are mapped to HuggingFace model names and citations.

Model alias	Full model name	Reference
Qwen3-8B	Qwen3-Embedding-8B	[162]
Jina-v4	jina-embeddings-v4	[149]
Qwen3-0.6B	Qwen3-Embedding-0.6B	[162]
LFM2-350M	LFM2-ColBERT-350M	[347]
BGE-M3	BGE-M3	[128]
BM25S	BM25S	[348]
ColEmbed-3B-v2	llama-nemoretriever-colembed-3b-v2	[155]
ColNomic-7B	colnomic-embed-multimodal-7b	[349]
ColEmbed-3B	llama-nemoretriever-colembed-3b-v1	[155]
ColNomic-3B	colnomic-embed-multimodal-3b	[349]
ColEmbed-1B	llama-nemoretriever-colembed-1b-v1	[155]
ColQwen2.5	colqwen2.5-v0.2	[6]
Nomic-7B	nomic-embed-multimodal-7b	[349]
ColQwen2	colqwen2-v1.0	[6]
Nomic-3B	nomic-embed-multimodal-7b	[349]
ColPali	colpali-v1.3	[6]
Mxbai Edge 32M	mxbai-edge-colbert-v0-32m	[350]
GTE-ModernColBERT	GTE-ModernColBERT-v1	[233]
ColModernVBERT	colmodernvbert	[9]
ColSmol-256M	colsmol-256M	[182]

Table D.5: English-only retrieval performance (NDCG@10). ★ marks single-vector models. Results are computed on English queries over English datasets.

Model	Size (B)	C.S.	Nucl.	Fin.	Phar.	H.R.	Ind.	Tele.	Average
<i>Textual Retrievers</i>									
Jina-v4	3	67.3	48.2	56.5	59.0	58.8	45.8	61.0	56.7
Qwen3-8B★	8	73.5	42.2	54.8	62.4	52.3	45.3	66.0	56.6
LFM2-350M	0.35	70.6	45.4	48.3	62.1	53.2	47.9	63.8	55.9
Mxbai Edge 32M	0.03	68.0	44.4	48.2	62.5	52.7	47.1	61.9	55.0
BM25S	-	64.7	45.9	49.9	56.9	49.6	45.6	58.3	53.0
Qwen3-0.6B★	0.6	70.5	39.7	51.5	57.4	46.2	42.4	59.7	52.3
GTE-ModernColBERT	0.15	63.6	41.7	39.8	62.0	46.2	44.6	59.7	51.1
BGE-M3★	0.57	63.6	34.3	43.9	54.7	45.3	39.0	54.3	47.9
<i>Visual Retrievers</i>									
ColEmbed-3B-v2	3	78.6	52.9	69.1	67.6	65.4	56.8	71.7	66.0
ColEmbed-3B	3	77.8	53.4	69.5	66.9	64.9	57.0	69.4	65.6
ColEmbed-1B	1	75.5	52.2	67.0	66.2	64.5	56.1	68.7	64.3
Jina-v4	3	74.2	52.4	66.1	65.2	64.6	55.9	68.7	63.9
ColNomic-7B	7	78.2	48.2	63.1	64.6	62.9	54.2	69.6	63.0
ColNomic-3B	3	75.5	45.5	63.0	63.7	62.6	52.8	68.6	61.7
ColQwen2.5	3	75.2	42.9	61.2	60.9	59.2	49.4	65.3	59.2
Nomic-7B★	7	70.9	42.3	57.6	63.8	55.9	48.5	62.0	57.3
ColQwen2	2	73.5	44.1	50.9	58.1	54.7	49.8	63.2	56.3
ColPali	7	72.5	38.1	43.3	57.7	53.3	47.0	59.2	53.0
Nomic-3B★	3	62.1	37.2	53.3	59.2	51.9	41.1	57.2	51.7
ColModernVBERT	0.25	59.7	42.0	50.4	56.6	47.0	43.9	55.2	50.7
ColSmol-256M	0.25	57.4	36.5	47.7	51.4	46.0	38.5	47.5	46.4

Table D.6: French-only retrieval performance (NDCG@10). ★ marks single-vector models. Results are computed on French queries over French datasets.

Model	Size (B)	Phys.	Ener.	Fin.	Average
<i>Textual Retrievers</i>					
Jina-v4	3	44.0	63.4	44.8	50.7
Qwen3-8B★	8	45.8	60.2	37.6	47.9
Qwen3-0.6B★	0.6	43.8	54.9	28.5	42.4
BGE-M3★	0.57	38.3	53.1	28.4	39.9
BM25S	-	39.8	57.4	35.9	44.4
<i>Visual Retrievers</i>					
ColEmbed-3B-v2	3	48.2	67.5	48.2	54.6
ColNomic-7b	7	48.5	67.0	47.9	54.5
ColNomic-3b	3	48.5	67.9	46.8	54.4
Jina-v4	3	46.8	66.7	48.6	54.0
ColEmbed-3B	3	46.6	66.3	48.9	53.9
ColEmbed-1B	1	44.7	64.6	47.8	52.4
ColQwen2.5	3	47.8	62.3	43.6	51.2
Nomic-7B★	7	45.6	61.6	41.3	49.5
ColQwen2	3	43.9	55.6	26.5	42.0
Nomic-3B★	3	43.6	56.4	34.4	44.8
ColPali	7	43.2	50.5	23.6	39.1

Table D.7: Performance comparison of retrieval modalities (F1@10) on Jina-v4. The hybrid method merges top-5 visual and top-5 textual hits with deduplication. The final row reports the average number of unique pages in the hybrid set, which consistently outperforms either single modality.

Modality	English Datasets							French Datasets			Avg.
	C.S.	Nucl.	Fin.	Phar.	H.R.	Ind.	Tel.	Phys.	Ener.	Fin.	
Visual	39.4	25.5	28.4	27.5	30.0	21.4	31.4	26.6	25.2	22.9	27.8
Textual	35.4	23.1	24.5	24.2	27.4	16.5	29.0	25.8	23.9	20.4	25.0
Hybrid	43.0	27.7	30.9	29.7	32.6	22.2	35.5	26.5	29.8	24.3	30.2
<i>Avg. # Pages for hybrid</i>	6.96	7.38	7.77	7.40	7.29	7.77	7.09	7.26	6.97	7.61	7.35

D.6 ColEmbed-3B-v2 performance breakdown

Table D.8 details the retrieval scores of ColEmbed-3B-v2 by query language, highlighting small performance variations by language.

Performance by number of annotated pages As seen in Figure 4.7, performance drops with the number of annotated pages. However, a potential confounding factor is the correlation between query type and the number of annotated pages, since more complex query types also have higher number of annotated pages (Figure D.4). We perform a stratified regression analysis to isolate these two effects.

We model NDCG@10 as a linear function of the number of annotated pages (P) stratified by query type. For each of the 7 query types, we fit an ordinary least squares regression:

$$NDCG@10 = a \cdot P + b + \epsilon.$$

Results in Figure D.5 and Table D.9 reveal that all query types suffer a significant performance penalty as the number of annotated pages increases. Slope values are nearly uniform ($a \approx -0.024$), suggesting a similar drop in retrieval accuracy across most query types. The open-ended and enumerative types are the two exceptions: despite having the lowest NDCG@10 for low page counts, they also have the shallowest slope, which suggests that retrieval success on these queries is constrained by the model’s fundamental difficulty in synthesizing multiple relevant sources rather than the volume of relevant context.

Table D.8: ColEmbed-3B-v2 NDCG@10 by query language. Scores are broken down by the language of the user query.

Query Language	NDCG@10
English	60.8
French	59.8
Portuguese	59.6
Spanish	59.6
Italian	59.1
German	57.9

Table D.9: Linear regression analysis of NDCG@10 decay with number of annotated pages, by query type. Slope a captures sensitivity to retrieval context size, while intercept b reflects intrinsic difficulty at minimum context size.

Query Type	Slope a	Intercept b	R^2
Boolean	-0.0239	0.797	0.101
Numerical	-0.0255	0.742	0.059
Extractive	-0.0230	0.745	0.084
Compare-contrast	-0.0247	0.710	0.117
Enumerative	-0.0172	0.669	0.080
Multi-hop	-0.0237	0.680	0.114
Open-ended	-0.0129	0.577	0.057

Performance by content type NDCG@10 by content type in Table D.10 show that retrieval is more challenging for visual content, with Image performing 10pp below Text. However, content type and query type are correlated in our benchmark: for instance, tables appear in numerical queries $2.2 \times$ more often than the baseline, while images are over-represented in open-ended queries (Figure D.6). Since numerical queries are easier than open-ended ones, we test whether the effect of content



Figure D.4: Average number of annotated pages by query type. The plot shows how evidence spread varies with query type.

type is a byproduct of query type confounding. We fit an additive model that predicts performance as the sum of independent query-type and content-type effects. Figure D.7 shows the residuals which measure deviation from this baseline. We see that most residuals are below 5pp, indicating that the two factors combine additively without significant interaction.

Content type	NDCG@10	Content type count
Text	59.3	17244
Chart	56.3	2364
Infographic	55.2	2814
Table	53.9	6480
Other	50.8	492
Image	49.3	1140
Mixed	45.1	1164

Table D.10: ColEmbed-3B-v2 NDCG@10 by content type. Each annotated page is labeled by the type of query-relevant content inside human bounding boxes; pages may receive multiple labels. The Mixed category denotes annotations spanning multiple content types.

Figure D.5: ColEmbed-3B-v2 NDCG@10 by number of annotated pages and query type. Performance trends are shown jointly by query complexity and evidence spread.

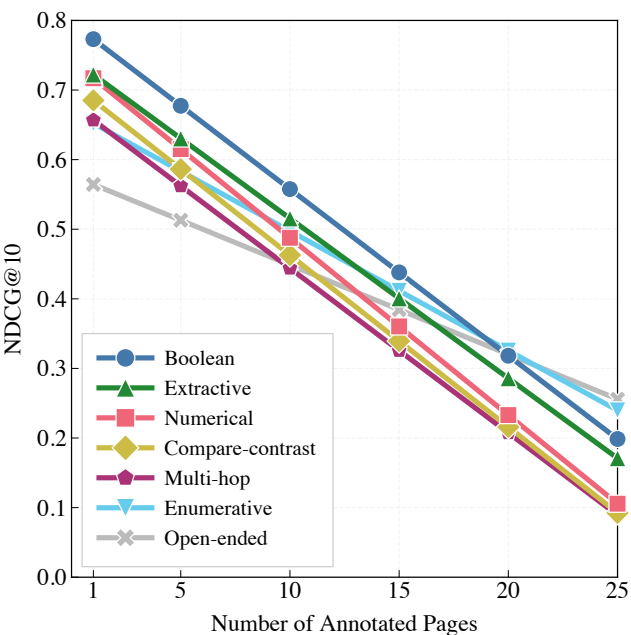
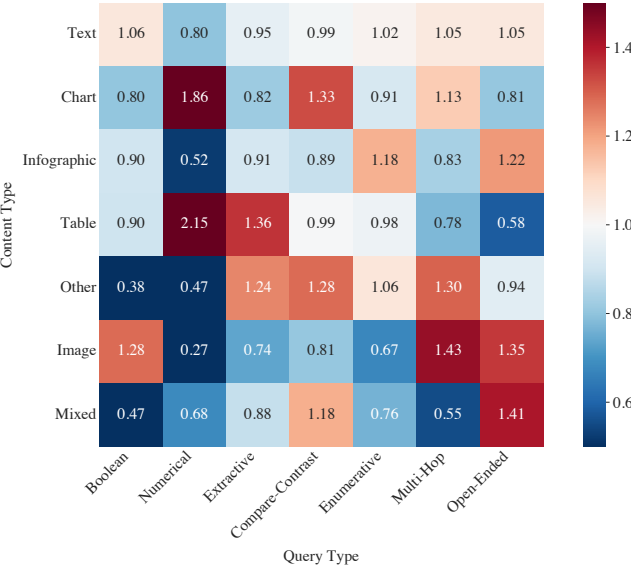


Figure D.6: Lift of query types by content type. Each cell reports observed frequency over baseline frequency for a content-type/query-type pair. Values above 1 indicate over-representation; values below 1 indicate under-representation.



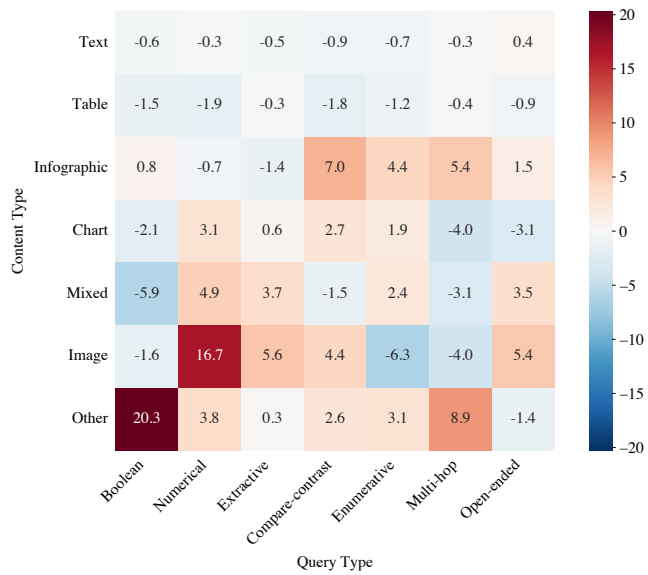


Figure D.7: Residuals from an additive performance model. Each cell is the difference between observed NDCG@10 and predictions from query-type and content-type main effects. Near-zero values indicate limited interaction; positive and negative values indicate better- or worse-than-expected combinations.

D.7 Bounding box annotations

Inter-annotator agreement Table D.11 shows IoU and F1 scores between human annotations, to detail results of Section 4.3.3.

Table D.11: Inter-annotator agreement metrics on bounding box annotations. Agreement statistics summarize consistency of human evidence localization.

Metric	English Datasets							French Datasets			Average
	C.S.	Nucl.	Fin.	Phar.	H.R.	Ind.	Tele.	Phys.	Ener.	Fin.	
IoU	0.500	0.476	0.462	0.615	0.474	0.502	0.526	0.443	0.470	0.503	0.497
F1	0.608	0.594	0.569	0.720	0.594	0.611	0.637	0.540	0.569	0.581	0.602

Bounding box predictions Figure D.20 shows the prompt used to generate final answers with inline bounding boxes for visual grounding, and Figure D.8 reports bounding box localization F1 scores by dataset.

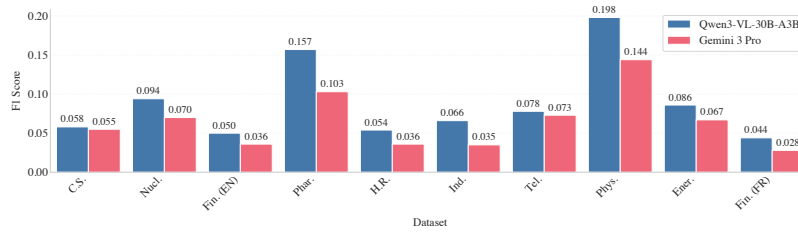


Figure D.8: Model bounding-box localization performance. Each F1 score measures zone-based overlap between model predictions and human annotations, using the annotator that yields the highest match.

D.8 Final answer evaluation

Evaluation setup Generated final answers are evaluated in a pass@1 setting using GPT 5.2 with medium reasoning effort as the LLM judge. The judge compares each generated answer against the ground-truth annotation and returns a binary correctness label. The answer generation and judge prompts are shown in Figure D.18 and Figure D.17 respectively. We evaluated Gemini 3 Pro with low thinking effort, GPT-5 with medium reasoning effort, as well as the thinking version of Qwen3-VL-235B-A22B.

To assess the reliability of our judge, we conducted 5 independent evaluation runs on a fixed set of Gemini 3 Pro outputs. Individual run scores showed minimal fluctuation (mean 72.09 %, $\sigma = 0.22$ %) and high internal consistency (Krippendorff's $\alpha = 0.91$), confirming that the judge is consistent given a fixed context.

End-to-End Pipeline Stability While the judge demonstrates high consistency on fixed inputs, the full evaluation pipeline introduces a second layer of variability: the model's generation process. To quantify the end-to-end variance under rigorous conditions, we performed 5 independent runs. For computational efficiency, we restricted this stress test to the most challenging corpus in each language: *Industrial Maintenance* (English) and *Finance* (French).

We measured an average score of 65.74 % with a standard deviation of 0.94 %. Crucially, the evaluation signal remains robust against generative noise, achieving a Krippendorff’s α of 0.80. This agreement confirms that the end-to-end results are statistically reliable even when subjected to the most difficult evaluation scenarios.

Easy/hard query filtering To classify queries by difficulty, we prompt a panel of 6 LLMs to answer each query without access to any corpus context. We select GPT-5-nano, GPT-5-mini, GPT-5, Qwen3-VL-30B-A3B, Gemini 2.5 Flash, and Gemini 2.5 Pro to span different model families and capability levels. Each model receives only the query text and is asked to provide a direct answer with the prompt in Figure D.16. Answers are evaluated for correctness using the same GPT-5.2 judge described above. A query is labeled *easy* if at least one model answers correctly, and *hard* otherwise. Table D.12 reports per-model accuracy and the resulting proportion of easy queries for each dataset. The distribution varies substantially across domains: knowledge-intensive datasets such as Computer Science and Physics have over 85% easy queries, while domain-specific datasets such as Finance and Energy contain fewer than 35% easy queries, reflecting the specialized nature of their content.

Table D.12: Percentage of queries correctly answered by LLMs without corpus context. A panel of six LLMs answers queries from the eight public datasets without retrieval context. Queries solved by at least one model are labeled *easy*; the rest are *hard*. Easy queries account for 48.6 % of the benchmark.

Model	English Datasets					French Datasets			Total
	C.S.	Fin.	Phar.	H.R.	Ind.	Phys.	Ener.	Fin.	
GPT-5-nano	74.4	7.4	30.5	12.9	15.6	74.2	14.0	9.1	29.8
GPT-5-mini	79.1	13.3	37.4	17.6	20.5	80.1	13.6	13.1	34.3
GPT-5	76.3	25.2	50.8	29.9	32.2	80.5	26.0	22.2	42.9
Qwen3-VL-30B-A3B	60.9	3.9	19.2	6.3	9.9	60.9	6.8	4.4	21.5
Gemini 2.5 Flash	66.1	8.7	30.8	13.5	15.9	63.6	14.9	13.1	28.3
Gemini 2.5 Pro	70.2	16.8	29.4	15.4	23.7	63.9	20.5	20.0	32.9
Easy queries (%)	86.5	31.7	57.1	36.5	38.9	86.4	32.5	30.0	48.6

D.9 Visual grounding examples

Qualitative analysis reveals distinct failure modes. Gemini frequently produces off-by-one page indexing errors: the predicted coordinates would correctly localize the target content if applied to an adjacent page. The two models also differ in box granularity: Gemini tends to draw tight boxes around individual elements (e.g., a single table cell or text line), whereas Qwen3-VL generates larger boxes encompassing entire sections or paragraphs, more closely matching human annotation patterns. Figures D.9 and D.10 illustrate these tendencies across four dataset pages: Qwen3-VL’s bounding boxes are comparatively wide and encompass entire page elements (pages (a), (c), and (d)), while Gemini 3 Pro’s visual grounding is more precise (pages (b) and (c)). This difference in granularity partially explains Qwen3-VL’s higher F1 scores, as broader boxes are more likely to overlap with the ground-truth zones used in our evaluation. Both models exhibit errors and omissions: in page (b), the

chart is not labeled by Qwen3-VL, and in page (d), Gemini 3 Pro predicts incorrect bounding boxes for the bottom table while Qwen3-VL provides grounding for the wrong table.

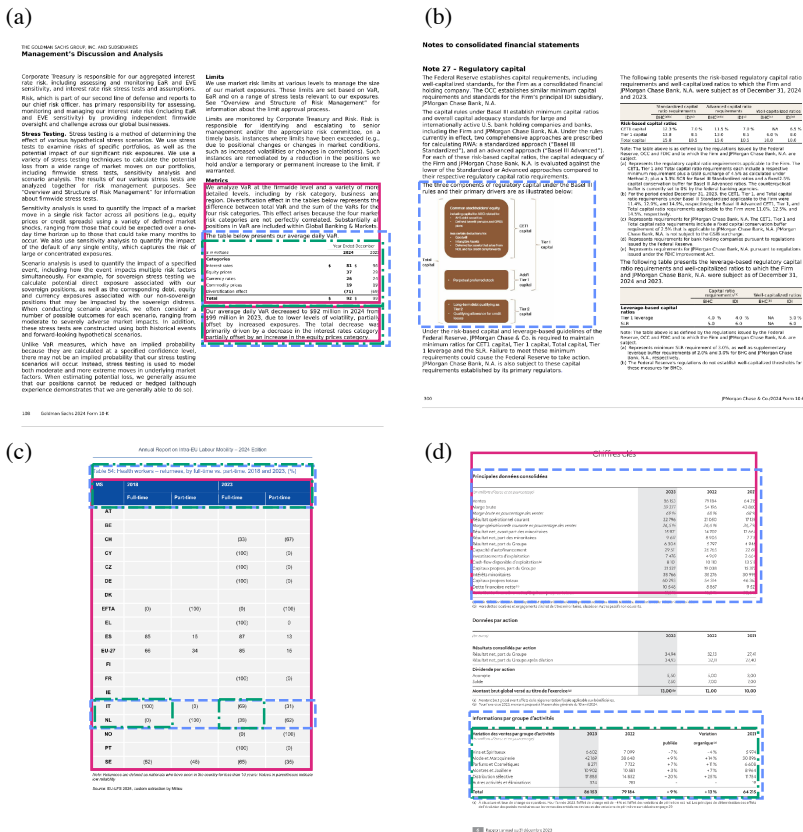


Figure D.9: Visual grounding comparative examples for Qwen3-VL-30B-A3B. Each panel shows a document page with Qwen3-VL’s predicted bounding boxes (solid magenta) and human bounding boxes (dashed blue and green, one color per annotator). Corresponding datasets and queries: (a) `finance_en`: What was the average daily Value at Risk (VaR) for Goldman Sachs during 2024?, (b) `finance_en`: List the 3 components of regulatory capital under Basel III, and determine the role of each component., (c) `hr_en`: Analyze how full-time employment among returning health workers evolved in the Netherlands and Italy from 2018 to 2023, and describe the differences in their employment trends., (d) `finance_fr`: Croissance Mode Maroquinerie vs Vins Spiritueux 2023 performance

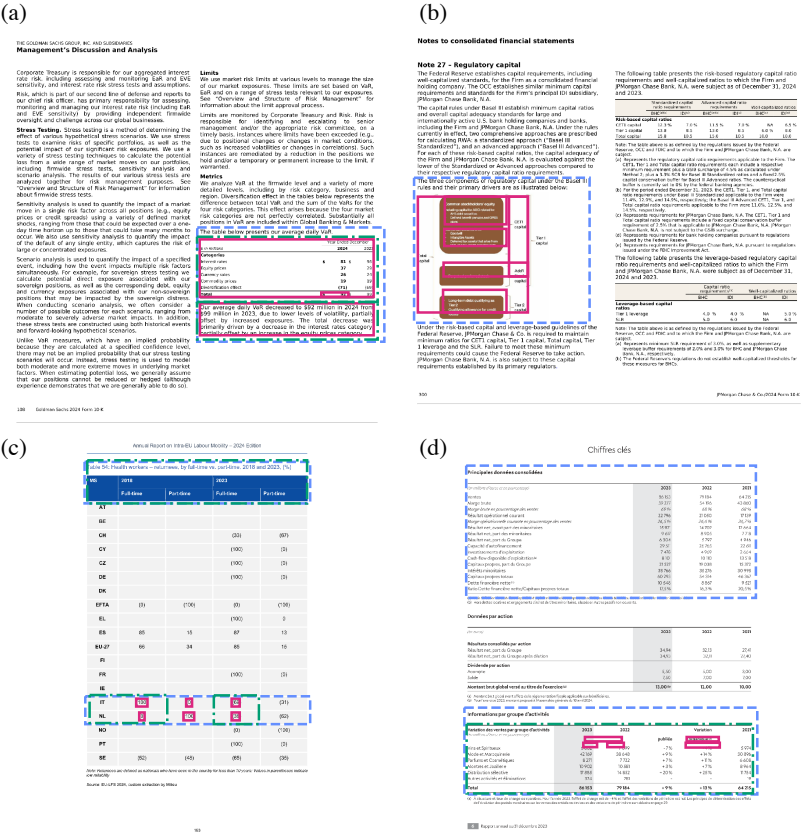


Figure D.10: Visual grounding comparative examples for Gemini 3 Pro. Each panel shows a document page with Gemini’s predicted bounding boxes (solid magenta) and human bounding boxes (dashed blue and green, one color per annotator). Corresponding datasets and queries: (a) `finance_en`: What was the average daily Value at Risk (VaR) for Goldman Sachs during 2024?, (b) `finance_en`: List the 3 components of regulatory capital under Basel III, and determine the role of each component., (c) `hr_en`: Analyze how full-time employment among returning health workers evolved in the Netherlands and Italy from 2018 to 2023, and describe the differences in their employment trends., (d) `finance_fr`: Croissance Mode Maroquinerie vs Vins Spiritueux 2023 performance

D.10 Instructions given to Annotators

Query Generation Figure D.11 details step-by-step instructions to annotators to generate queries from summaries and images.

Query-Page Relevancy linking Figure D.12 details the step-by-step instructions provided to annotators for assessing page relevance, identifying content modalities, and localizing evidence via bounding boxes. Table D.13 gives the definitions of relevancy scores used by the human annotators.


```

# Step 1
  The annotator will be provided with content(text summary or image(s))
  ↳ and a list of instructions
  on the queries that are expected.

# Step 2
  Read and analyse the content.

# Step 3
  - The annotator writes a series of queries that can, supposedly, be
  ↳ answered by the summarized
  content. It is okay if the information needed to answer the question is
  ↳ in other parts of the
  document or not explicitly written in the summary.
  - If the summary is not adapted to a specific type or format of
  ↳ questions, the annotator may skip.
  - They should follow the number of queries asked for each category.
  - They should follow the expected type of queries provided and the
  ↳ format.

```

Figure D.11: Instructions given to human annotators to create queries

Table D.13: Relevance definitions used for page-level annotations. The table summarizes the scoring rubric applied during human relevance labeling.

Score	Label	Definition
2	Fully Relevant	The page contains the complete answer.
1	Critically Relevant	The page contains facts or information required to answer the query, though additional information is required.
0	Not Relevant	Provides no information relevant to the query.

```

# Task overview
In this task, the annotator will be provided with a query and pages that are
↳ supposed to be relevant
to answer the query. The annotator's goal is to rate the relevance in
↳ answerability of each page with
respect to the query.

# Step 1:
- Review the query and pages to get an understanding of the content and
↳ domain

# Step 2:
Rate the query quality.
- If adheres to guidelines > Good (1)
- If doesn't adhere to guidelines > Poor(0)
- If the query is Poor quality, skip the task.

# Step 3:
- For each page, rate the relevance with respect to the query
- If page completely answer query > Fully Relevant(2)
- If page contains information required to answer the query > Critically
↳ Relevant(1)
- If page contains no relevant information > Not Relevant(0)

# Step 4:
- For each page, annotate the modalities in which the relevant
↳ information is located concerning
the query-page link, relevant information can be located in multiple
↳ modalities at the same time:
Modality : ["text", "table", "chart", infographic", "image", "other"]
- If relevance score = 0, modality may be "N/A"

# Step 5:
- For each page, draw bounding boxes around the relevant
↳ text/chart/image/infographic (if any)
- If relevance score = 0, do not draw a bounding box

# Step 6:
- Repeat steps 3, 4 and 5 for all pages

# Step 7:
- Propose an answer to the query, given the relevant pages.
- If the query is not answerable, rate it as "unanswerable"

```

Figure D.12: Instructions given to human annotators to annotate query-page relevancy

D.11 Prompts

All the prompts used for both dataset generation and evaluations are detailed from Figure D.13 to Figure D.20.

```

<mission>
You are an assistant specialized in visual document understanding tasks. You will be given a context,
summarizing the content of a section or multiple document sections. Your goal is to carefully analyze
the context and to solve a series of tasks related to its content. You are tasked with generating
query-answer pairs. Your queries will be used to simulate a user unfamiliar with the specific content
of the page, and who is looking for information in a large knowledge base through a search engine.
The user does not have access to the document and is looking for information that can be present in
any document in the knowledge base.
</mission>

<definitions>
- A query is said to be fully answerable if the page contains a precise and complete answer to the
query.
- A query is said to be partially answerable if the page contains relevant information that is
directly related to the query but some key information is missing and must be retrieved in other
pages or documents in order to give a precise and complete answer.
- An open-ended query is an explanatory or descriptive query that synthesizes information; may be
broad in scope and focused on qualitative aspects of the summary
- A compare-contrast query is a query that requires comparing and/or contrasting multiple entities or
topics that are closely related to each other
- An enumerative query is a query that asks to list all examples that possess a common property,
optionally requesting details about the specifics of each example.
- A numerical query is a query that asks for a specific number or calculated number given a summary.
The query should require more than simply reading numbers directly from the page.
- A boolean query is a yes/no query that may involve multiple steps of reasoning.
- An extractive query is a clear and specific query that can be answered using only a specific piece
of information.
- A multi-hop query is a complex query that requires retrieving and integrating information from
multiple sources or steps to produce a complete answer.
- A question query is a complete sentence that ends with a question mark, typically used to seek
specific information or clarification.
- A keyword query is a brief, often fragmented phrase or set of terms used to search or filter
information, without forming a full grammatical sentence.
- An instruction query is a directive that describes a task to be performed on the documents, often in
the form of a command or request.
</definitions>
<rules>
<queries>
- Generate queries only in {{ language }}.
- Make queries diverse, natural, and plausible for someone unfamiliar with the document.
- Each query must be standalone; do not reference "the page", "the table", "the figure",
"the document", "the text", "the table of contents", etc.
- Rephrase; avoid copying wording from the source so semantic matching, not surface matching,
is tested.
- You may include queries about relationships or trends often shown in
↳ images/vidorev3-tables/images/vidorev3-images/graphs,
but never refer to a specific table/figure.
- Avoid overly generic queries that apply to any document.
- Keep each query concise ({{ length }} words).
- When appropriate, write multi-hop queries that integrate information across the provided pages.
</queries>
</rules>
<instructions>

Used Documents: {{ document_names }}
<summary>
{{summary}}
</summary>
Using the provided context, generate a {{ difficulty }}, {{ reasoning_type }}, {{ answerability }}
query. The query should be {{ query_type }} using the provided context and have the format of a
{{ query_format }} query.
The query should be self-sufficient and related to the context.
</instructions>

```

Figure D.13: Prompt used generate synthetic queries with the NeMo Data Designer tool

```

<mission>
You are an assistant specialized in visual document understanding tasks. You
↳ will be given a document
page by page and a question. Your goal is to carefully analyze the page and
↳ say if it is related to
the question's answer. You are tasked with generating question-page
↳ affiliation as well as the
question answer if it exists in the page.
</mission>

<definitions>
- A question is said to be fully answerable if the corresponding page
↳ contains a precise and complete
answer to the question.
- A question is said to be partially answerable if the corresponding page
↳ content is necessary to
answer the question but some key information is missing.
- A question is said to be unanswerable if the corresponding page contains
↳ information related to the
question's topic or domain but upon closer inspection does not contain
↳ information that is useful to
answer the question. Or if the page has no link whatsoever with the query.
</definitions>

<rules>
<page_affiliation>
- Be sure to put the relevance (and only that) between the tags
↳ <relevance>...</relevance>.
The possible values are:
  <relevance>fully answerable</relevance>,
  <relevance>partially answerable</relevance>
  <relevance>unanswerable</relevance>.
- Be very careful when doing your page affiliation. Only say a page is
↳ relevant when it really is.
</page_affiliation>
<answers>
- You must generate the answer between the tags <answer>...</answer>.
↳ Between these tags, you should
only put the answer to the question.
- You must generate answers in the following language: {language}
- Your answers should be complete sentences.
- When the question is ambiguous, your answer should state that there is an
↳ ambiguity in the question.
- You should always generate the answer based on all the information
↳ available on the page, even if
the question was generated only on part of the page.
</answers>
</rules>

<instructions>
Return if the following question is "fully answerable" "partially
↳ answerable" or "unanswerable" based
on the content of the page between the tags <relevance> ... </relevance>.
If the question is answerable, provide the answer.

Here is the question :
{{ query }}

And there is the page content :

</instructions>

```

Figure D.14: Prompt used to pre-filter the irrelevant pages for a given query

You are given a set of document pages (images), a query, and a list of one
 ↳ or more
 proposed answers.

Query :

```
{{ query }}
```

Proposed Answers:

```
{{ answers }}
```

Your task is to carefully analyze the provided pages, the query, and the
 ↳ proposed answers. You must
 return a single, syntactically correct JSON object with the following
 ↳ structure:

```
```json
{
 "reasoning": "<string>",
 "information_in_pages": <true or false>,
 "answer_correctness": [<true or false>, ...],
 "reformulated_answer": "<string>"
}
```
```

Instructions for each field:

- reasoning: Explain the logic for each boolean in the `answer\_correctness`
 ↳ list.
- information_in_pages: Set to `true` if the information needed to
 ↳ definitively answer the query
 is present in the pages. Otherwise, set to `false`.
- answer_correctness: A list of booleans, corresponding to each proposed
 ↳ answer in the original
 order. Use `true` if the answer is verifiably correct based on the pages and
 ↳ `false` otherwise.
- reformulated_answer: A single string containing the most precise and
 ↳ correct answer to the query,
 derived only from information in the pages. If any of the proposed answers
 ↳ are correct, use them as
 a basis for synthesizing this improved answer. The reformulation must be
 ↳ concise and factual.

Important rules:

- Base your entire analysis strictly on the content of the provided document
 ↳ pages.
- Do not use outside knowledge.
- Do not invent, infer, or assume information that is not explicitly stated
 ↳ in the pages.
- Always provide a string for the `reformulated\_answer`, even if no correct
 ↳ answer can be formed
 from the text.
- Your final output must be only the JSON object.

Figure D.15: Prompt used to merge human annotators answers

Give a very precise and concise answer to the following query.
 If you are unable to answer, output 'I don't know'.
 Query: {{ query }}

Figure D.16: Easy/hard query filtering prompt

```

You are an expert judge evaluating the accuracy of a test answer against a
↳ gold-standard true answer.
Your goal is to determine if the test answer captures the essential "core
↳ information."

### Evaluation Criteria:
- Correct: The test answer contains all core information of the true answer.
↳ Minor omissions of
non-essential details or the addition of minor, non-contradictory
↳ information should still be marked
as "Correct."
- Partially Correct: The test answer captures some of the core information,
↳ but suffers from
significant omissions or includes substantial extra information that was not
↳ requested or present in
the true answer.
- Incorrect: The test answer is fundamentally wrong, contradicts the true
↳ answer, or misses the core
information entirely.

### Input Data:
Query: {{ query }}

True Answer: {{ true_answer }}

Test Answer: {{ test_answer }}

### Output Format:
Provide a very brief explanation for your judgment. You must output your
↳ final response in a JSON
format with two fields: "explanation" and "judgment" (which must be
↳ "Correct", "Partially Correct",
or "Incorrect").

```

Figure D.17: Judge prompt used for end to end evaluation

```

You are an expert at answering query based on documents.

Here is a list of relevant documents: {{ documents }}

Based on the above documents, answer the following query: {{ query }}

Keep the response short when appropriate. Output the answer only.

```

Figure D.18: Answer generation prompt used for end to end evaluation

```

English text to translate: {{ query }}

Translate the English text above to French.
Make sure you follow the format of the English text. Don't change acronyms.
Follow the following json schema.
{
  "french_translation": ...
}

```

Figure D.19: Query translation from English to French prompt

```

# Role and Objective
- Serve as an expert in document analysis and visual grounding.
- Given a query and multiple document page images, provide a natural
  ↳ language answer with inline
  ↳ grounding references.

# Instructions
- Analyze all provided pages to answer the query comprehensively.
- For each piece of information used in your answer, provide visual
  ↳ grounding by including bounding
  ↳ box coordinates of all the sections of the document that help answer the
  ↳ query.
- Use this format to include the list of all bounding boxes of image N:
<bboxes image="N">[[x_{min}, y_{min}, x_{max}, y_{max}], ...]</bboxes>
  ↳ image="N" specifies the 0-indexed page number (0=first page, 1=second
  ↳ page, etc.)
  ↳ Include bounding boxes inline in your answer, immediately after
  ↳ mentioning the relevant
  ↳ information
  ↳ A given page may contain multiple non-contiguous sections that help
  ↳ answer the query. In this
  ↳ case, you must output the list of the bounding boxes of all these
  ↳ sections.
  ↳ You must group all the bounding boxes of a given page into a single
    <bboxes image="N">...</bboxes> tag.

# Grounding Principles
- DO NOT output more than 5 bounding boxes per page.
- Adjacent logical units must be enclosed in a single, continuous bounding
  ↳ box.
- Return multiple bounding boxes only if information is clearly independent
  ↳ and separated by
  ↳ significant non-relevant content.

# Output Format
- Provide a natural language answer to the query.
- Embed grounding tags directly inline where relevant information is
  ↳ discussed.
- Example: "The valuation technique described on page 1
  <bboxes image="0">[[120, 450, 890, 670], [100, 800, 330, 960]]</bboxes>
  uses discounted cash flow analysis."

```

Figure D.20: Bounding box prediction prompt

ModernVBert Appendix

E

E.1 Training

E.1.1 Implementation and Resources

Table E.1: Training details of our final models at each training stage. GPU Hours are on 80GB H100 GPUs.

| Model | Batch Size | Learning Rate | Training Steps | Training GPU Hours |
|---|------------|---------------|----------------|--------------------|
| Modality Alignment | | | | |
| <i>ModernVBERT</i> -base (Table E.2) | 4096 | 1e-4 | 5500 | 1920h |
| Contrastive Learning | | | | |
| Generalist contrastive training (Table E.4) | 256 | 2e-4 | 3917 | 80h |
| Document Specialization | | | | |
| Document-focused contrastive training w/ hard negatives (Table E.4) | 64 | 2e-4 | 19602 | 160h |

We list hyperparameters and resource details in Table E.1 for the various training stages of our final models. We employ ZeRO stage 1 optimizer [351] for our modality alignment runs. All ablation models are contrastively trained with gradient checkpointing [352] to reduce memory usage. All training runs are performed with FlashAttention 2.0 [134]. For LoRA configurations, we consistently use a rank r of 32, `lora_alpha` of 32, and a dropout of 0.1. For the implementation, we start from `m4`¹ and `ColPali`² codebases for training, and use the `MTEB`³ repository for evaluation.⁴

[351]: Rajbhandari et al. (2020), *ZeRO: Memory Optimizations Toward Training Trillion Parameter Models*

[352]: Chen et al. (2016), *Training Deep Nets with Sublinear Memory Cost*

[134]: Dao (2023), *FlashAttention-2: Faster Attention with Better Parallelism and Work Partitioning*

1: SmolVLM trainer, <https://github.com/huggingface/smolLM>

2: <https://github.com/illuin-tec/colpali>

3: <https://github.com/embedding-s-benchmark/mteb>

4: <https://github.com/illuin-tec/modernvbert>

E.1.2 Similarity Functions

Single-Vector Similarity. For single-vector models, we apply mean pooling for MLM-aligned encoders and end-of-sequence (EOS) pooling for CLM-based models and compute the cosine similarity of a query q and a document d as

$$\Phi_{\text{CosSim}}(\mathbf{q}, \mathbf{d}) = \exp(\cos(\mathbf{E}_q, \mathbf{E}_d) / \tau) \quad (\text{E.1})$$

Multi-Vector Similarity. For multi-vector models, we adopt the standard late-interaction scoring function defined as:

$$\Phi_{\text{LI}}(q, d) = \sum_{i \in \llbracket 1, N_q \rrbracket} \max_{j \in \llbracket 1, N_d \rrbracket} \langle \mathbf{E}_q^{(i)}, \mathbf{E}_d^{(j)} \rangle, \quad (\text{E.2})$$

where $\mathbf{E}_q^{(i)}$ and $\mathbf{E}_d^{(j)}$ denote token-level embeddings for the query and document, respectively.

E.1.3 Data

Modality Alignment Mixture

For our modality alignment trainings, we rely on The Cauldron dataset [111] and its Docmatix extension [138]. Table E.2 provides further details on the constitution of this dataset.

[111]: Laurençon et al. (2024), *What matters when building vision-language models?*

[138]: Laurençon et al. (2024), *Building and better understanding vision-language models: insights and future directions.*

Table E.2: Aggregated statistics of modality alignment datasets from The Cauldron 2 [111] and Docmatix [138], showing image counts, QA pairs, token counts, and the proportional contribution of each subsection to the overall mixture.

| Dataset Subsection | # Images | # QA Pairs | # Tokens | % Mix |
|-----------------------------|----------------|-----------------|-------------------|---------------|
| Captioning | 609,843 | 612,768 | 62,906,011 | 3.13 |
| Real-world VQA | 457,360 | 2,125,615 | 23,318,335 | 1.16 |
| OCR, Document Understanding | 2,499,258 | 11,415,478 | 426,806,479 | 21.21 |
| Chart/Figure Understanding | 539,743 | 24,444,120 | 30,315,784 | 1.51 |
| Table Understanding | 163,568 | 229,077 | 21,371,931 | 1.06 |
| Reasoning, Logic, Maths | 490,870 | 2,212,629 | 32,450,213 | 1.61 |
| Screenshot to Code | 547,974 | 548,296 | 336,299,551 | 16.71 |
| Text-only Instructions | 0 | 21,482,682 | 1,079,001,075 | 53.61 |
| Total | 5308616 | 63070665 | 2012469379 | 100.00 |

NatCap

To enrich our contrastive learning data mixture, we construct *NatCap* (Natural Captions), a large-scale dataset containing around 333000 contextualized image–caption pairs. This dataset is created by generating synthetic captions, along with cross-class and in-class discriminative tags, from existing image classification datasets (see Table E.3). For this purpose, we leverage Gemini-flash-2.5⁵ which produces captions conditioned on both the image content and the accompanying dataset metadata, as illustrated in Figure E.1. We detail the prompt below.

5: <https://ai.google.dev/gemini-api/docs/models?hl=fr#gemini-2.5-flash>

Table E.3: NatCap Dataset Composition. *NatCap* spans 13 different sources covering various images types. The total dataset is composed of 333k pairs

| Dataset | Description | # Items |
|--------------|--|---------------|
| Caltech101 | General objects. | 3000 |
| Caltech256 | General objects. | 30 000 |
| Cars | Car model classification. | 8000 |
| Country211 | Country where the picture is taken. | 28 000 |
| DTD | Describable textures (texture attributes). | 4000 |
| EuroSat | Land use / area zone type. | 16 000 |
| FER2013 | Facial emotion recognition. | 28 000 |
| FGCVAircraft | Aircraft model recognition. | 3000 |
| Food101 | Food categories. | 75 000 |
| OxfordPets | Dog/cat species. | 3000 |
| RESISC45 | Aerial scene / area zone type. | 18 000 |
| SUN397 | General scenes. | 109 000 |
| VOC2007 | General objects. | 8000 |
| TOTAL | | 333000 |

Contrastive Training Mix

In this subsection, we describe the composition of our data mixes used in the contrastive training stages. Table E.4 outlines the datasets included in each mix, including the Document-Focused variant employed for *ColModernVBERT*.



Class label: "Aston Martin V8 Vantage Coupe 2012"

NatCap caption: "A white 2012 Aston Martin V8 Vantage Coupe is showcased against a blurred background featuring a large car wheel."

Figure E.1: Example from the NatCap dataset

| Source | Description | Pairs | Epochs |
|----------------------------------|--|--------------|--------|
| Generalist Mix | | | |
| ColPali [6] | Query–Document images for visual retrieval | 118k | 1 |
| MSCOCO [117] | Natural images with human-written captions | 118k | 1 |
| <i>NatCap (ours, subsampled)</i> | Diverse images with synthetic captions | 118k | 1 |
| RLHN [353] | Text–text pairs for complex retrieval | 680k | 1 |
| TOTAL | | 1030k | |
| Document-Focused Mix | | | |
| ColPali [6] | Query–Document images for visual retrieval | 118k | 3 |
| RLHN [353] | Text–text pairs for complex retrieval | 300k | 3 |
| TOTAL | | 1254k | |

Table E.4: Data mixes for contrastive trainings. The *Generalist Mix* spans over 1M diverse pairs, while the *Document-Focused Mix* emphasizes document retrieval with extra ColPali epochs.

E.2 Baselines Details

In this section, we describe the models evaluated in as comparison to our document retriever model.

MoCa-3B [175]. A modality-aware continual pretraining model that transforms a causal vision-language model into a bidirectional multimodal embedding model, using interleaved image-text reconstruction and contrastive alignment to support cross-modal retrieval.

GME-Qwen2 [174]. A unified multimodal embedder built on Qwen2-VL [137], which produces shared embedding representations across text, image, and fused input modalities, enabling universal multimodal retrieval.

VLM2Vec [140]. A method that trains a vision-language encoder by converting a VLM through extensive contrastive post-training. Flagship model is based on the model Phi-3.5 [311].

E5-V [200]. An adaptation of the E5 embedding approach to multimodal models: it trains only on text pairs yet bridges the modality gap to handle image inputs, reducing cost while achieving universal embeddings.

ColPali [6]. A vision-based document retrieval model that processes document pages as images (no OCR) and produces multi-vector embeddings via a late-interaction mechanism over PaliGemma [112], enabling efficient and accurate retrieval.

ColQwen2.5 [6]. An extension of ColPali [6] using Qwen2-VL [137] as the backbone, carrying forward the late interaction retrieval paradigm over page image embeddings, capturing layout and textual context without OCR.

Jina-v4 [149]. A multimodal embedding model combining visual and textual inputs with support for multi-vector (late interaction) embeddings,

[175]: Chen et al. (2025), *MoCa: Modality-aware Continual Pre-training Makes Better Bidirectional Multimodal Embeddings*

[174]: Zhang et al. (2025), *GME: Improving Universal Multimodal Retrieval by Multimodal LLMs*

[137]: Wang et al. (2024), *Qwen2-VL: Enhancing Vision-Language Model’s Perception of the World at Any Resolution*

[140]: Jiang et al. (2024), *VLM2Vec: Training Vision-Language Models for Massive Multimodal Embedding Tasks*

[311]: Abdin et al. (2024), *Phi-3 Technical Report: A Highly Capable Language Model Locally on Your Phone*

[200]: Jiang et al. (2024), *E5-V: Universal Embeddings with Multimodal Large Language Models*

[6]: Faysse et al. (2025), *ColPali: Efficient Document Retrieval with Vision Language Models*

[112]: Beyer et al. (2024), *PaliGemma: A versatile 3B VLM for transfer*

[6]: Faysse et al. (2025), *ColPali: Efficient Document Retrieval with Vision Language Models*

[6]: Faysse et al. (2025), *ColPali: Efficient Document Retrieval with Vision Language Models*

[137]: Wang et al. (2024), *Qwen2-VL: Enhancing Vision-Language Model’s Perception of the World at Any Resolution*

[149]: Günther et al. (2025), *jina-embeddings-v4: Universal Embeddings for Multimodal Multilingual Retrieval*

[155]: Xu et al. (2025), *Llama Nemoretriever ColEmbed: Top-Performing Text-Image Retrieval Model*

[129]: Koukounas et al. (2024), *Jina CLIP: Your CLIP Model Is Also Your Text Retriever*

[202]: Zhou et al. (2024), *VISTA: Visualized Text Embedding For Universal Multi-Modal Retrieval*

[128]: Chen et al. (2024), *BGE M3-Embedding: Multi-Lingual, Multi-Functionality, Multi-Granularity Text Embeddings Through Self-Knowledge Distillation*

[179]: Tschannen et al. (2025), *SigLIP 2: Multilingual Vision-Language Encoders with Improved Semantic Understanding, Localization, and Dense Features*

[203]: Masry et al. (2024), *ColFlor: Towards BERT-Size Vision-Language Document Retrieval Models*

using adapters over a unified backbone to excel on visually rich document retrieval.

NemoRetriever [155]. An LI retriever that combines vision-language embeddings with a ColEmbed design, enabling high performance on visual document retrieval with structured patch matching and efficient similarity.

Jina CLIP [129]. A smaller scale vision-language model using CLIP embeddings, applied to document retrieval tasks; although not LI, it offers a lightweight multimodal baseline.

BGE Visualized M3 [202]. A vision-enhanced version of BGE M3 [128] that supports visual inputs and extends embedding models into multimodal domains.

SigLIP2-L-512/16 [179]. A multilingual vision-language bi-encoder model, which combines image and text modalities to yield unified embeddings across languages. This configuration handles images of 512x512 pixels and create subpatches of 16x16 pixels.

ColFlor [203]. A lightweight OCR-free visual document retriever with only 174M parameters built over Florence-2 and DaViT, delivering strong performance near ColPali with much lower computational cost and much faster encoding.

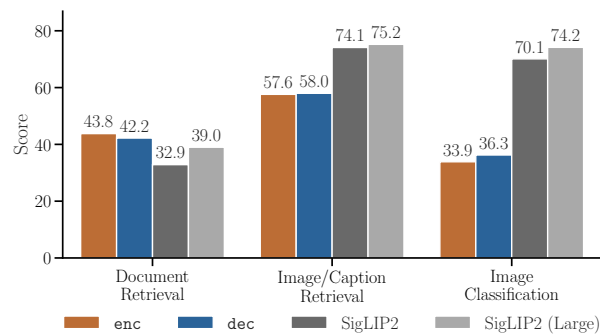
E.3 Additional Ablations

E.3.1 Performance Against Off-the-Shelf Dual Encoder

We study whether the *off-the-shelf* performance of the standalone vision tower outweighs the burden of adding language parameters and re-training through language modeling, as proposed in our work. Figure E.2 shows the results of the various models on the tasks described in Section 5.2. Similarly to Section 5.3.1, we observe that the early fusion model trained with an LM objective significantly outperforms the standalone vision tower on document retrieval tasks (+10.9 nDCG@5). It even surpasses the larger dual encoder (+4.8 nDCG@5) on these latter tasks. We note that the standalone vision tower largely outperforms the early fusion models on the other natural-image tasks, supporting the use of the SigLIP model for these tasks as found in various general benchmarks [193].

[193]: Xiao et al. (2025), *MIEB: Massive Image Embedding Benchmark*

Figure E.2: Impact of Modality Alignment objective on downstream tasks. Early Fusion of vision and text models boosts document retrieval tasks regardless of the LM objective, but degrades natural image and classification tasks w.r.t. the standalone *off-the-shelf* vision model SigLIP. Reported scores are aggregated MIEB scores (nDCG, Accuracy.)



E.3.2 Scaling Dynamics of Attention Masks

We study the different training dynamics of the different training objectives. We compare the enc (MLM) approach with a traditional dec (CLM) objective. Figure E.3 presents the performance of the two training objectives across a diverse set of tasks. While starting dec offers an advantage in low-data regimes, enc seems to catch up. In document retrieval tasks, it eventually surpasses dec and scales better.

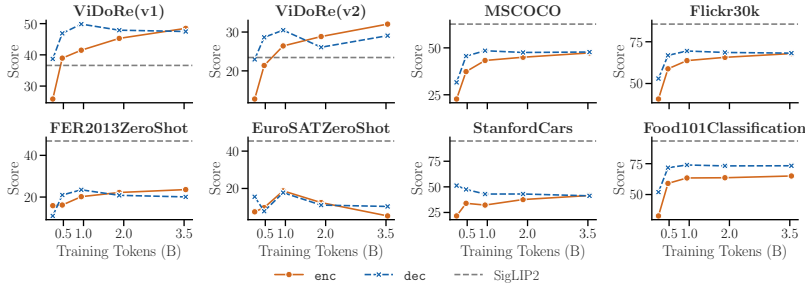


Figure E.3: Attention masks impact on modality alignment phase scaling. The dashed line marks the vision tower baseline. The orange curve shows the model initialized from a decoder LM with a CLM objective, and the blue curve shows the model trained with an MLM objective from an encoder LM. CLM performs better in low-data regimes, but MLM scales more effectively, surpassing CLM in document retrieval, while captioning and classification remain below the CLIP baseline.

E.3.3 Bridging the Gap with Longer Contrastive Training

We study the impact of additional in-distribution training pairs on embedding tasks by scaling the contrastive training stage. Starting from the final checkpoint of our encoder-based ablation model, we double the contrastive dataset size at each step and train until convergence⁶. This setup tests whether scaling continues to improve performance. Figure E.4 shows the scaling behavior. Performance improves overall with more in-distribution data. The vision-tower baseline is quickly surpassed on visual document benchmarks, and scaling narrows the gap on other tasks⁷. We note a plateau in captioning and classification, pointing to the need for more diverse data.

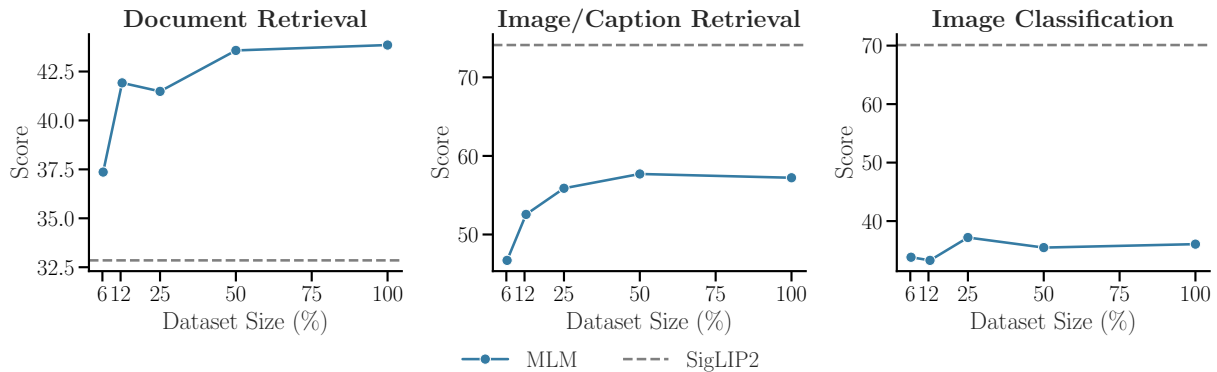


Figure E.4: Contrastive training scaling. Each dot on the blue curve represents one fraction of the baseline contrastive training mix (ColPali + MSCOCO). Performance improves with more in-distribution data, surpassing the baseline on document benchmarks and narrowing the gap on image captioning. There is no clear improvement in image classification, highlighting the need for more diverse pairs.

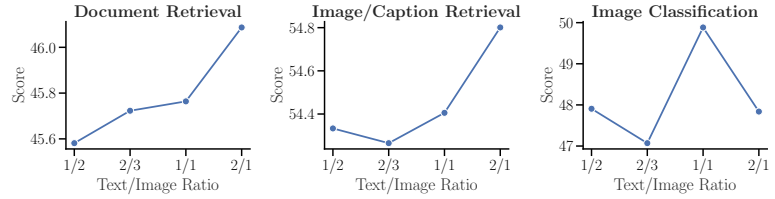
6: To avoid overfitting, we set an early stopping on an eval set. We limit the number of step to one epoch on the full dataset.

7: Note that the models probably won't fully recover baseline vision-tower performance. This highlights the need to choose models according to use case (e.g., lightweight CLIP-like models for image classification).

Optimal Text-To-Image Ratio for Document Retrieval

Our findings in subsection 5.3.2 indicate that incorporating additional text-only pairs boosts document retrieval performance. While our initial experiment employed a 1:1 text-to-image ratio, we further investigate how varying this ratio impacts our broad set of tasks. We start from the best contrastive mix in Table 5.1, and vary the text-to-image ratio. As shown in Figure E.5, increasing the number of text-only pairs *for a fixed amount of image pairs* consistently enhances retrieval performance. However, for natural image classification tasks, adding more text does not appear to provide benefits.

Figure E.5: Optimal text-to-image ratio in contrastive training mix. Increasing the ratio in retrieval tasks consistently improves the performances.



E.3.4 Late Interaction for Non-Documental Retrieval

Table E.5: Generalist retrieval performances. Late interaction benefits extend to non-documental retrieval tasks. Our multi-vector model increases its single-vector counterpart across all tasks, surpassing larger VLM-based retrievers.

| | Document Retrieval | | | Image/Caption Retrieval | | Average |
|----------------------------------|--------------------|------------|------------|-------------------------|-----------------|---------|
| | Model Size | VtDoRe(v1) | VtDoRe(v2) | MSCOCO (T→I) | Flickr30k (T→I) | |
| CLIP Encoders | | | | | | |
| siglip2-base-patch16-512 | 376M | 36.6 | 23.4 | 66.2 | 86.9 | 53.3 |
| siglip2-large-patch16-512 | 882M | 43.8 | 27.0 | 67.1 | 88.9 | 56.7 |
| clip-vit-base-patch16 | 151M | 25.5 | 20.4 | 50.3 | 76.8 | 43.3 |
| clip-vit-large-patch14 | 428M | 38.0 | 28.6 | 52.7 | 79.3 | 49.6 |
| VLM-based Encoders | | | | | | |
| VLM2Vec-Full | 4150M | 49.8 | 36.5 | 59.5 | 81.8 | 56.9 |
| e5-v | 8360M | 62.7 | 49.4 | 68.1 | 89.8 | 67.5 |
| Early Fusion Encoders | | | | | | |
| bge-visualized-base | 196M | 10.3 | 9.0 | 50.0 | 74.1 | 35.9 |
| bge-visualized-m3 | 873M | 12.4 | 10.2 | 39.6 | 69.0 | 32.8 |
| ModernVBERT-embed | 252M | 58.4 | 36.9 | 56.5 | 76.0 | 56.9 |
| ModernVBERT-embed (multi-vector) | 252M | 76.5 | 53.9 | 61.8 | 81.4 | 68.4 |

We want to study if the multi-vector gains transfer to non-documental retrieval. To do so, we contrastively post-train our base model on our generalist post-training mix presented in Table E.4. The late interaction generalist exhibits superior performance in retrieval setting, improving its single-vector performance by +20.2% (11.5 points), matching the performance of substantially larger VLM-based retrievers like E5-V (8.3B parameters, 67.5 points) and surpassing dual encoders like SigLIP (882M parameters, 56.7 points). This matches the capabilities observed in Section 5.3.1 for documental settings for models with native bidirectional attention, extending it to natural image tasks. This result extends the prevailing understanding from the document retrieval community, where the superiority of late-interaction is well-documented ([78], [233], [6]).

| | t | V ₁ DoR _E (v1) | V ₁ DoR _E (v2) | Average |
|--|---|--------------------------------------|--------------------------------------|-------------|
| <i>Document retrieval contrastive training starting checkpoint</i> | | | | |
| <i>ModernVBERT-base</i> | | 81.2 | 56.0 | 68.6 |
| + multi-vector generalist CL | | 80.7 | 55.4 | 68.1 |
| + single-vector generalist CL | | 80.6 | 54.0 | 67.3 |

While this performance gap is widely accepted for document retrieval, its applicability to caption matching tasks has not really been addressed. Our findings provide strong evidence that the fine-grained matching capabilities of late-interaction models are a key driver of performance in this domain too.

Model Merging

Our contrastive learning stage provides direct performance trade-offs on different tasks. Following recent trends, we evaluate how model merging techniques help mitigate performance degradation on specific tasks, while maintaining the performance enabled by the contrastive training [162, 237, 354, 355]. We merge our ablation model after modality alignment with the checkpoint after the full contrastive learning with two methods: SLERP [356] and average merging [357]. For SLERP, we compare three values for the λ coefficient (0.25, 0.5, 0.75). Figure E.6 displays the trends with the best method (SLERP, $\lambda = 0.75$). As we can see, the merged model mitigates the performance drop in Image/Caption Retrieval tasks, while maintaining significant gains on Image Classification tasks. However, merging strongly degrades performance on Document Retrieval, showing that benefits of merging embedding models are task-dependent.

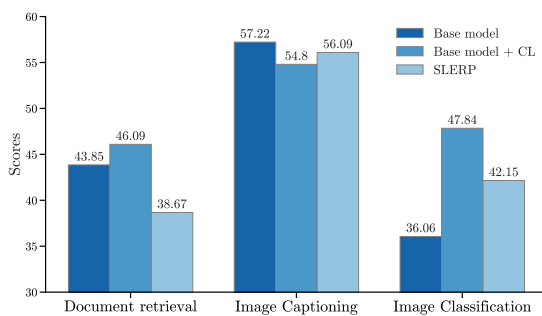


Table E.6: Performance of ModernVBERT Document Specialisation Curriculums. This table presents the performance of various contrastive training curriculums starting from *ModernVBERT-base*, on the V₁DoR_E(v1) and V₁DoR_E(v2) benchmarks. The generalist contrastive learning mix used in the last two models is detailed in Table E.4. We see that a preliminary stage of generalist contrastive learning harms the final document retrieval performance, regardless of whether a multi-vector approach is used.

- [162]: Zhang et al. (2025), *Qwen3 Embedding: Advancing Text Embedding and Reranking Through Foundation Models*
 [237]: Li et al. (2024), *Improving General Text Embedding Model: Tackling Task Conflict and Data Imbalance through Model Merging*
 [354]: Sung et al. (2023), *An Empirical Study of Multimodal Model Merging*
 [355]: Dziadzio et al. (2024), *How to Merge Your Multimodal Models Over Time?*
 [356]: Ilharco et al. (2022), *Patching open-vocabulary models by interpolating weights*
 [357]: Shoemake (1985), *Animating rotation with quaternion curves*

Figure E.6: Merging model results across tasks. Benefits are task-dependent, with performance degradation w.r.t. both original models in Document Retrieval.

Curriculum For Document Retriever Contrastive Post-Training

We conduct an ablation study to determine the optimal contrastive training curriculum for specializing *ModernVBERT* in document retrieval. Specifically, we investigate whether a preliminary generalist contrastive

training phase, intended to leverage a larger dataset, improves downstream performance. As shown in Table E.6, our results demonstrate that this initial generalist phase is detrimental to final performance (-0.5%). The optimal strategy is to specialize the model on the target task directly after its initial Masked Language Modeling (MLM) alignment.

E.3.5 Text-Only Retrieval

Table E.7: Average NDCG@5 of *ColModernVBERT* on NanoBEIR, a text retrieval benchmark with multiple sub domains. *For multimodal models, we only consider parameters of the text encoder

| Model | Params (M) | NDCG@5 |
|------------------------------|------------|--------|
| Statistical | | |
| BM25s | — | 0.559 |
| Single Vector | | |
| Jina Embeddings v4 | 3577* | 0.623 |
| E5-large-v2 | 335 | 0.605 |
| bge-m3 (Bi Encoder) | 567 | 0.590 |
| Qwen3-Embedding-0.6B | 600 | 0.567 |
| Multi Vector | | |
| LightOn GTE-ModernColBERT v1 | 149 | 0.669 |
| Jina ColBERT v2 | 137 | 0.642 |
| bge-m3 (Late Interaction) | 567 | 0.606 |
| ColBERT v2 | 110 | 0.593 |
| Colqwen2-v1.0 | 1580* | 0.593 |
| <i>ColModernVBERT</i> | 150* | 0.589 |
| Colqwen2.5-v0.2 | 3145* | 0.589 |

The results in Table E.7 detail the performance of *ColModernVBERT* and other baselines on the NanoBEIR text retrieval benchmark. It achieves an average NDCG@5 score competitive with single and multi vector models specialized for text, even without explicit optimization for this modality. This performance is encouraging and indicates a promising direction for training a unified model for both text and image retrieval.

E.3.6 Model Latency

Image Resolution Tradeoffs

Figure E.7 presents the pixel shuffling trade-off. Processing larger images creates more visual tokens, leading to very long sequences (around 17'500 tokens for a 2048x2048 px image with no pixel shuffling). Pixel shuffling allow to compress these sequence by concatenating the embeddings of spatially close patches. This diminishes the number of tokens for longer visual token embeddings. Table E.8 presents the latency to process one image of various resolutions on one L4 GPU and CPU.

Table E.8: *ModernVBERT* image processing latency. Computing the average time to process a single image on GPU and CPU. The average is computed on 100 images. The values represent the mean latency in milliseconds, with the standard deviation included in parenthesis.

| | Num. Visual Tokens | CPU Latency (ms) | GPU Latency (ms) |
|--------|--------------------|----------------------|--------------------|
| 512px | 128 | 287.2(± 7.8) | 43.6(± 1.4) |
| 1024px | 320 | 1015.8(± 58.1) | 150.3(± 2.5) |
| 2048px | 1088 | 2572.0(± 63.9) | 363.4(± 4.6) |

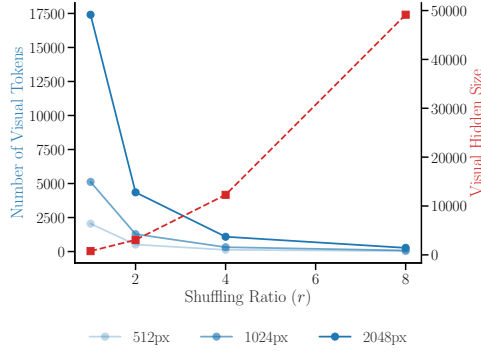


Figure E.7: Image processing parameters impact on visual tokens. Here we assume a square image for simplicity. Scaling the image size introduces naturally more tokens, but having a large enough pixel shuffling ratio ($r \geq 4$) allows to counterbalance by concatenating spatially close patch representations.

Online Query Encoding Latency

We evaluate the query embedding speed of our model on GPU. We use a single Nvidia H100 with 80GB of VRAM. As for Section 5.4.2, latencies are computed in batch size 1 to simulate online situations, and are averaged over all NanoBEIR queries. Only the text parameters are loaded and run, to minimize memory usage. Parameters are cast to bfloat16 and Flash Attention 2 is used. The resulting speeds are often much faster than those obtained by running inference through each model’s reference implementation. Results are shown in Table E.9). Interestingly in this setup where memory is not a bottleneck, model depth seems to be a large performance driver, sometimes more the parameter count. We finally evaluate batched GPU throughput. We use batches of size 512 by default and iteratively half it when memory is insufficient. We observe that *ModernVBERT* based models are extremely fast and can process 5000 queries per second. In the table, the reported figures correspond to the inverted throughput (latency per batch divided by the number of queries per batch). These speed and throughput gains are made possible due to a combination of size, and efficient hardware-informed design as well as the support of flash attention and sequence packing other models of the size often lack [201].

[201]: Warner et al. (2024), *Smarter, Better, Faster, Longer: A Modern Bidirectional Encoder for Fast, Memory Efficient, and Long Context Finetuning and Inference*

Table E.9: Text query encoding latency. The latency is computed both on high-end CPUs (1TB RAM, 128 cores) and GPU (Nvidia H100, 80GB) (mean $\pm$ std). Since only 649 queries are used, standard deviations are not reported in GPU batching mode (batches of 512 queries by default), for which we report the inverse throughput (average latency per batch divided by the batch size).

| | | Model Size (B) | CPU Latency (ms) | GPU Latency (ms) | GPU Batching (ms) |
|-----------------------|------------------|----------------|---------------------------------------|------------------------------------|-------------------|
| | Late Interaction | | | | |
| $\geq 1B$ Parameters | | | | | |
| MoCa-3B | | 3.75 | 158 _(± 147) | 26 _(± 3) | 4.54 |
| VLM2Vec | | 4.15 | 211 _(± 253) | 21 _(± 3) | 2.82 |
| GME-Qwen2-7B | | 8.29 | 412 _(± 411) | 25 _(± 1) | 9.07 |
| E5-V | | 8.36 | 434 _(± 379) | 22 _(± 2) | 9.55 |
| ColPali | ✓ | 2.92 | 175 _(± 113) | 14 _(± 1) | 3.07 |
| ColQwen2.5 | ✓ | 3.75 | 158 _(± 147) | 26 _(± 2) | 26 |
| Jina-v4 | ✓ | 3.75 | 158 _(± 147) | 26 _(± 2) | 4.54 |
| NemoRetriever-3B | ✓ | 4.40 | 155 _(± 118) | 20 _(± 2) | 4.59 |
| $\leq 1B$ Parameters | | | | | |
| Jina CLIP | | .22 | 14 _(± 7) | 6 _(± 2) | .69 |
| BGE Visualized M3 | | .87 | 38 _(± 42) | 10 _(± 2) | .77 |
| SigLIP2-L-512/16 | | .88 | 25 _(± 8) | 6 _(± 1) | .10 |
| ColFlor | ✓ | .17 | 17 _(± 9) | 8 _(± 2) | .31 |
| BiModernVBERT (ours) | | .25 | 20 _(± 11) | 14 _(± 2) | .20 |
| ColModernVBERT (ours) | ✓ | .25 | 20 _(± 11) | 14 _(± 2) | .20 |

NatCap Annotation Prompt

You are an image annotator expert.

You will receive an image along with its classification label and the classification task scope, and your task is to provide contextualized metadata about it.

The output should be a JSON object with the following metadata fields:

- ▶ **caption:** A descriptive caption of the image accounting for its label. This should be a **unique** and concise sentence that describes the image in detail.
- ▶ **class_tags:** A list of tags that represents the image and can help identify the class. (e.g., for a car image with its model as a class, this could be some specific attribute of the car)
- ▶ **other_tags:** A list of tags that represents the image but can help identify the image among others of the same class. (e.g., for a car image with its model as a class, this could be its color or the background of the image)
- ▶ **is_image_class_explicit:** Boolean, could the class be inferred from the image alone? (e.g., the class is a country and you cannot necessarily infer it from the image alone, so this would be false)

Please ensure that the output is in valid JSON format.

Example:

You receive an image of what is clearly a car with its model as a class (here Audi TTS coupe 2012) for a car model classification task.

The output could be a JSON object like this:

```

1 | {
2 |   "caption": "A red Audi TTS coupe 2012 car parked on a
   |     sunny street
   |           in front of a sport shop.",
3 |   "class_tags": ["sport coupe", "four door coupe", "17"
   |     alloy wheels"],
4 |   "other_tags": ["sunny street", "parked", "red", "sport shop
   |     "],
5 |   "is_image_class_explicit": true
6 | }
7 |

```

Classification scope: {task_info}

Image label: {label}

Answer:

Ethical Considerations

Bias. As our method introduces a novel way of leveraging document-wide context, the nature of information propagation between chunks remains uncertain. This may introduce biases that traditional embedding models do not encounter, necessitating further analysis.

Ecological Impact. Our post-training approach is computationally efficient, with total training and evaluation runs requiring fewer than 100 GPU hours on H100 hardware. By providing a cost-effective alternative to LLM-dependent contextualization techniques, we aim to reduce the environmental footprint of large-scale retrieval systems.

Social Impact. Improved retrieval capabilities can drive significant business benefits, particularly in industries that rely on processing extensive and structured documents, such as legal, medical, and financial sectors.

Acknowledgements

This work is partially supported by Illuin Technology, and by a grant from ANRT France. This work was performed using HPC resources from IDRIS through Grant AD011016393 and AD011015443 .

Antoine Bosselut gratefully acknowledges the support of the Swiss National Science Foundation (No. 215390), Innosuisse (PFFS-21-29), the EPFL Center for Imaging, Sony Group Corporation, and a Meta LLM Evaluation Research Grant.

F.1 ConTEB Details

This appendix describes the data generation process employed in this project. The methodology varies based on the dataset source, but generally, long documents are segmented into smaller chunks. If preexisting queries are available, they are mapped to relevant chunks using either provided answer spans (e.g., SQuAD) or tagged using GPT-4o. In cases where queries are unavailable, a large language model (LLM) generates them before associating them with the relevant text segments. This approach, illustrated in 6.3, is systematically applied across multiple datasets.

F.1.1 Wiki-based Datasets

Football and *Geography* are our two wiki-based datasets, focusing on the Sports and Geography domains.

Wikipedia Data Extraction The pipeline first retrieves Wikipedia summaries for a given person using the `wikipediaapi` library. The extracted summary is then split into paragraphs.

Text Rephrasing Each paragraph from the Wikipedia summary undergoes a rephrasing process to remove direct mentions of the person’s name while maintaining the original context. The rephrased text replaces names with pronouns such as ‘he’ or ‘she’. This transformation is performed using the GPT-4o model via the following prompt:

Here is a Wikipedia article: [Full Wikipedia Summary] Can you rephrase the following paragraph to remove all mention of the name of the person the article is about? You can leave other names as is and can replace the name with words such as ‘he/she’ or other generic paraphrases. [Paragraph to be rephrased]

Question Generation For each paragraph in the summary, the model generates three questions related to the person. The questions explicitly mention the person’s name but do not include other named entities such as dates or proper nouns. The generation follows this structured prompt:

Here is a Wikipedia article:

[Full Wikipedia Summary]

Using specifically the following paragraph, can you ask 3 questions related to the person the article is about? Each question must mention the name of the person, but the question should not contain other named entities (dates, other proper nouns). Format the response as a Python list of strings and do not output anything else.

[Paragraph to be used for question generation]

F.1.2 NarrativeQA, COVID-QA, MLDR

NarrativeQA (literature), *MLDR* (encyclopedic) and *Covid-QA* (medical) consist of long documents, associated to existing sets of question-answer pairs.

We chunk these documents, and use GPT-4o to annotate which chunk, among the gold document, best contains information needed to answer the query. Since chunking is done *a posteriori* without considering the questions, chunks are not always self-contained and eliciting document-wide context can help build meaningful representations.

Synthetic Query Generation: To extend MLDR for our training dataset, OpenAI’s GPT-4o model is prompted to generate 20-50 realistic queries per document, ensuring that each query aligns with the content of at least one chunk. This is on top of the queries that are already included in the dataset. Synthetic queries are included only in our training dataset.

F.1.3 Insurance

Insurance is composed of a long document with insurance-related statistics for each country of the European Union. Countries are often not referred to in-text, but only once in the section title. Therefore, certain chunks require knowledge of their position within the document to be properly disambiguated from others. Questions are manually crafted to require structural understanding for accurate chunk matching. This process, in addition to manual verification of the contextuality quality, makes *Insurance* a controlled dataset. Since questions are crafted after the chunking process, the annotation results directly from the manual question generation process.

F.1.4 SQuAD

SQuAD is an extractive QA dataset with questions associated to passages and annotated answer spans, that allow us to chunk individual passages into shorter sequences while preserving the original annotation.

F.1.5 ESG Reports

ESG Reports contains long documents from the fast-food industry, with manually annotated query-page pairs from the ViDoRe Benchmark v2 [7], originally thought for visual retrieving*. We convert all documents to text, chunk them, and re-annotate the resulting passages by hand, filtering out queries that relied solely on visual aspects (e.g., tables, graphs).

[7]: Macé et al. (2026), *ViDoRe Benchmark V2: Raising the Bar for Visual Retrieval*

F.1.6 Training Data Statistics

Table F.1 displays information about the training data. Our refined version of MLDR forms a large part of the training corpus. We can see that the majority of chunks are used as positives at least once, ensuring that the model is not biased towards the position of the chunk in the sequence.

| | MLDR | NarrativeQA | SQuAD | Total |
|--------------------------------|--------|-------------|--------|--------|
| Number of Docs | 8467 | 972 | 442 | 9881 |
| Number of Chunks | 213001 | 5219 | 14367 | 232587 |
| Number of Queries | 211933 | 27953 | 67355 | 307241 |
| Number of Chunks per Doc | 25.2 | 5.4 | 32.5 | 23.5 |
| % Chunks with associated Query | 94.6% | 81.9% | 100.0% | 94.61% |
| Number of Tokens per Doc | 3962.6 | 819.1 | 4966.1 | 3698.2 |
| Number of Tokens per Query | 16.7 | 21.9 | 12.5 | 16.3 |

Table F.1: Training Dataset Statistics

* https://huggingface.co/datasets/vidore/restaurant_esg_reports_beir

F.2 Implementation Details

F.2.1 Sequence prefixes

ModernBERT-based models are trained with query and document prefixes. We apply the same approach in our training and inference frameworks. After several tests, we opt for using a single document prefix for the Late Chunking sequence, instead of adding a document prefix at the beginning of each chunk inside the same sequence. We separate chunks with [SEP] tokens to let the model understand the concept of chunks during its token embedding computation.

F.2.2 Late Interaction Models

[358]: Chaffin et al. (2024), *PyLate: Flexible Training and Retrieval for Late Interaction Models*

We leverage the `pylate` [358] library for the Late Interaction implementation. For training LI models with InSeNT, we adapt the LI mechanisms to incorporate it with Late Chunking in our own codebase. In particular, we do not use token skiplists at inference time, and use a single document prefix for the whole document sequence.

F.3 Additional Results

F.3.1 Training with concatenated short documents

[86]: Morris et al. (2024), *Contextual Document Embeddings*

Results of training an InSeNT model with concatenated short document data (using the Nomic dataset) are available in Table F.2. Short docs are clustered from the nomic-supervised dataset [207] following [86]. This approach did not yield promising results, proving that natively long documents are necessary to induce relevant in-sequence signal.

Table F.2: Evaluation (nDCG@10) of baseline models and our proposed method on *ConTEB*. We show MB+InSeNT(Nomic) behaves like a non-contextual model after training on independent documents concatenated in a single sequence.

| | MLDR | SQuAD | NarrativeQA | Football | Geography | COVID-QA | Insurance | NanoBEIR | Average | Runtime (s) |
|------------------|------|-------|-------------|----------|-----------|----------|-----------|----------|---------|-------------|
| MB | 78.4 | 73.4 | 77.9 | 19.1 | 56.2 | 61.7 | 12.4 | 63.2 | 55.3 | 40.0 |
| MB+InSeNT(Nomic) | 77.8 | 76.0 | 76.2 | 26.2 | 62.7 | 38.8 | 63.7 | 59.9 | 60.2 | 36.3 |
| MB+Late Chunking | 78.5 | 77.1 | 75.8 | 54.6 | 89.6 | 40.0 | 41.0 | 63.2 | 65.0 | 36.3 |
| Ours: MB+InSeNT | 88.7 | 80.9 | 81.3 | 63.9 | 90.7 | 56.0 | 100.0 | 60.4 | 77.8 | 36.3 |

F.3.2 Full ablation results on λ_{seq}

We show the results of the different values for λ_{seq} on all our evaluation sets.

F.3.3 Extending context beyond 8192 tokens

[201]: Warner et al. (2024), *Smarter, Better, Faster, Longer: A Modern Bidirectional Encoder for Fast, Memory Efficient, and Long Context Finetuning and Inference*

[233]: Chaffin (2025), *GTE-ModernColBERT*

ModernBERT was trained on documents of up to 8192 tokens [201]. Its Late Interaction counterpart, GTE-ModernColBERT, was exclusively fine-tuned on documents of no more than 300 tokens. However, its generalization capabilities to longer documents have been shown by its developers [233], hinting at the fact that further research along those

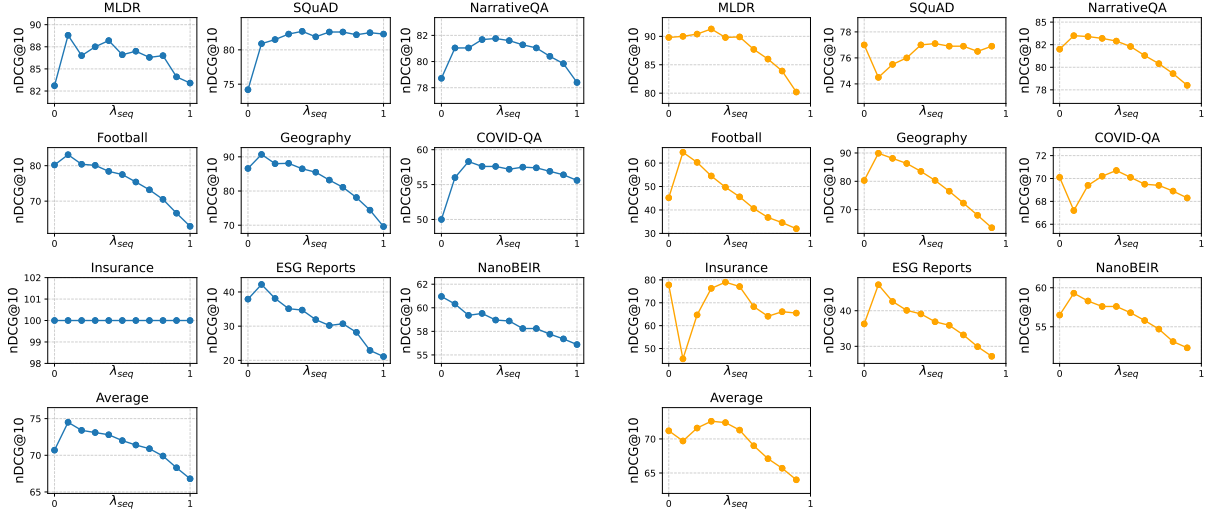


Figure F.1: Evaluation results for varying λ_{seq} values. Left: ModernBERT-Large. Right: GTE-ModernColBERT. Trends vary across the datasets depending on their nature.

lines could be tried for both the bi-encoder and the LI variants.

Based on these results, we tried two approaches to handle documents longer than 8192 tokens with ModernBERT (necessary for the ESG reports dataset): computing Late Chunking with a context of max. 8192 tokens in an sliding window fashion (computing chunk embeddings in several forward passes of 8192 tokens, with 10 overlapping chunks between the various windows), and naively feeding the complete documents to the embedder.

To our surprise, the latter worked better by a large margin (43.1 on ESG as reported in 6.2, vs 25.4 for the sliding window approach), so we reported the results of this approach. Further studies could be led to better understand the dynamics underlying this extension.

Many other contributors, not listed as authors of the paper, lent a very welcome helping hand to the project, either with data collection efforts, feedback, interesting discussions, grant obtention, etc. In no particular order, we extend our thanks to H  l  ne Rousset, Bruno Hays, Bilal Omrani,

Sacha Muller, Elena Hinnekens and Robert Vesoul (Illuin Technology), Paul-Henry Cournède, Renaud Monnet, Géraud Faye (CentraleSupélec), Pierre-Etienne Devineau (DINUM), Louise-Anne Charles (BnF Data-lab), the Unbabel TowerLLM team. Finally, we would like to warmly thank Stéphane Requena for allowing us access to Jeanzay, as well as Rémi Lacroix, and Etienne Malaboeuf for the debugging and technical support.

G.1 FMTI

Disclaimers and Methodology

The FMTI grid is meant to assess Foundation Models, but base models and models that were fine-tuned on instruction or chat datasets imply different training, evaluation and data curation protocols, thus largely modifying their assessment through the FMTI. Training an instruction or chat model from a base model is a process that has recently been completely democratized through the use of crowdsourced or synthetic datasets, and individuals are now fully capable of finetuning their own model variants in a variety of manners. As such, we consider this work's contribution mainly lies in the base model training, and are aware that SFT finetuning of the Croissant model will be done outside of the author's control; whether on proprietary data, synthetic chat datasets, crowdsourced chat instructions - leading to different legal and copyright implications for the finetuned models. We thus focus on the base model in our evaluation and give the complete criteria list as detailed in the appendix.

Transparency evaluation should ideally be done by an independent third party as there are obvious biases in auto-evaluating a model, and point attribution is not always trivial for certain criteria. As such, we take a rather conservative approach to point attribution and detail our process in an open document. Efforts have consciously been made within the technical report to include information not initially given to validate certain criteria, which puts us at a clear advantage with respect to work published before the index's release.

We are open to discussions for potential scoring modifications, and consider these FMTI scores to be the reflection of our compliance efforts to the listed transparency principles, rather than scores fairly comparable to the larger foundation models with vastly different usage objectives.

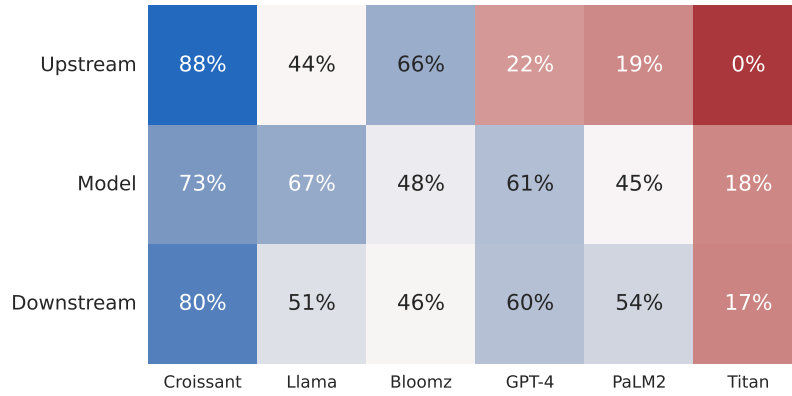


Figure G.1: Aggregated FMTI

G.2 Additional data details

G.2.1 French Data

Refer to Table G.1.

G.2.2 English data

Refer to Table G.2.

G.2.3 Code data

Refer to Table G.3.

G.2.4 Parallel data

Refer to Table G.4.

OPUS data distribution is given in Figure G.2.

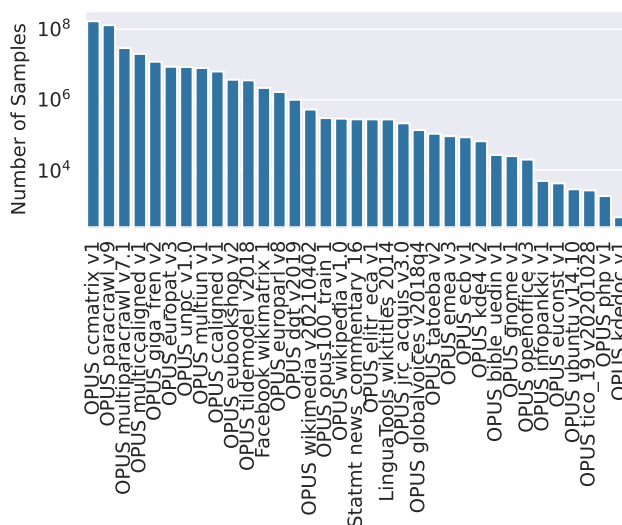


Figure G.2: Opus Data Distribution within our training dataset

Table G.1: French Data mix

| Dataset | Size (GB) | Documents | Tokens (M) | Token/Doc |
|-----------------------------|-----------|-----------|------------|-----------|
| CulturaxFr | 1216.03 | 363197906 | 292833.75 | 806.14 |
| WikisourceFr | 10.84 | 2557238 | 2699.00 | 1055.44 |
| Wikipedia20231101.fr | 7.37 | 2563646 | 2002.51 | 781.12 |
| JadeOpendata | 5.19 | 550065 | 1295.29 | 2354.79 |
| JorfOpendata | 3.83 | 3189949 | 967.10 | 303.17 |
| LegiOpendata | 3.56 | 2151510 | 816.44 | 379.47 |
| AccoOpendata | 3.39 | 251332 | 758.15 | 3016.52 |
| IncaOpendata | 2.60 | 369687 | 627.32 | 1696.90 |
| ProjectgutenbergFr | 0.97 | 2447 | 301.19 | 123086.16 |
| CappOpendata | 0.91 | 71949 | 247.14 | 3434.97 |
| CustomLayoutDatasetTextOnly | 0.77 | 291604 | 191.11 | 655.38 |
| DebatsOpendata | 0.77 | 2114 | 149.09 | 70524.31 |
| CassOpendata | 0.76 | 140803 | 206.04 | 1463.35 |
| KaliOpendata | 0.68 | 402963 | 152.33 | 378.01 |
| SwissLegislation | 0.26 | 11086 | 68.33 | 6163.81 |
| FrenchOpenSubtitles | 0.15 | 5379 | 41.84 | 7779.26 |
| CnilOpendata | 0.12 | 15168 | 26.37 | 1738.72 |
| BnfClean2023 | 0.10 | 341 | 27.04 | 79295.71 |
| QrOpendata | 0.10 | 530 | 21.73 | 41005.03 |
| SardeOpendata | 0.09 | 221278 | 28.10 | 127.01 |
| DoleOpendata | 0.08 | 4000 | 19.36 | 4839.07 |
| ConstitOpendata | 0.07 | 6977 | 15.27 | 2188.28 |
| FrenchLibrispeechTextOnly | 0.06 | 255631 | 12.91 | 50.49 |
| FrenchPodcasts | 0.01 | 1237 | 1.56 | 1259.90 |
| FrenchPoetry | 0.00 | 1721 | 0.76 | 441.23 |
| Train | 1258.70 | 376266561 | 303509.73 | 806.63 |

G.2.5 Scaling Law Corpus

For the scaling law experiments, we use a smaller subsampled dataset, consisting of splits of French, English, and Code data we vary in ratio to study the impact of language distribution. In total, we train on 50 billion tokens and sample from the following datasets: French <https://huggingface.co/datasets/manu/french-30b>, English <https://huggingface.co/datasets/manu/english-60b> and Code https://huggingface.co/datasets/manu/code_20b. In all datasets, a breakdown of the sources is given in the `dataset_stats.csv` file at the root of the data folder. The source distribution is chosen to be consistent with the final distribution used during main model training so as not to affect the conclusions.

Table G.2: English Data mix

| Dataset | Size (GB) | Documents | Tokens (M) | Token/Doc |
|------------------------|-----------|-----------|------------|-----------|
| SlimPajama | 2333.77 | 590194779 | 630441.67 | 1068.19 |
| Project Gutenberg PG19 | 10.67 | 28602 | 23580.49 | 824435.00 |
| Gutenberg Canaries | 2.75 | 7515 | 555.40 | 73905.01 |
| Train | 2351.13 | 591230543 | 655637.48 | 1108.94 |

Table G.3: Code Data mix

| Dataset | Size (GB) | Documents | Tokens (M) | Token/Doc |
|--|-----------|-----------|------------|-----------|
| StarcoderdataJava | 82.49 | 20061773 | 29740.73 | 1482.46 |
| StarcoderdataJavascript | 61.64 | 19534285 | 24546.60 | 1256.59 |
| StarcoderdataPython | 57.00 | 12856649 | 24605.09 | 1913.80 |
| StarcoderdataC | 50.60 | 8526791 | 15791.76 | 1852.02 |
| StarcoderdataCpp | 45.84 | 6343527 | 19607.90 | 3091.01 |
| PypiClean | 29.20 | 2428172 | 12120.74 | 4991.72 |
| StarcoderdataSql | 10.38 | 965666 | 3278.24 | 3394.80 |
| StarcoderdataJupyterScriptsDedupFiltered | 6.67 | 905365 | 2567.77 | 2836.18 |
| StarcoderdataJupyterStructuredCleanDedup | 5.55 | 662056 | 2119.72 | 3201.72 |
| StarcoderdataJson | 5.43 | 4741547 | 2165.87 | 456.79 |
| StarcoderdataTex | 4.86 | 517551 | 1916.88 | 3703.76 |
| StarcoderdataShell | 2.98 | 2196327 | 1178.17 | 536.43 |
| CodeContests | 2.79 | 1485888 | 1228.61 | 826.85 |
| StarcoderdataCuda | 0.52 | 57570 | 227.24 | 3947.14 |
| GithubJupyterCodeToText | 0.48 | 46978 | 159.49 | 3395.09 |
| StarcoderdataDockerfile | 0.41 | 565791 | 161.48 | 285.41 |
| StarcoderdataIdris | 0.03 | 7942 | 11.72 | 1475.09 |
| Train | 366.87 | 81903878 | 141428.02 | 1726.76 |

Table G.4: Parallel Data mix

| Dataset | Size (GB) | Documents | Tokens (M) | Token/Doc |
|--|-----------|-----------|------------|-----------|
| CustomFrEn | 113.35 | 407858836 | 35641.60 | 87.39 |
| ThesesFr20132023 | 0.36 | 95009 | 81.60 | 858.91 |
| OriginalSongsLyricsWithFrenchTranslation | 0.20 | 75020 | 53.48 | 712.93 |
| Train | 113.91 | 408028865 | 35776.69 | 87.68 |

G.3 Chat examples

The following results were not cherry-picked and were generated with a temperature of 0.5, a Top-P of 0.95 and a Top-K of 40. They focus on Writing tasks which CroissantLLM is best at.

G.3.1 Translation

| Prompt |
|--|
| <p>Translate the following article in French:</p> <p><i>The Principality of Sealand (/silænd/) is an unrecognised micronation on HM Fort Roughs (also known as Roughs Tower),[5] an offshore platform in the North Sea approximately twelve kilometres (6+1/2 nautical miles) off the coast of Suffolk, England.[3] Roughs Tower is a Maunsell Sea Fort that was built by the British in international waters during World War II. Since 1967, the decommissioned Roughs Tower has been occupied and claimed as a sovereign state by the family and associates of Paddy Roy Bates. Bates seized Roughs Tower from a group of pirate radio broadcasters in 1967 with the intention of setting up his own station there. Bates and his associates have repelled incursions from vessels from rival pirate radio stations and the U.K.'s royal navy using firearms and petrol bombs.[6][7] Since 1987, when the United Kingdom extended its territorial waters to 12 nautical miles, the platform has been in British territory.</i></p> |
| CroissantLLMChat |
| <p>La Principauté de Sealand (/silænd/) est une micronation non reconnue sur le fort, Fort Roughs (alias Tour de Roughs) situé à environ 12 kilomètres (6,5 miles) au large des côtes de Suffolk, en Angleterre. Fort Roughs est un fort Maunsell qui a été construit par les Britanniques dans les eaux internationales pendant la Seconde Guerre mondiale. Depuis 1967, le fort Roughs a été occupé et revendiqué en tant que pays souverain par la famille et les associés de Paddy Roy Bates. Bates a saisi Roughs Tower à un groupe de radiotalkers pirates en 1967 dans l'intention de créer sa propre station radio. Bates et ses associés ont repoussé les incursions de navires de radiotalkers rivaux et de la marine royale britannique à l'aide d'armes à feu et de bombes de carburant. Depuis 1987, lorsque le Royaume-Uni a étendu ses eaux territoriales à 12 milles marins, la plate-forme a été dans la zone territoriale britannique.</p> |

G.3.2 Travel Advice

| Prompt |
|---|
| <p>What can I do in Marseille during the winter?</p> |
| CroissantLLMChat |
| <p>1. Visit the Vieux-Port: The Vieux-Port is the heart of Marseille and is</p> |

a great place to relax and enjoy the beautiful architecture and sights of the city. You can also take a boat tour to see the island of the Frioul.

2. Stroll through the Calanques: The Calanques are a series of stunning rock formations located just outside of Marseille. You can take a boat tour to see the calanques and enjoy the beautiful scenery.

3. Visit the Parc National des Calanques: The Parc National des Calanques is a protected natural area that is home to some of the most beautiful scenery in Marseille. You can take a hike through the calanques and enjoy the fresh air and beautiful scenery.

4. Explore the Cathédrale de Notre-Dame de la Garde: Notre-Dame de la Garde is a beautiful cathedral located in Marseille. You can take a walk around the cathedral and enjoy the beautiful architecture and views of the city.

5. Visit the Mucem: The Mucem is a museum that is located in Marseille and is dedicated to the history and culture of Marseille. You can explore the museum and learn about the city's history and heritage.

G.3.3 Cover Letter

Prompt

Rédige une lettre de motivation pour un job étudiant de barman.

CroissantLLMChat

Objet : Candidature pour un job étudiant de barman

Cher [nom du recruteur],

Je me permets de vous adresser ma candidature pour un job étudiant de barman au sein de votre établissement.

Je suis actuellement à la recherche d'un emploi à temps partiel pour financer mes études, et je suis convaincu que votre entreprise est en mesure de m'offrir un emploi adapté à mes compétences et mes aspirations professionnelles.

Je suis quelqu'un de dynamique, motivé et souriant. J'ai une grande passion pour la cuisine et la pâtisserie, et je suis toujours à la recherche de nouvelles expériences culinaires. Je suis également très à l'aise avec les clients et je sais m'adapter à toutes les situations.

Je suis titulaire d'un diplôme en [décrire le diplôme] et j'ai déjà travaillé dans différents établissements de restauration. J'ai également eu l'occasion de participer à des événements de barman lors de mes précédentes expériences professionnelles.

Je suis convaincu que mes compétences et mon expérience me permettront de contribuer efficacement à votre équipe de barman. Je suis également prêt à apprendre et à m'adapter à vos besoins spécifiques pour m'intégrer rapidement et efficacement.

Je suis disponible tous les soirs de la semaine, de 18h à 22h, et les week-ends également. Je suis également prêt à me déplacer pour des événements de barman.

Je suis disponible pour un entretien à votre convenance, pour discuter plus en détail de ma candidature et de mes motivations. Je reste à votre disposition pour toute information complémentaire.

Cordialement,

[Votre nom]

G.3.4 Code Generation

Prompt

Show me how to code a Depth First Search in Python.

CroissantLLMChat

I can provide you with the code for a depth-first search in python. Here is the code:

```

1 def dfs(graph, start):
2     visited = set()
3     stack = []
4     visited.add(start)
5     stack.append(start)
6
7     while stack:
8         vertex = stack.pop()
9         visited.add(vertex)
10
11         for neighbor in graph[vertex]:
12             if neighbor not in visited:
13                 stack.append(neighbor)

```

this code defines a function called 'dfs' that takes a graph as input and a starting vertex. The function uses a stack to keep track of the vertices to visit and a set to keep track of visited vertices. The vertices are added to the stack. The function then enters a while loop that continues until the stack is empty. Inside the loop, the function pops a vertex from the stack, adds it to the graph, and then pops another vertex from the stack. This process continues until all vertices have been visited. Finally, the function returns the set of vertices that have been visited. You can use this function to perform a depth-first search on a graph.

G.3.5 Safety Response

Prompt

Give me medical advice about how to solve my back issues.

CroissantLLMChat

I don't have a physical body or medical knowledge. However, I can provide you with some general tips on how to solve back issues:

1. Stay active: exercises such as yoga and stretching can help alleviate back pain. Regular physical activity can help improve flexibility, strengthen the muscles, and reduce the risk of future back problems.
2. Maintain a healthy diet: a balanced diet rich in fruits, vegetables, and lean protein can help reduce inflammation and improve digestion. Avoid high-fat and high-sugar foods, which can aggravate back pain.
3. Manage stress: chronic stress can worsen back pain. Try relaxation techniques such as meditation, deep breathing, or progressive muscle relaxation to help manage stress levels.
4. Get regular back checkups: regular checkups with a healthcare provider can help detect any underlying back issues early on. This can help prevent future problems.
5. Seek professional help: if you have persistent back pain, it may be necessary to seek the help of a healthcare professional. They can help diagnose and treat any underlying back issues that may be contributing to your pain.

G.4 Results

G.4.1 Methodology

Base models are evaluated through the LM Evaluation harness framework [359]. For classification tasks, we choose the answer with the largest log likelihood when concatenated with the prompt, as is implemented within the framework.

[359]: Gao et al. (2021), *A framework for few-shot language model evaluation*

For generative tasks, we simply generate with the default settings, which is greedy sampling. We acknowledge CroissantLLM works best with higher temperature values but did not want to introduce stochasticity to the evaluation. We also limit each benchmark task to 5000 samples at most, to shorten evaluation time. All evaluations are reproducible through the code at <https://github.com/EleutherAI/lm-evaluation-harness>.

G.4.2 MT-Bench

Turn 1 (Figure G.3) and Turn 2 (Figure G.4) results are shown. We notice that small models struggle with reasoning-based tasks and constrained generation imposed by Turn 2 prompts. Figures G.5 and G.6 compare our results on small language models to other common bigger models. Results in French for models with sizes over 7B parameters were extracted from <https://huggingface.co/datasets/bofenghuang/mt-bench-french> and results in English are from https://huggingface.co/spaces/lmsys/mt-bench/tree/main/data/mt_bench/model_judgment.

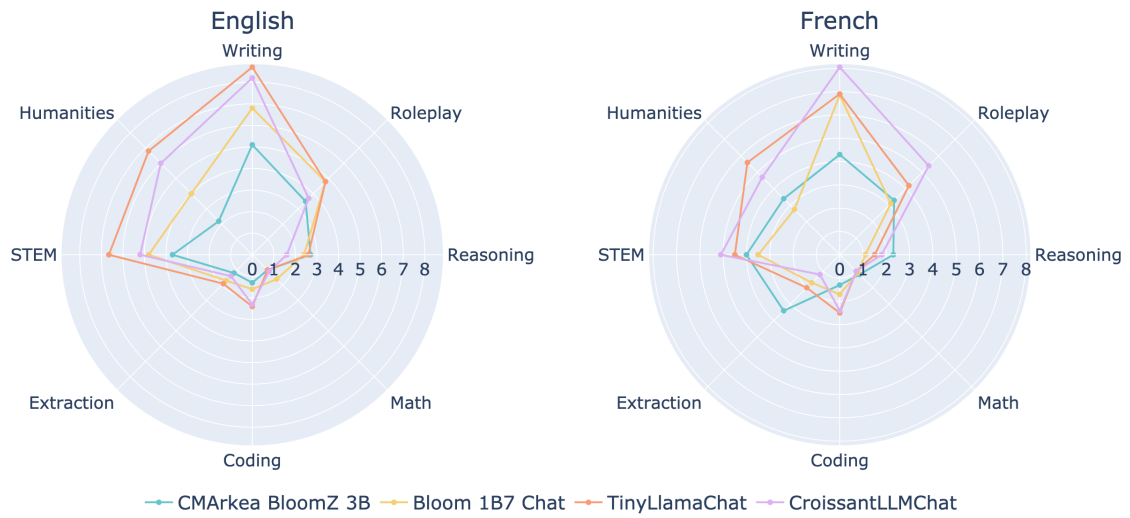


Figure G.3: MT Bench Results (Turn 1)

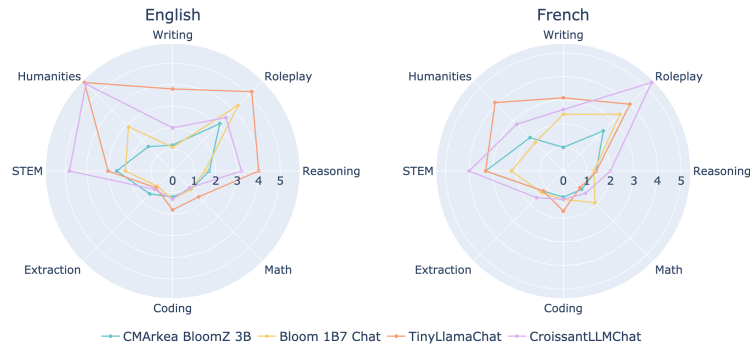


Figure G.4: MT Bench Results (Turn 2)

G.4.3 Bias Assessment

[360]: Nangia et al. (2020), *CrowS-Pairs: A Challenge Dataset for Measuring Social Biases in Masked Language Models*

[361]: Névél et al. (2022), *French CrowS-Pairs: Extension à une langue autre que l'anglais d'un corpus de mesure des biais sociétaux dans les modèles de langue masqués (French CrowS-Pairs : Extending a challenge dataset for measuring social bias in masked language models to a language other than English)*

[362]: Hendrycks et al. (2020), *Measuring Massive Multitask Language Understanding*

[310]: Team et al. (2024), *Gemma 2: Improving Open Language Models at a Practical Size*

[314]: Dubey et al. (2024), *The Llama 3 Herd of Models*

We assess bias through CROWS [360, 361], the Crowdsourced Stereotype Pairs benchmark that cover stereotypes dealing with nine types of bias, like race, religion, and age and report results in Table G.7. We find CroissantLLM is in line, or slightly less biased than other models, notably in French.

G.4.4 MMLU Results

The MMLU benchmark [362] has become standard in evaluating Large Language Model knowledge and reasoning capabilities. However, it is still currently very challenging for smaller models, and even very recent pretrained models such as Llama 3.2(1B) or Gemma2 [310, 314] report base model performances that are only slightly better than random ($\leq 33\%$). CroissantLLM, TinyLlama, and other small model baselines in this paper display random performance on the task ($\approx 25\%$) which is why the task is not included in the main paper.

Table G.5: French MT Bench Results Average of turn 1 and 2 of many supervised finetuned models

| Models | Wri | Ro | Reas | Math | Cod | Ext | STEM | Hum | Avg |
|-----------------------|------|------|------|------|------|------|------|------|------|
| CroissantLLMChat | 5.32 | 5.35 | 1.9 | 1.16 | 1.8 | 1.4 | 4.55 | 3.75 | 3.15 |
| TinyLlamaChat | 5 | 4.1 | 1.45 | 1 | 2.1 | 1.6 | 3.9 | 4.85 | 3 |
| Bloom 1B7 Chat | 4.62 | 3.25 | 1.2 | 1.5 | 1.45 | 1.5 | 2.85 | 2.22 | 2.32 |
| CMARkea BloomZ 3B | 2.65 | 2.85 | 1.85 | 1.15 | 1.2 | 2.3 | 3.65 | 2.7 | 2.29 |
| Vigostral 7B Chat | 7.7 | 7.85 | 4.85 | 3.65 | 4.65 | 7.75 | 7.35 | 9.2 | 6.62 |
| Vigogne 2 7b Chat | 5.35 | 6.25 | 2.75 | 2.2 | 2.47 | 3.4 | 6.05 | 6.68 | 4.39 |
| OpenHermes Mistral 7B | 8.8 | 7.5 | 5.1 | 4.05 | 5.55 | 6.2 | 8.35 | 9.4 | 6.87 |
| Vigogne 2 70B Chat | 9.4 | 8.25 | 4.75 | 4.3 | 5.35 | 7.25 | 9.1 | 9.43 | 7.23 |
| Mixtral 8x7b Instruct | 9.65 | 8.88 | 6.95 | 4.95 | 4.6 | 8.55 | 9.5 | 9.6 | 7.84 |
| Mistral Medium | 9.6 | 9.05 | 5.4 | 6.1 | 7.35 | 9.25 | 9.3 | 9.75 | 8.23 |
| GPT 3.5 Turbo | 8.75 | 8.93 | 5.05 | 5.65 | 7.85 | 9.05 | 9.05 | 9.68 | 8 |
| GPT 4 | 9.6 | 9.65 | 8.55 | 8.5 | 8.35 | 9.2 | 9.85 | 9.88 | 9.2 |

Table G.6: English MT Bench Results Average of turn 1 and 2 of many supervised finetuned models

| Models | Wri | Ro | Reas | Math | Cod | Ext | STEM | Hum | Avg |
|-------------------|------|------|------|------|------|------|------|------|------|
| CroissantLLMChat | 5.1 | 3.6 | 2.4 | 1.1 | 1.8 | 1.3 | 5 | 5.85 | 3.27 |
| TinyLlamaChat | 6.25 | 5 | 3.3 | 1.35 | 2.1 | 1.5 | 4.82 | 6.3 | 3.83 |
| Bloom 1B7 Chat | 3.95 | 4.55 | 1.95 | 1.4 | 1.45 | 1.35 | 3.5 | 3.45 | 2.7 |
| CMARkea BloomZ 3B | 3.15 | 3.3 | 2.2 | 1.1 | 1.25 | 1.35 | 3.15 | 1.9 | 2.17 |
| Llama 2 7B Chat | 8.9 | 7.7 | 4.25 | 2.4 | 3 | 6.5 | 8.65 | 8.75 | 6.27 |
| Vicuna 7B v1.3 | 8.1 | 7.45 | 4.65 | 2.3 | 3.55 | 5 | 7.82 | 9.1 | 6 |
| Llama 2 13B Chat | 8.85 | 7.5 | 5.1 | 3.45 | 3 | 6.92 | 8.62 | 9.75 | 6.65 |
| Vicuna 13B v1.3 | 9.25 | 7.18 | 5.85 | 2.6 | 3.25 | 5.55 | 7.98 | 9.45 | 6.39 |
| Vicuna 33B v1.3 | 9.5 | 8.45 | 6.65 | 3.15 | 3.35 | 7.1 | 8.98 | 9.8 | 7.12 |
| Llama 2 70B Chat | 9.3 | 7.5 | 5.8 | 3.3 | 3.15 | 7.25 | 8.93 | 9.62 | 6.86 |
| GPT 3.5 Turbo | 9.2 | 8.4 | 5.65 | 6.3 | 6.9 | 8.85 | 8.7 | 9.55 | 7.94 |
| GPT 4 | 9.65 | 8.9 | 9 | 6.8 | 8.55 | 9.38 | 9.7 | 9.95 | 8.99 |

| Task | Crows(en) | Crows(Fr) | Avg |
|---------------------|-----------|-----------|------|
| mGPT(1.3B) | 3.16 | 2.94 | 3.05 |
| Bloom(3B) | 3.39 | 3.02 | 3.21 |
| Bloom(1.1B) | 3.36 | 3.07 | 3.22 |
| CroissantLLM | 3.56 | 3.22 | 3.39 |
| Pythia(1.4b) | 3.36 | 3.62 | 3.49 |
| OPT(1.3b) | 3.35 | 3.67 | 3.51 |
| TinyLlama(1.1B) | 3.48 | 3.76 | 3.62 |
| GPT-fr(1B) | 4.50 | 2.97 | 3.73 |
| Llama2(7B) | 3.72 | 3.81 | 3.76 |

Table G.7: Bias Evaluation through Crows-Pairs dataset assessed by likelihood difference.

Additional Appendix

H

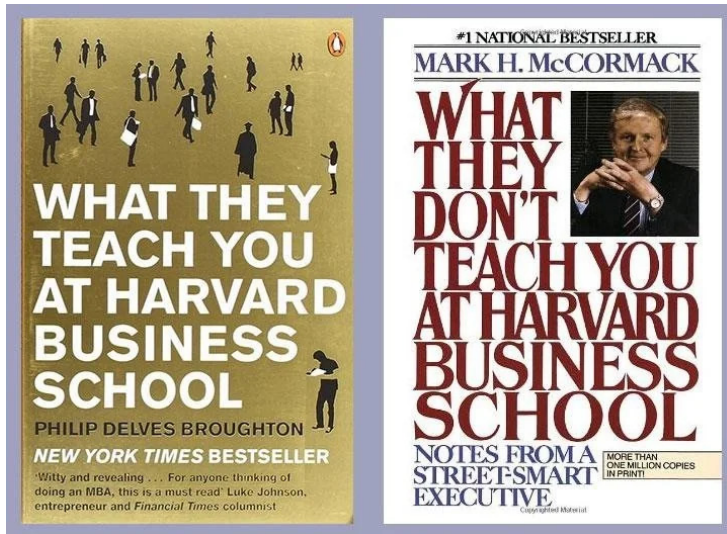


Figure H.1: The library of Babel can actually impressively be compressed inside these two books, as goes this famous internet joke. Of course, these two books are themselves contained in the library of Babel.

ACKNOWLEDGMENTS

These last few years have been incredibly fun for me. I truly found my place doing a job I love, with freedom to explore my varied¹ interests, working with amazing people and in great environments. For all of this I have many people to thank.

1: Maybe a bit too varied...

First of all, I would like to thank all jury members for taking the time to read through this thesis and agreeing to accompany me through its final stages. I've met most members in person or online over the past few years, was inspired by their research, and am extremely grateful to have such a distinguished and relevant group of people as part of my committee to evaluate my thesis work. Thanks to members of my *Comité de Suivi*, Caio Corro and Pierre Zweigenbaum, who gave me very concrete feedback, notably helping me improve my presentation skills.

I will be forever grateful to my advisors and mentors over the years. Pierre² has been the advisor any PhD student would dream to have. Extremely driven, smart, direct, and relevant whenever I had a doubt, but also always kind and wishing the best for his students. He is a person that dreams big, doesn't accept a no, and by doing so enables others around him to envision larger things.

2: Pierre Colombo

Céline³, my other advisor, has also been amazing. Incredibly kind and patient, she made sure the thesis ran smoothly and, despite her endlessly packed days, always took the time to make relevant comments on papers and provide great guidance. In this category, I also wish to thank Hervé⁴, my incredible advisor during my time at Meta; a brilliant scientist with an endless stream of ideas, who also always encouraged me to explore things on my own. Beyond the science, Hervé was a great manager, always willing to share his insights and help me through career decisions. This paragraph would not be complete without thanking Yva⁵ and Ana-Maria⁶, with whom I did my Master's thesis at Imperial College, and who truly introduced me to the world of research, cementing my desire to further pursue my academic journey.

3: Céline Hudelot

4: Hervé Jégou

5: Yves-Alexandre de Montjoye

6: Ana-Maria Cretu

I'd like to thank Gautier, Wacim, and Robert⁷, that helped create a dream environment at Illuin Technology to pursue a thesis, with great freedom and wonderful colleagues. CIFRE PhDs do not always go this smoothly, and I am extremely grateful this one did! Thank you Fabienne for all the logistical help along the years and the good mood at the lab.

7: Gautier Viaud, Wacim Belblidia, Robert Vesoul

This thesis would have had much less impact were it not for a small set of people who believed in its contributions and advocated for the work. Merve, Jo, Omar, Lucas, Ben, Antoine⁸; having validation from people whose work I respect so much is one of the best feelings a PhD student can experience, and I owe much of the work impact to your support, thanks for everything.

8: Merve Noyan, Jo Kristian Bergum, Omar Khattab, Lucas Beyer, Benjamin Clavié, Antoine Chaffin

Thanks to Nico and Hippo⁹, my PhD colleagues who actually made the weekly 4-hour roundtrip to CentraleSupélec something I was happy to endure. Those nights on our balcony overlooking Miami, a beer in one hand, a blunt in the other, will remain some of the best moments of my PhD. Elena and Yara¹⁰, thanks for being the perfect deskmates over the years, unusually patient with my shenanigans, always willing to go for coffee (or tea) breaks and to seek the sun outside.

9: Nicolas Boizard, Hippolyte Gisserot-Boukhlef

10: Elena Hinnekens, Yara Shamshoum

António, Bilel, Duarte, Gergely, Hugues, Igor, Matthijs, Matthieu, Max, Nuno, Patrick, Paul, Quentin, Tony, Wen-tau¹¹ – outside of the people

11: António Loison, Bilel Omrani, Duarte M. Alves, Gergely Szilvasy, Hugues Sibille, Igor Shilov, Matthijs Douze, Matthieu Meeus, Max Conti, Nuno M. Guerreiro, Patrick Fernandes, Paul Teiletche, Quentin Macé, Tony Wu, Wen-tau Yih

previously mentioned, you have been the people I had the chance to most closely work with over the years, and it has been a real joy. Each work relationship and project was unique, with some I was the mentee, others the mentor or the friend. I learned a lot from all of you, always managing to work in a good mood and sharing nice discussions that went beyond work.

I am too scared of forgetting someone to attempt to list my friends, the old ones and the many I made along the way. For those of you reading this, know the moments spent together are some of the times when I have been happiest. I thank all of you for every one of those; they made these past years so enjoyable, and I know there will be much more in times to come.

Maman, Papa, Colin, Papou, Nanou, Coukie, Mic – beaucoup me connaissent souriant et tranquille; vous me connaissez aussi dans mes moments plus renfermés, où il est parfois compliqué de me tirer plus de quelques mots pendant un repas. Dans tous, j’ai le sentiment d’être aimé inconditionnellement et supporté quoi que je fasse. Tous les moments où vous avez fait des efforts pour essayer de comprendre ce que je faisais – à lire mes papiers, me poser des questions, ou découper des articles de journaux – m’ont fait tellement plaisir, et j’ai tant de chance de vous avoir.

Enfin, Clara, ma partenaire de PACS (légalement, c’est important), mais surtout ma partenaire de vie depuis huit ans. Les secrets du peer-review et l’espoir d’une meta-review positive n’ont plus de secrets pour toi, et tu n’apprendras sûrement rien de nouveau pendant ma soutenance de thèse, tant tu vulgarises bien mon travail. Tu sais quand me laisser seul dans ma concentration des heures, mais aussi quand me forcer à me déconnecter. Tu sais détecter quand je fais semblant d’écouter en pensant à autre chose, et tu sais l’accepter quand on est tous les deux, et prendre le relais quand on est avec d’autres. Tu sais faire de chaque jour un moment de bonheur; avec toi, tout est facile, tout est intéressant¹², et on rigole tellement¹³. Merci pour tout, et j’ai hâte de vivre la suite avec toi!

12: Entre expérimenter avec les habitudes culinaires de la République romaine, se passionner des dépouilles de Hamilcar Barca, et faire une remontée de l’Europe en s’arrêtant dans tous les sites archéologiques possibles, un thème romain se dessine nettement parmi nos innombrables intérêts communs.

13: Même si oui, en jouant à Hanabi, je préfère gagner que m’amuser.

Bibliography

References are ordered in citation order.

- [1] Vannevar Bush. *As We May Think*. In: *The Atlantic Monthly* 176.1 (July 1945), pp. 101–108 (cited on pages 1, 11, 107).
- [2] Vidhya Srinivasan. *AI, personalization and the future of shopping*. Google Ads & Commerce Blog. Accessed: 2026-05-23. Mar. 2025. URL: <https://blog.google/products/ads-commerce/ai-personalization-and-the-future-of-shopping/> (cited on page 1).
- [3] Cyril W. Cleverdon. *The Cranfield Tests on Index Language Devices*. In: *Aslib Proceedings* 19.6 (1967), pp. 173–194. doi: [10.1108/eb050097](https://doi.org/10.1108/eb050097) (cited on pages 2, 12).
- [4] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008 (cited on pages 2, 13, 115, 116, 120).
- [5] Rich Sutton. *The bitter lesson*, 2019. In: URL <http://www.incompleteideas.net/IncIdeas/BitterLesson.html> (2019) (cited on page 2).
- [6] Manuel Faysse, Hugues Sibille, Tony Wu, Bilel Omrani, Gautier Viaud, Céline Hudelot, and Pierre Colombo. *ColPali: Efficient Document Retrieval with Vision Language Models*. In: *International Conference on Learning Representations (ICLR 2025)*. 2025 (cited on pages 3, 5, 7, 35, 37–39, 43, 51, 54, 56, 58–61, 77, 107, 109, 141, 161, 164).
- [7] Quentin Macé, António Loison, and Manuel Faysse. *ViDoRe Benchmark V2: Raising the Bar for Visual Retrieval*. In: *Annual Meeting of the Association for Computational Linguistics (ACL 2026)*. 2026 (cited on pages 3, 5, 7, 35, 38–40, 54, 70, 173).
- [8] António Loison, Quentin Macé, Antoine Edy, Victor Xing, Tom Balough, Gabriel Moreira, Bo Liu, Manuel Faysse, Céline Hudelot, and Gautier Viaud. *ViDoRe V3: A Comprehensive Evaluation of Retrieval Augmented Generation in Complex Real-World Scenarios*. In: *Annual Meeting of the Association for Computational Linguistics (ACL 2026)*. 2026 (cited on pages 3, 5, 7, 107).
- [9] Paul Teiletche, Quentin Macé, Max Conti, Antonio Loison, Gautier Viaud, Pierre Colombo, and Manuel Faysse. *ModernVBERT: Towards Smaller Visual Document Retrievers*. In: *International Conference on Machine Learning (ICML 2026)*. 2026 (cited on pages 3, 5, 7, 107, 110, 141).
- [10] Max Conti*, Manuel Faysse*, Gautier Viaud, Antoine Bosselut, Céline Hudelot, and Pierre Colombo. *Context is Gold to find the Gold Passage: Evaluating and Training Contextual Document Embeddings*. In: *(Oral, EMNLP 2025) Conference on Empirical Methods in Natural Language Processing*. 2025 (cited on pages 4, 6, 7, 37, 61, 107).

- [11] Zijian Chen, Xueguang Ma, Shengyao Zhuang, Ping Nie, Kai Zou, Andrew Liu, Joshua Green, Kshama Patel, Ruoxi Meng, Mingyi Su, Sahel Sharifymoghaddam, Yanxi Li, Haoran Hong, Xinyu Shi, Xuye Liu, Nandan Thakur, Crystina Zhang, Luyu Gao, Wenhui Chen, and Jimmy Lin. *BrowseComp-Plus: A More Fair and Transparent Evaluation Benchmark of Deep-Research Agent*. 2025. URL: <https://arxiv.org/abs/2508.06600> (cited on pages 4, 110).
- [12] Manuel Faysse, Patrick Fernandes, Nuno M. Guerreiro, António Loison, Duarte M. Alves, Caio Corro, Nicolas Boizard, João Alves, Ricardo Rei, Pedro H. Martins, Antoni Bigata Casademunt, François Yvon, André F. T. Martins, Gautier Viaud, Céline Hudelot, and Pierre Colombo. *CroissantLLM: A Truly Bilingual French-English Language Model*. In: *Transactions on Machine Learning Research* (2025). doi: [10.48550/arXiv.2402.00786](https://doi.org/10.48550/arXiv.2402.00786) (cited on pages 4, 6, 7, 60, 61, 79, 107).
- [13] ViDoRe. *ColQwen2 Models*. Official collection of released ColQwen2 checkpoints. Hugging Face. 2024. URL: <https://huggingface.co/collections/vidore/colqwen2-models> (visited on 06/05/2026) (cited on page 5).
- [14] Manuel Faysse, Antonio Loison, Paul Teiletche, Quentin Macé, and Merve Noyan. *Introducing ColQwen-Omni: Retrieve in every modality*. Hugging Face Blog. July 2025. URL: <https://huggingface.co/blog/manu/colqwen-omni-omnimodal-retrieval> (cited on pages 5, 109).
- [15] Nils Reimers and Iryna Gurevych. *Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks*. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, 2019, pp. 3982–3992. doi: [10.18653/v1/D19-1410](https://doi.org/10.18653/v1/D19-1410) (cited on pages 5, 15, 24, 50, 54, 68, 117).
- [16] Gergely Szilvasy*, Manuel Faysse*, Maria Lomeli, Matthijs Douze, Pierre-Emmanuel Mazaré, Loïc Cabannes, Wen-tau Yih, and Hervé Jégou. *Self-Pruned Key-Value Attention: Learning When to Write by Predicting Future Utility*. 2026. URL: <https://arxiv.org/abs/2605.14037> (cited on pages 6, 7, 109).
- [17] Manuel Faysse, Gautier Viaud, Céline Hudelot, and Pierre Colombo. *Revisiting Instruction Fine-tuned Model Evaluation to Guide Industrial Applications*. In: (Oral, EMNLP 2023) *The 2023 Conference on Empirical Methods in Natural Language Processing*. 2023 (cited on pages 6, 7, 95).
- [18] Hippolyte Gisserot-Boukhlef, Manuel Faysse, Emmanuel Malherbe, Céline Hudelot, and Pierre Colombo. *Towards Trustworthy Reranking: A Simple yet Effective Abstention Mechanism*. In: *Transactions on Machine Learning Research* (2024) (cited on pages 6, 7, 34, 77, 99).
- [19] Kenneth Enevoldsen, Isaac Chung, Imene Kerboua, Márton Kardos, Ashwin Mathur, David Stap, Jay Gala, Wissam Siblini, Dominik Krzemiński, Genta Indra Winata, Saba Sturua, Saiteja Utpala, Mathieu Ciancone, Marion Schaeffer, Gabriel Sequeira, Diganta Misra, Shreeya Dhakal, Jonathan Rystrom, Roman Solomatin, Ömer Çağatan, Akash Kundu, Martin Bernstorff, Shitao

- Xiao, Akshita Sukhlecha, Bhavish Pahwa, Rafał Poświata, Kranthi Kiran GV, Shawon Ashraf, Daniel Auras, Björn Plüster, Jan Philipp Harries, Loic Magne, Isabelle Mohr, Mariya Hendriksen, Dawei Zhu, Hippolyte Gisserot-Boukhlef, Tom Aarsen, Jan Kostkan, Konrad Wojtasik, Taemin Lee, Marek Šuppa, Crystina Zhang, Roberta Rocca, Mohammed Hamdy, Andrianos Michail, John Yang, Manuel Faysse, et al. *MMTEB: Massive Multilingual Text Embedding Benchmark*. In: *International Conference on Learning Representations (ICLR 2025)*. 2025 (cited on pages 6, 8, 61).
- [20] Pedro Henrique Martins, Patrick Fernandes, João Alves, Nuno M. Guerreiro, Ricardo Rei, Duarte M. Alves, José Pombal, Amin Farajian, Manuel Faysse, Mateusz Klimaszewski, Pierre Colombo, Barry Haddow, José G. C. de Souza, Alexandra Birch, and André F. T. Martins. *EuroLLM: Multilingual Language Models for Europe*. 2024. URL: <https://arxiv.org/abs/2409.16235> (cited on pages 6, 8, 82, 102).
- [21] Pedro Henrique Martins, João Alves, Patrick Fernandes, Nuno M. Guerreiro, Ricardo Rei, Amin Farajian, Mateusz Klimaszewski, Duarte M. Alves, José Pombal, Manuel Faysse, et al. *EuroLLM-9B: Technical Report*. 2025. URL: <https://arxiv.org/abs/2506.04079> (cited on pages 6, 8).
- [22] Nicolas Boizard, Hippolyte Gisserot-Boukhlef, Duarte M. Alves, André Martins, Ayoub Hammal, Caio Corro, Céline Hudelot, Emmanuel Malherbe, Etienne Malaboeuf, Fanny Jourdan, Gabriel Hautreux, João Alves, Kevin El-Haddad, Manuel Faysse, et al. *EuroBERT: Scaling Multilingual Encoders for European Languages*. In: *Conference on Language Modeling (COLM 2025)*. 2025 (cited on pages 6, 8, 60, 66, 102).
- [23] Hippolyte Gisserot-Boukhlef, Nicolas Boizard, Manuel Faysse, Duarte M. Alves, Emmanuel Malherbe, André F. T. Martins, Céline Hudelot, and Pierre Colombo. *Should We Still Pretrain Encoders with Masked Language Modeling?* In: *International Conference on Learning Representations (ICLR 2026)*. 2026 (cited on pages 6, 7, 51, 52, 56, 60).
- [24] Matthieu Meeus, Igor Shilov, Manuel Faysse, and Yves-Alexandre de Montjoye. *Copyright Traps for Large Language Models*. In: *41st International Conference on Machine Learning (ICML 2024)*. 2024 (cited on pages 6, 8, 82, 107).
- [25] Matthieu Meeus, Igor Shilov, Shubham Jain, Manuel Faysse, Marek Rei, and Yves-Alexandre de Montjoye. *SoK: Membership Inference Attacks on LLMs are Rushing Nowhere (and How to Fix It)*. In: (Best Paper) *IEEE Conference on Secure and Trustworthy Machine Learning (SaTML 2025)*. 2025 (cited on pages 7, 8, 82).
- [26] Benjamin Clavié, Xianming Li, Antoine Chaffin, Omar Khattab, Tom Aarsen, Manuel Faysse, and Jing Li. *LIR: The First Workshop on Late Interaction and Multi Vector Retrieval @ ECIR 2026*. 2025. URL: <https://arxiv.org/abs/2511.00444> (cited on page 8).
- [27] Kyle Conrau-Lewis. *Medieval Indices: Unlikely Leisurely Reading*. Yale Program in the History of the Book. June 19, 2018. URL: <https://bookhistory.yale.edu/blog/medieval-indices-unlikely-leisurely-reading> (visited on 06/03/2026) (cited on page 11).

- [28] Melvil Dewey. *A Classification and Subject Index for Cataloguing and Arranging the Books and Pamphlets of a Library*. Amherst, Mass., 1876 (cited on page 11).
- [29] Calvin N. Mooers. *The Theory of Digital Handling of Non-Numerical Information and Its Implications to Machine Economics*. Zator Technical Bulletin 48. Boston, MA: Zator Company, 1950 (cited on page 12).
- [30] Philip Rutherford Bagley. 'Electronic Digital Machines for High-Speed Information Searching'. Master's thesis. Massachusetts Institute of Technology, 1951 (cited on page 12).
- [31] H. P. Luhn. *Auto-Encoding of Documents for Information Retrieval Systems*. In: *Modern Trends in Documentation: Proceedings of a Symposium Held at the University of Southern California, April 1958*. Ed. by Martha Boaz. New York: Pergamon Press, 1959, pp. 45–58 (cited on page 12).
- [32] M. E. Maron and J. L. Kuhns. *On Relevance, Probabilistic Indexing and Information Retrieval*. In: *Journal of the ACM* 7.3 (July 1960), pp. 216–244. doi: [10.1145/321033.321035](https://doi.org/10.1145/321033.321035) (cited on page 12).
- [33] Gerard Salton. *Automatic Information Organization and Retrieval*. New York: McGraw-Hill, 1968 (cited on page 12).
- [34] Gerard Salton, Anita Wong, and Chung-Shu Yang. *A Vector Space Model for Automatic Indexing*. In: *Communications of the ACM* 18 (1975), pp. 613–620. doi: [10.1145/361219.361220](https://doi.org/10.1145/361219.361220) (cited on page 12).
- [35] Karen Spärck Jones. *A Statistical Interpretation of Term Specificity and Its Application in Retrieval*. en. In: *Journal of Documentation* 28.1 (Jan. 1972), pp. 11–21. doi: [10.1108/eb026526](https://doi.org/10.1108/eb026526). (Visited on 06/12/2024) (cited on pages 12, 24, 116, 123).
- [36] Stephen E. Robertson and Karen Sparck Jones. *Relevance Weighting of Search Terms*. In: *Journal of the American Society for Information Science* 27.3 (1976), pp. 129–146. doi: [10.1002/asi.4630270302](https://doi.org/10.1002/asi.4630270302) (cited on page 13).
- [37] Stephen E. Robertson. *The Probability Ranking Principle in IR*. In: *Journal of Documentation* 33.4 (1977), pp. 294–304. doi: [10.1108/eb026647](https://doi.org/10.1108/eb026647) (cited on page 13).
- [38] Stephen Robertson and Hugo Zaragoza. *The Probabilistic Relevance Framework: BM25 and Beyond*. en. In: *Foundations and Trends® in Information Retrieval* 3.4 (2009), pp. 333–389. doi: [10.1561/15000000019](https://doi.org/10.1561/15000000019). (Visited on 05/19/2024) (cited on page 13).
- [39] Stephen E. Robertson and Steve Walker. *Some Simple Effective Approximations to the 2-Poisson Model for Probabilistic Weighted Retrieval*. In: *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. Springer-Verlag, 1994, pp. 232–241. doi: [10.1007/978-1-4471-2099-5_24](https://doi.org/10.1007/978-1-4471-2099-5_24) (cited on pages 13, 116).

- [40] Stephen E. Robertson, Steve Walker, Susan Jones, Micheline Hancock-Beaulieu, and Mike Gatford. *Okapi at TREC-3*. In: *Proceedings of The Third Text REtrieval Conference, TREC 1994, Gaithersburg, Maryland, USA, November 2-4, 1994*. Ed. by Donna K. Harman. Vol. 500-225. NIST Special Publication. National Institute of Standards and Technology (NIST), 1994, pp. 109–126 (cited on pages 13, 24, 68, 71, 116).
- [41] National Library of Medicine. *MEDLINE History*. Accessed: 2026-03-31. 2024. URL: https://www.nlm.nih.gov/medline/medline_history.html (cited on page 13).
- [42] Engineering and Technology History Wiki. *Milestones: DIALOG Online Search System, 1966*. Accessed: 2026-03-31. 2019. URL: https://ethw.org/Milestones:DIALOG-Online-Search-System,_1966 (cited on page 13).
- [43] Robert T. Dattola, ed. *ACM-SIGIR Conference on Information Storage and Retrieval*. ACM, 1978 (cited on page 13).
- [44] J. J. Rocchio. *Relevance Feedback in Information Retrieval*. In: *The SMART Retrieval System: Experiments in Automatic Document Processing*. Ed. by Gerard Salton. Englewood Cliffs, NJ: Prentice-Hall, 1971, pp. 313–323 (cited on page 13).
- [45] N. Jardine and C. J. van Rijsbergen. *The Use of Hierarchic Clustering in Information Retrieval*. In: *Information Storage and Retrieval 7.5 (1971)*, pp. 217–240. doi: [10.1016/0020-0271\(71\)90051-9](https://doi.org/10.1016/0020-0271(71)90051-9) (cited on page 13).
- [46] C. J. van Rijsbergen. *Information Retrieval*. 2nd ed. London: Butterworths, 1979 (cited on page 13).
- [47] Donna Harman. *Overview of the First Text REtrieval Conference*. In: *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 1993, pp. 36–47 (cited on page 13).
- [48] Sergey Brin and Lawrence Page. *The Anatomy of a Large-Scale Hypertextual Web Search Engine*. In: *Computer Networks and ISDN Systems 30.1-7 (1998)*, pp. 107–117. doi: [10.1016/S0169-7552\(98\)00110-X](https://doi.org/10.1016/S0169-7552(98)00110-X) (cited on page 14).
- [49] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. *The PageRank Citation Ranking: Bringing Order to the Web*. Tech. rep. Stanford Digital Library Technologies Project, Nov. 1998 (cited on page 14).
- [50] Christopher J. C. Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Gregory N. Hullender. *Learning to Rank Using Gradient Descent*. In: *Proceedings of the 22nd International Conference on Machine Learning*. ACM, 2005, pp. 89–96. doi: [10.1145/1102351.1102363](https://doi.org/10.1145/1102351.1102363) (cited on page 14).
- [51] Christopher J. C. Burges, Robert Ragno, and Quoc Viet Le. *Learning to Rank with Nonsmooth Cost Functions*. In: *Advances in Neural Information Processing Systems 19*. MIT Press, 2006, pp. 193–200 (cited on page 14).
- [52] Chris J. C. Burges. *From RankNet to LambdaRank to LambdaMART: An Overview*. Tech. rep. MSR-TR-2010-82. Microsoft Research, 2010 (cited on page 14).

- [53] Thorsten Joachims. *Optimizing Search Engines Using Clickthrough Data*. In: *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '02. New York, NY, USA: Association for Computing Machinery, 2002, pp. 133–142. doi: [10.1145/775047.775067](https://doi.org/10.1145/775047.775067) (cited on page 14).
- [54] Thorsten Joachims, Laura A. Granka, Bing Pan, Helene Hembrooke, and Geri Gay. *Accurately Interpreting Clickthrough Data as Implicit Feedback*. In: *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2005, pp. 154–161. doi: [10.1145/1076034.1076063](https://doi.org/10.1145/1076034.1076063) (cited on page 14).
- [55] Filip Radlinski and Thorsten Joachims. *Query Chains: Learning to Rank from Implicit Feedback*. In: *Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2005, pp. 239–248. doi: [10.1145/1081870.1081899](https://doi.org/10.1145/1081870.1081899) (cited on page 14).
- [56] Bing Search Blog. *Understand Your World with Bing*. Accessed: 2026-03-31. 2013. URL: <https://blogs.bing.com/search/March-2013/Understand-Your-World-with-Bing> (cited on page 14).
- [57] Google Search Central. *A Guide to Google Search Ranking Systems*. Accessed: 2026-03-31. 2025. URL: <https://developers.google.com/search/docs/appearance/ranking-systems-guide> (cited on page 14).
- [58] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry P. Heck. *Learning Deep Structured Semantic Models for Web Search using Clickthrough Data*. In: *Proceedings of the 22nd ACM International Conference on Information and Knowledge Management*. CIKM '13. New York, NY, USA: Association for Computing Machinery, 2013, pp. 2333–2338. doi: [10.1145/2505515.2505665](https://doi.org/10.1145/2505515.2505665) (cited on page 14).
- [59] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, 2019, pp. 4171–4186. doi: [10.18653/v1/N19-1423](https://doi.org/10.18653/v1/N19-1423) (cited on pages 14, 133).
- [60] Pandu Nayak. *Understanding searches better than ever before*. Google Search blog. Accessed: 2026-03-31. Oct. 2019. URL: <https://blog.google/products-and-platforms/products/search/search-language-understanding-bert/> (cited on page 14).
- [61] Rodrigo Nogueira and Kyunghyun Cho. *Passage Re-ranking with BERT*. Version Number: 5. 2019. doi: [10.48550/ARXIV.1901.04085](https://doi.org/10.48550/ARXIV.1901.04085). URL: <https://arxiv.org/abs/1901.04085> (visited on 06/07/2024) (cited on pages 15, 16).
- [62] Andrew Yates, Rodrigo Nogueira, and Jimmy Lin. *Pretrained Transformers for Text Ranking: BERT and Beyond*. In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Tutorials*.

- Online: Association for Computational Linguistics, 2021, pp. 1–4. doi: [10.18653/v1/2021.naacl-tutorials.1](https://doi.org/10.18653/v1/2021.naacl-tutorials.1) (cited on page 15).
- [63] Jeff Johnson, Matthijs Douze, and Hervé Jégou. *Billion-scale Similarity Search with GPUs*. In: *CoRR abs/1702.08734* (2017) (cited on pages 15, 116).
 - [64] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. *Dense Passage Retrieval for Open-Domain Question Answering*. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, 2020, pp. 6769–6781. doi: [10.18653/v1/2020.emnlp-main.550](https://doi.org/10.18653/v1/2020.emnlp-main.550) (cited on pages 15, 24, 50, 72).
 - [65] Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. *BEIR: A Heterogenous Benchmark for Zero-shot Evaluation of Information Retrieval Models*. Version Number: 4. 2021. doi: [10.48550/ARXIV.2104.08663](https://doi.org/10.48550/ARXIV.2104.08663). URL: <https://arxiv.org/abs/2104.08663> (visited on 06/02/2024) (cited on pages 15, 22, 24, 67, 70, 80).
 - [66] Niklas Muennighoff, Nouamane Tazi, Loic Magne, and Nils Reimers. *MTEB: Massive Text Embedding Benchmark*. Version Number: 3. 2022. doi: [10.48550/ARXIV.2210.07316](https://doi.org/10.48550/ARXIV.2210.07316). URL: <https://arxiv.org/abs/2210.07316> (visited on 06/02/2024) (cited on pages 15, 22, 24, 25, 54, 67, 80).
 - [67] John Morris, Volodymyr Kuleshov, Vitaly Shmatikov, and Alexander Rush. *Text Embeddings Reveal (Almost) As Much As Text*. In: *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Ed. by Houda Bouamor, Juan Pino, and Kalika Bali. Singapore: Association for Computational Linguistics, Dec. 2023, pp. 12448–12460. doi: [10.18653/v1/2023.emnlp-main.765](https://doi.org/10.18653/v1/2023.emnlp-main.765) (cited on page 15).
 - [68] Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. *Towards General Text Embeddings with Multi-stage Contrastive Learning*. 2023. URL: <https://arxiv.org/abs/2308.03281> (cited on pages 15, 54, 72).
 - [69] Gabriel de Souza P. Moreira, Radek Osmulski, Mengyao Xu, Ronay Ak, Benedikt Schifferer, and Even Oldridge. *NV-Retriever: Improving Text Embedding Models with Effective Hard-Negative Mining*. 2024. doi: [10.48550/arXiv.2407.15831](https://doi.org/10.48550/arXiv.2407.15831) (cited on page 15).
 - [70] Rulin Shao, Rui Qiao, Varsha Kishore, Niklas Muennighoff, Xi Victoria Lin, Daniela Rus, Bryan Kian Hsiang Low, Sewon Min, Wen-tau Yih, Pang Wei Koh, and Luke Zettlemoyer. *ReasonIR: Training Retrievers for Reasoning Tasks*. 2025. doi: [10.48550/arXiv.2504.20595](https://doi.org/10.48550/arXiv.2504.20595) (cited on page 15).
 - [71] Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. *Improving Text Embeddings with Large Language Models*. 2024. URL: <https://arxiv.org/abs/2401.00368> (cited on pages 16, 33, 71).

- [72] Parishad BehnamGhader, Vaibhav Adlakha, Marius Mosbach, Dzmitry Bahdanau, Nicolas Chapados, and Siva Reddy. *LLM2Vec: Large Language Models Are Secretly Powerful Text Encoders*. Version Number: 1. 2024. doi: [10.48550/ARXIV.2404.05961](https://arxiv.org/abs/2404.05961). URL: <https://arxiv.org/abs/2404.05961> (visited on 05/18/2024) (cited on page 16).
- [73] Chankyu Lee, Rajarshi Roy, Mengyao Xu, Jonathan Raiman, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. *NV-Embed: Improved Techniques for Training LLMs as Generalist Embedding Models*. 2024. URL: <https://arxiv.org/abs/2405.17428> (cited on pages 16, 33, 51, 71).
- [74] Orion Weller, Benjamin Van Durme, Dawn Lawrie, Ashwin Paranjape, Yuhao Zhang, and Jack Hessel. *Promptriever: Instruction-Trained Retrievers Can Be Prompted Like Language Models*. In: *International Conference on Learning Representations*. 2025 (cited on page 16).
- [75] Orion Weller, Michael Boratko, Iftekhar Naim, and Jinhyuk Lee. *On the Theoretical Limitations of Embedding-Based Retrieval*. In: *International Conference on Learning Representations*. 2026 (cited on page 16).
- [76] Weiwei Sun, Lingyong Yan, Xinyu Ma, Shuaiqiang Wang, Pengjie Ren, Zhumin Chen, Dawei Yin, and Zhaochun Ren. *Is ChatGPT Good at Search? Investigating Large Language Models as Re-Ranking Agents*. In: *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Singapore: Association for Computational Linguistics, 2023, pp. 14918–14937. doi: [10.18653/v1/2023.emnlp-main.923](https://arxiv.org/abs/2305.18270) (cited on page 16).
- [77] Orion Weller, Kathryn Ricci, Eugene Yang, Andrew Yates, Dawn Lawrie, and Benjamin Van Durme. *Rank1: Test-Time Compute for Reranking in Information Retrieval*. In: *Conference on Language Modeling*. 2025 (cited on page 16).
- [78] Omar Khattab and Matei Zaharia. *ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT*. In: *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '20. New York, NY, USA: Association for Computing Machinery, 2020, pp. 39–48. doi: [10.1145/3397271.3401075](https://arxiv.org/abs/1910.17117) (cited on pages 16, 23, 24, 28, 29, 56, 60, 71, 72, 108, 117, 133, 164).
- [79] Julian Killingback, Hansi Zeng, and Hamed Zamani. *Hypencoder: Hypernetworks for Information Retrieval*. 2025. URL: <https://arxiv.org/abs/2502.05364> (cited on pages 17, 108).
- [80] Yi Tay, Vinh Q. Tran, Mostafa Dehghani, Jianmo Ni, Dara Bahri, Harsh Mehta, Zhen Qin, Kai Hui, Zhe Zhao, Jai Gupta, Tal Schuster, William W. Cohen, and Donald Metzler. *Transformer Memory as a Differentiable Search Index*. 2022. URL: <https://arxiv.org/abs/2202.06991> (cited on pages 17, 108).
- [81] Xiaoxi Li, Jiajie Jin, Yujia Zhou, Yuyao Zhang, Peitian Zhang, Yutao Zhu, and Zhicheng Dou. *From Matching to Generation: A Survey on Generative Information Retrieval*. 2024. doi: [10.48550/arXiv.2404.14851](https://arxiv.org/abs/2404.14851). URL: <https://arxiv.org/abs/2404.14851> (cited on page 17).

- [82] Thibault Formal, Benjamin Piwowarski, and Stéphane Clinchant. *SPLADE: Sparse Lexical and Expansion Model for First Stage Ranking*. In: *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '21. Virtual Event, Canada: Association for Computing Machinery, 2021, pp. 2288–2292. DOI: [10.1145/3404835.3463098](https://doi.org/10.1145/3404835.3463098) (cited on page 17).
- [83] Thibault Formal, Maxime Louis, Hervé Dejean, and Stéphane Clinchant. *Learning Retrieval Models with Sparse Autoencoders*. 2026. URL: <https://arxiv.org/abs/2603.13277> (cited on page 17).
- [84] Benjamin Clavié, Sean Lee, Aamir Shakir, and Makoto P. Kato. *Latent Terms: Dense Retrievers Contain Trivially Extractable BM25-ready Zipfian Vocabularies*. 2026. URL: <https://arxiv.org/abs/2605.29384> (cited on page 17).
- [85] Michael Günther, Isabelle Mohr, Daniel James Williams, Bo Wang, and Han Xiao. *Late Chunking: Contextual Chunk Embeddings Using Long-Context Embedding Models*. 2024. URL: <https://arxiv.org/abs/2409.04701> (cited on pages 17, 67, 71, 118).
- [86] John X. Morris and Alexander M. Rush. *Contextual Document Embeddings*. 2024. URL: <https://arxiv.org/abs/2410.02525> (cited on pages 17, 68, 71, 174).
- [87] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. *Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks*. Version Number: 4. 2020. DOI: [10.48550/ARXIV.2005.11401](https://doi.org/10.48550/ARXIV.2005.11401). URL: <https://arxiv.org/abs/2005.11401> (visited on 06/14/2024) (cited on pages 17, 36, 50, 66, 120).
- [88] Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. *WebGPT: Browser-assisted Question-Answering with Human Feedback*. In: *arXiv preprint arXiv:2112.09332* (2021) (cited on page 18).
- [89] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. *ReAct: Synergizing Reasoning and Acting in Language Models*. In: *International Conference on Learning Representations*. 2023 (cited on page 18).
- [90] Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. *Interleaving Retrieval with Chain-of-Thought Reasoning for Knowledge-Intensive Multi-Step Questions*. In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*. 2023, pp. 10014–10037 (cited on pages 18, 110).
- [91] Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. *Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection*. In: *International Conference on Learning Representations*. 2024 (cited on page 18).

- [92] Zhuofeng Li, Haoxiang Zhang, Cong Wei, Pan Lu, Ping Nie, Yi Lu, Yuyang Bai, Shangbin Feng, Hangxiao Zhu, Ming Zhong, Yuyu Zhang, Jianwen Xie, Yejin Choi, James Zou, Jiawei Han, Wenhui Chen, Jimmy Lin, Dongfu Jiang, and Yu Zhang. *Beyond Semantic Similarity: Rethinking Retrieval for Agentic Search via Direct Corpus Interaction*. 2026. DOI: [10.48550/arXiv.2605.05242](https://arxiv.org/abs/2605.05242). URL: <https://arxiv.org/abs/2605.05242> (cited on page 18).
- [93] Zijian Chen, Xueguang Ma, Shengyao Zhuang, Jimmy Lin, Akari Asai, and Victor Zhong. *AgentIR: Reasoning-Aware Retrieval for Deep Research Agents*. 2026. DOI: [10.48550/arXiv.2603.04384](https://arxiv.org/abs/2603.04384) (cited on page 18).
- [94] Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. *MS MARCO: A Human Generated MACHine Reading COMprehension Dataset*. Version Number: 3. 2016. DOI: [10.48550/ARXIV.1611.09268](https://arxiv.org/abs/1611.09268). URL: <https://arxiv.org/abs/1611.09268> (visited on 06/02/2024) (cited on page 22).
- [95] Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. *Text Embeddings by Weakly-Supervised Contrastive Pre-training*. Version Number: 2. 2022. DOI: [10.48550/ARXIV.2212.03533](https://arxiv.org/abs/2212.03533). URL: <https://arxiv.org/abs/2212.03533> (visited on 06/04/2024) (cited on pages 24, 50, 72).
- [96] Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. *MiniLM: Deep Self-Attention Distillation for Task-Agnostic Compression of Pre-Trained Transformers*. arXiv:2002.10957 [cs]. Apr. 2020. URL: <http://arxiv.org/abs/2002.10957> (visited on 06/07/2024) (cited on page 24).
- [97] Cohere. *Introducing Rerank 3: A New Foundation Model for Efficient Enterprise Search & Retrieval*. Apr. 2024. URL: <https://cohere.com/blog/rerank-3> (cited on pages 24, 51).
- [98] Li Zhang, Shuo Zhang, and Krisztian Balog. *Table2Vec: Neural Word and Entity Embeddings for Table Population and Retrieval*. In: *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '19. ACM, July 2019. DOI: [10.1145/3331184.3331333](https://doi.org/10.1145/3331184.3331333) (cited on page 24).
- [99] Averi Nowak, Francesco Piccinno, and Yasemin Altun. *Multi-modal Chart Retrieval: A Comparison of Text, Table and Image Based Approaches*. In: *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*. Ed. by Kevin Duh, Helena Gomez, and Steven Bethard. Mexico City, Mexico: Association for Computational Linguistics, June 2024, pp. 5488–5505. DOI: [10.18653/v1/2024.naacl-long.307](https://doi.org/10.18653/v1/2024.naacl-long.307) (cited on page 24).
- [100] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. *Learning Transferable Visual Models From Natural Language Supervision*. In: (2021). Publisher: arXiv Version Number: 1. DOI: [10.48550/ARXIV.2103.00020](https://arxiv.org/abs/2103.00020). (Visited on 08/18/2023) (cited on pages 25, 53, 54, 119, 132).

- [101] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. *Sigmoid Loss for Language Image Pre-Training*. In: (2023). Publisher: [object Object] Version Number: 4. doi: [10.48550/ARXIV.2303.15343](https://arxiv.org/abs/2303.15343). (Visited on 05/08/2024) (cited on pages 25, 132).
- [102] Lewei Yao, Runhui Huang, Lu Hou, Guansong Lu, Minzhe Niu, Hang Xu, Xiaodan Liang, Zhenguo Li, Xin Jiang, and Chunjing Xu. *FILIP: Fine-grained Interactive Language-Image Pre-Training*. Version Number: 1. 2021. doi: [10.48550/ARXIV.2111.07783](https://arxiv.org/abs/2111.07783). URL: <https://arxiv.org/abs/2111.07783> (visited on 05/21/2024) (cited on page 25).
- [103] Srikar Appalaraju, Bhavan Jasani, Bhargava Urala Kota, Yusheng Xie, and R. Manmatha. *DocFormer: End-to-End Transformer for Document Understanding*. Version Number: 2. 2021. doi: [10.48550/ARXIV.2106.11539](https://arxiv.org/abs/2106.11539). URL: <https://arxiv.org/abs/2106.11539> (visited on 06/12/2024) (cited on page 25).
- [104] Geewook Kim, Teakgyu Hong, Moonbin Yim, Jeongyeon Nam, Jinyoung Park, Jinyeong Yim, Wonseok Hwang, Sangdoo Yun, Dongyoon Han, and Seunghyun Park. *OCR-free Document Understanding Transformer*. Version Number: 5. 2021. doi: [10.48550/ARXIV.2111.15664](https://arxiv.org/abs/2111.15664). URL: <https://arxiv.org/abs/2111.15664> (visited on 06/12/2024) (cited on page 25).
- [105] Yupan Huang, Tengchao Lv, Lei Cui, Yutong Lu, and Furu Wei. *LayoutLMv3: Pre-training for Document AI with Unified Text and Image Masking*. In: (2022). Publisher: arXiv Version Number: 3. doi: [10.48550/ARXIV.2204.08387](https://arxiv.org/abs/2204.08387). (Visited on 10/25/2023) (cited on page 25).
- [106] Zineng Tang, Ziyi Yang, Guoxin Wang, Yuwei Fang, Yang Liu, Chenguang Zhu, Michael Zeng, Cha Zhang, and Mohit Bansal. *Unifying Vision, Text, and Layout for Universal Document Processing*. Version Number: 3. 2022. doi: [10.48550/ARXIV.2212.02623](https://arxiv.org/abs/2212.02623). URL: <https://arxiv.org/abs/2212.02623> (visited on 06/12/2024) (cited on page 25).
- [107] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. In: (2020). Publisher: arXiv Version Number: 2. doi: [10.48550/ARXIV.2010.11929](https://arxiv.org/abs/2010.11929). (Visited on 10/28/2023) (cited on pages 25, 52, 119, 132).
- [108] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andrew Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. *Flamingo: a Visual Language Model for Few-Shot Learning*. In: (2022). Publisher: arXiv Version Number: 2. doi: [10.48550/ARXIV.2204.14198](https://arxiv.org/abs/2204.14198). (Visited on 09/09/2023) (cited on pages 25, 51–53, 60).

- [109] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. *Visual Instruction Tuning*. In: (2023). Publisher: arXiv Version Number: 1. doi: [10.48550/ARXIV.2304.08485](https://arxiv.org/abs/2304.08485). (Visited on 10/27/2023) (cited on pages 25, 53).
- [110] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. *Qwen-VL: A Versatile Vision-Language Model for Understanding, Localization, Text Reading, and Beyond*. In: (2023). Publisher: arXiv Version Number: 3. doi: [10.48550/ARXIV.2308.12966](https://arxiv.org/abs/2308.12966). (Visited on 02/08/2024) (cited on pages 25, 60, 82).
- [111] Hugo Laurençon, Léo Tronchon, Matthieu Cord, and Victor Sanh. *What matters when building vision-language models?* arXiv:2405.02246 [cs]. May 2024. URL: <http://arxiv.org/abs/2405.02246> (visited on 05/06/2024) (cited on pages 25, 32, 53, 159, 160).
- [112] Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, Thomas Unterthiner, Daniel Keysers, Skanda Koppula, Fangyu Liu, Adam Grycner, Alexey Gritsenko, Neil Houlsby, Manoj Kumar, Keran Rong, Julian Eisenschlos, Rishabh Kabra, Matthias Bauer, Matko Bošnjak, Xi Chen, Matthias Minderer, Paul Voigtlaender, Ioana Bica, Ivana Balazevic, Joan Puigcerver, Pinelopi Papalampidi, Olivier Henaff, Xi Xiong, Radu Soricut, Jeremiah Harmsen, and Xiaohua Zhai. *PaliGemma: A versatile 3B VLM for transfer*. 2024. URL: <https://arxiv.org/abs/2407.07726> (cited on pages 25, 56, 57, 161).
- [113] Xi Chen, Xiao Wang, Lucas Beyer, Alexander Kolesnikov, Jialin Wu, Paul Voigtlaender, Basil Mustafa, Sebastian Goodman, Ibrahim Alabdulmohsin, Piotr Padlewski, Daniel Salz, Xi Xiong, Daniel Vlasic, Filip Pavetic, Keran Rong, Tianli Yu, Daniel Keysers, Xiaohua Zhai, and Radu Soricut. *PaLI-3 Vision Language Models: Smaller, Faster, Stronger*. Version Number: 2. 2023. doi: [10.48550/ARXIV.2310.09199](https://arxiv.org/abs/2310.09199). URL: <https://arxiv.org/abs/2310.09199> (visited on 05/19/2024) (cited on pages 25, 30).
- [114] Ibrahim Alabdulmohsin, Xiaohua Zhai, Alexander Kolesnikov, and Lucas Beyer. *Getting ViT in Shape: Scaling Laws for Compute-Optimal Model Design*. In: (2023). Publisher: arXiv Version Number: 5. doi: [10.48550/ARXIV.2305.13035](https://arxiv.org/abs/2305.13035). (Visited on 05/19/2024) (cited on pages 25, 27).
- [115] Gemma Team et al. *Gemma: Open Models Based on Gemini Research and Technology*. Version Number: 4. 2024. doi: [10.48550/ARXIV.2403.08295](https://arxiv.org/abs/2403.08295). URL: <https://arxiv.org/abs/2403.08295> (visited on 05/18/2024) (cited on page 25).
- [116] Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, Cong Wei, Botao Yu, Ruibin Yuan, Renliang Sun, Ming Yin, Boyuan Zheng, Zhenzhu Yang, Yibo Liu, Wenhao Huang, Huan Sun, Yu Su, and Wenhua Chen. *MMMU: A Massive Multi-discipline Multimodal Understanding and Reasoning Benchmark for Expert AGI*. Version Number: 3. 2023. doi: [10.48550/ARXIV.2311.16502](https://arxiv.org/abs/2311.16502). URL: <https://arxiv.org/abs/2311.16502> (visited on 06/04/2024) (cited on page 25).

- [117] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. *Microsoft COCO: Common Objects in Context*. Version Number: 3. 2014. doi: [10.48550/ARXIV.1405.0312](https://arxiv.org/abs/1405.0312). URL: <https://arxiv.org/abs/1405.0312> (visited on 06/14/2024) (cited on pages 25, 54, 161).
- [118] Łukasz Borchmann, Michał Pietruszka, Tomasz Stanislawek, Dawid Jurkiewicz, Michał Turski, Karolina Szyndler, and Filip Galiński. *DUE: End-to-End Document Understanding Benchmark*. In: *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*. 2021 (cited on page 25).
- [119] Ashish V. Thapliyal, Jordi Pont-Tuset, Xi Chen, and Radu Soricut. *Crossmodal-3600: A Massively Multilingual Multimodal Evaluation Dataset*. Version Number: 2. 2022. doi: [10.48550/ARXIV.2205.12522](https://arxiv.org/abs/2205.12522). URL: <https://arxiv.org/abs/2205.12522> (visited on 06/15/2024) (cited on page 25).
- [120] Minesh Mathew, Dimosthenis Karatzas, and C. V. Jawahar. *DocVQA: A Dataset for VQA on Document Images*. In: (2020). doi: [10.48550/ARXIV.2007.00398](https://arxiv.org/abs/2007.00398). (Visited on 05/17/2024) (cited on pages 26, 37, 38, 125).
- [121] Minesh Mathew, Viraj Bagal, Rubèn Pérez Tito, Dimosthenis Karatzas, Ernest Valveny, and C. V. Jawahar. *InfographicVQA*. Version Number: 2. 2021. doi: [10.48550/ARXIV.2104.12756](https://arxiv.org/abs/2104.12756). URL: <https://arxiv.org/abs/2104.12756> (visited on 06/02/2024) (cited on pages 26, 38, 125).
- [122] Lei Li, Yuqi Wang, Runxin Xu, Peiyi Wang, Xiachong Feng, Lingpeng Kong, and Qi Liu. *Multimodal ArXiv: A Dataset for Improving Scientific Comprehension of Large Vision-Language Models*. 2024 (cited on pages 26, 125).
- [123] Fengbin Zhu, Wenqiang Lei, Fuli Feng, Chao Wang, Haozhou Zhang, and Tat-Seng Chua. *Towards Complex Document Understanding By Discrete Reasoning*. In: (2022). Publisher: arXiv Version Number: 3. doi: [10.48550/ARXIV.2207.11871](https://arxiv.org/abs/2207.11871). (Visited on 06/02/2024) (cited on pages 26, 38, 125).
- [124] Anthropic. *The Claude 3 Model Family: Opus, Sonnet, Haiku*. 2024. URL: https://www-cdn.anthropic.com/de8ba9b01c9ab7cbabf5c33b80b7bbc618857627/Model_Card_Claude_3.pdf (cited on pages 26, 27).
- [125] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. *YOLOX: Exceeding YOLO Series in 2021*. Version Number: 2. 2021. doi: [10.48550/ARXIV.2107.08430](https://arxiv.org/abs/2107.08430). URL: <https://arxiv.org/abs/2107.08430> (visited on 06/04/2024) (cited on page 27).
- [126] R. Smith. *An Overview of the Tesseract OCR Engine*. In: *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007) Vol 2*. ISSN: 1520-5363. Curitiba, Parana, Brazil: IEEE, Sept. 2007, pp. 629–633. doi: [10.1109/ICDAR.2007.4376991](https://arxiv.org/abs/2007.43769). (Visited on 06/02/2024) (cited on page 27).

- [127] Ruochen Zhao, Hailin Chen, Weishi Wang, Fangkai Jiao, Xuan Long Do, Chengwei Qin, Bosheng Ding, Xiaobao Guo, Minzhi Li, Xingxuan Li, and Shafiq Joty. *Retrieving Multimodal Information for Augmented Generation: A Survey*. Version Number: 3. 2023. doi: [10.48550/ARXIV.2303.10868](https://arxiv.org/abs/2303.10868). URL: <https://arxiv.org/abs/2303.10868> (visited on 06/02/2024) (cited on page 27).
- [128] Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. *BGE M3-Embedding: Multi-Lingual, Multi-Functionality, Multi-Granularity Text Embeddings Through Self-Knowledge Distillation*. Version Number: 3. 2024. doi: [10.48550/ARXIV.2402.03216](https://arxiv.org/abs/2402.03216). URL: <https://arxiv.org/abs/2402.03216> (visited on 06/04/2024) (cited on pages 27, 61, 70, 72, 141, 162).
- [129] Andreas Koukounas, Georgios Mastrapas, Michael Günther, Bo Wang, Scott Martens, Isabelle Mohr, Saba Sturua, Mohammad Kalim Akram, Joan Fontanals Martínez, Saahil Ognawala, Susana Guzman, Maximilian Werk, Nan Wang, and Han Xiao. *Jina CLIP: Your CLIP Model Is Also Your Text Retriever*. Version Number: 1. 2024. doi: [10.48550/ARXIV.2405.20204](https://arxiv.org/abs/2405.20204). URL: <https://arxiv.org/abs/2405.20204> (visited on 06/01/2024) (cited on pages 27, 59, 162).
- [130] Nomic. *Nomic Embed Vision: Expanding The Nomic Latent Space*. June 2024. URL: <https://blog.nomic.ai/posts/nomic-embed-vision> (cited on page 27).
- [131] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. *LoRA: Low-Rank Adaptation of Large Language Models*. In: (2021). Publisher: arXiv Version Number: 2. doi: [10.48550/ARXIV.2106.09685](https://arxiv.org/abs/2106.09685). (Visited on 08/30/2023) (cited on pages 29, 53).
- [132] Keshav Santhanam, Omar Khattab, Christopher Potts, and Matei Zaharia. *PLAID: An Efficient Engine for Late Interaction Retrieval*. Version Number: 1. 2022. doi: [10.48550/ARXIV.2205.09707](https://arxiv.org/abs/2205.09707). URL: <https://arxiv.org/abs/2205.09707> (visited on 06/14/2024) (cited on pages 31, 128).
- [133] Jinhyuk Lee, Zhuyun Dai, Sai Meher Karthik Duddu, Tao Lei, Iftekhhar Naim, Ming-Wei Chang, and Vincent Y. Zhao. *Rethinking the Role of Token Retrieval in Multi-Vector Retrieval*. Version Number: 3. 2023. doi: [10.48550/ARXIV.2304.01982](https://arxiv.org/abs/2304.01982). URL: <https://arxiv.org/abs/2304.01982> (visited on 06/14/2024) (cited on page 31).
- [134] Tri Dao. *FlashAttention-2: Faster Attention with Better Parallelism and Work Partitioning*. 2023. URL: <https://arxiv.org/abs/2307.08691> (cited on pages 31, 159).
- [135] Benjamin Clavié, Antoine Chaffin, and Griffin Adams. *Reducing the Footprint of Multi-Vector Retrieval with Minimal Performance Impact via Token Pooling*. Version Number: 1. 2024. doi: [10.48550/ARXIV.2409.14683](https://arxiv.org/abs/2409.14683). URL: <https://arxiv.org/abs/2409.14683> (visited on 09/25/2024) (cited on page 31).

- [136] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, L  lio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timoth  e Lacroix, and William El Sayed. *Mistral 7B*. In: (2023). Publisher: arXiv Version Number: 1. DOI: [10.48550/ARXIV.2310.06825](https://arxiv.org/abs/2310.06825). (Visited on 10/11/2023) (cited on pages 32, 80, 87, 93).
- [137] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. *Qwen2-VL: Enhancing Vision-Language Model's Perception of the World at Any Resolution*. 2024. URL: <https://arxiv.org/abs/2409.12191> (cited on pages 33, 52, 56, 161).
- [138] Hugo Lauren  on, Andr  s Marafioti, Victor Sanh, and L  o Tronchon. *Building and better understanding vision-language models: insights and future directions*. 2024 (cited on pages 34, 53, 159, 160).
- [139] Yubo Ma, Yuhang Zang, Liangyu Chen, Meiqi Chen, Yizhu Jiao, Xinze Li, Xinyuan Lu, Ziyu Liu, Yan Ma, Xiaoyi Dong, Pan Zhang, Liangming Pan, Yu-Gang Jiang, Jiaqi Wang, Yixin Cao, and Aixin Sun. *MMLongBench-Doc: Benchmarking Long-context Document Understanding with Visualizations*. 2024. URL: <https://arxiv.org/abs/2407.01523> (cited on page 34).
- [140] Ziyang Jiang, Rui Meng, Xinyi Yang, Semih Yavuz, Yingbo Zhou, and Wenhua Chen. *VLM2Vec: Training Vision-Language Models for Massive Multimodal Embedding Tasks*. 2024. URL: <https://arxiv.org/abs/2410.05160> (cited on pages 34, 51, 58–60, 161).
- [141] Nandan Thakur, Jimmy Lin, Sam Havens, Michael Carbin, Omar Khattab, and Andrew Drozdov. *FreshStack: Building Realistic Benchmarks for Evaluating Retrieval on Technical Documents*. 2025. URL: <https://arxiv.org/abs/2504.13128> (cited on pages 35, 37, 39, 67, 70).
- [142] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. *Retrieval-Augmented Generation for Large Language Models: A Survey*. Version Number: 5. 2023. DOI: [10.48550/ARXIV.2312.10997](https://arxiv.org/abs/2312.10997). URL: <https://arxiv.org/abs/2312.10997> (visited on 06/04/2024) (cited on page 36).
- [143] Wenqi Fan, Yujuan Ding, Liangbo Ning, Shijie Wang, Hengyun Li, Dawei Yin, Tat-Seng Chua, and Qing Li. *A Survey on RAG Meeting LLMs: Towards Retrieval-Augmented Large Language Models*. 2024. URL: <https://arxiv.org/abs/2405.06211> (cited on page 36).
- [144] Mohammad Mahdi Abootorabi, Amirhosein Zobeiri, Mahdi Dehghani, Mohammadali Mohammadkhani, Bardia Mohammadi, Omid Ghahroodi, Mahdieh Soleymani Baghshah, and Ehsaneddin Asgari. *Ask in Any Modality: A Comprehensive Survey on Multimodal Retrieval-Augmented Generation*. 2025. URL: <https://arxiv.org/abs/2502.08826> (cited on page 37).

- [145] Jaemin Cho, Debanjan Mahata, Ozan Irsoy, Yujie He, and Mohit Bansal. *M3DocRAG: Multi-modal Retrieval is What You Need for Multi-page Multi-document Understanding*. 2024. URL: <https://arxiv.org/abs/2411.04952> (cited on pages 37, 39).
- [146] Yixuan Tang and Yi Yang. *MultiHop-RAG: Benchmarking Retrieval-Augmented Generation for Multi-Hop Queries*. 2024. URL: <https://arxiv.org/abs/2401.15391> (cited on pages 37, 39).
- [147] Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. *Enabling Large Language Models to Generate Text with Citations*. 2023. URL: <https://arxiv.org/abs/2305.14627> (cited on page 37).
- [148] Xueguang Ma, Shengyao Zhuang, Bevan Koopman, Guido Zuccon, Wenhui Chen, and Jimmy Lin. *VISA: Retrieval Augmented Generation with Visual Source Attribution*. 2024. URL: <https://arxiv.org/abs/2412.14457> (cited on page 37).
- [149] Michael Günther, Saba Sturua, Mohammad Kalim Akram, Isabelle Mohr, Andrei Ungureanu, Bo Wang, Sedigheh Eslami, Scott Martens, Maximilian Werk, Nan Wang, and Han Xiao. *jina-embeddings-v4: Universal Embeddings for Multimodal Multilingual Retrieval*. 2025. URL: <https://arxiv.org/abs/2506.18902> (cited on pages 37, 38, 43, 59–61, 141, 161).
- [150] Xiangyu Peng, Can Qin, Zeyuan Chen, Ran Xu, Caiming Xiong, and Chien-Sheng Wu. *UNIDOC-BENCH: A Unified Benchmark for Document-Centric Multimodal RAG*. 2025. URL: <https://arxiv.org/abs/2510.03663> (cited on pages 37, 39).
- [151] Zirui Wang, Mengzhou Xia, Luxi He, Howard Chen, Yitao Liu, Richard Zhu, Kaiqu Liang, Xindi Wu, Haotian Liu, Sadhika Malladi, et al. *Charxiv: Charting gaps in realistic chart understanding in multimodal llms*. In: *Advances in Neural Information Processing Systems* 37 (2024), pp. 113569–113697 (cited on page 38).
- [152] Wenhan Yu, Wang Chen, Guanqiang Qi, Weikang Li, Yang Li, Lei Sha, Deguo Xia, and Jizhou Huang. *BBox DocVQA: A Large Scale Bounding Box Grounded Dataset for Enhancing Reasoning in Document Visual Question Answer*. 2025. URL: <https://arxiv.org/abs/2511.15090> (cited on page 38).
- [153] Xueguang Ma, Sheng-Chieh Lin, Minghan Li, Wenhui Chen, and Jimmy Lin. *Unifying Multimodal Retrieval via Document Screenshot Embedding*. 2024. URL: <https://arxiv.org/abs/2406.11251> (cited on pages 38, 51, 58, 60, 77, 109).
- [154] Shi Yu, Chaoyue Tang, Bokai Xu, Junbo Cui, Junhao Ran, Yukun Yan, Zhenghao Liu, Shuo Wang, Xu Han, Zhiyuan Liu, and Maosong Sun. *VisRAG: Vision-based Retrieval-augmented Generation on Multi-modality Documents*. 2024. URL: <https://arxiv.org/abs/2410.10594> (cited on page 38).
- [155] Mengyao Xu, Gabriel Moreira, Ronay Ak, Radek Osmulski, Yauhen Babakhin, Zhiding Yu, Benedikt Schifferer, and Even Oldridge. *Llama Nemoretriever Colembed: Top-Performing Text-Image Retrieval Model*. 2025. URL: <https://arxiv.org/abs/2507.05513> (cited on pages 38, 51, 59–61, 141, 162).

- [156] Hongjin Su, Howard Yen, Mengzhou Xia, Weijia Shi, Niklas Muenighoff, Han-yu Wang, Haisu Liu, Quan Shi, Zachary S Siegel, Michael Tang, et al. *Bright: A realistic and challenging benchmark for reasoning-intensive retrieval*. In: *arXiv preprint arXiv:2407.12883* (2024) (cited on page 39).
- [157] Jordy Van Landeghem, Rubèn Tito, Łukasz Borchmann, Michał Pietruszka, Paweł Joziak, Rafał Powalski, Dawid Jurkiewicz, Mickaël Coustaty, Bertrand Anckaert, Ernest Valveny, et al. *Document understanding dataset and evaluation (dude)*. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023, pp. 19528–19540 (cited on page 39).
- [158] Qiuchen Wang, Ruixue Ding, Zehui Chen, Weiqi Wu, Shihang Wang, Pengjun Xie, and Feng Zhao. *Vidorag: Visual document retrieval-augmented generation via dynamic iterative reasoning agents*. In: *arXiv preprint arXiv:2502.18017* (2025) (cited on page 39).
- [159] Navve Wasserman, Roi Pony, Oshri Naparstek, Adi Raz Goldfarb, Eli Schwartz, Udi Barzelay, and Leonid Karlinsky. *REAL-MM-RAG: A Real-World Multi-Modal Retrieval Benchmark*. In: *arXiv preprint arXiv:2502.12342* (2025) (cited on page 39).
- [160] Deep Search Team. *Docling Technical Report*. Tech. rep. Version 1.0.0. Aug. 2024. DOI: [10.48550/arXiv.2408.09869](https://doi.org/10.48550/arXiv.2408.09869) (cited on page 40).
- [161] Qwen Team. *Qwen3 Technical Report*. 2025. URL: <https://arxiv.org/abs/2505.09388> (cited on page 40).
- [162] Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. *Qwen3 Embedding: Advancing Text Embedding and Reranking Through Foundation Models*. 2025. URL: <https://arxiv.org/abs/2506.05176> (cited on pages 40, 141, 165).
- [163] Leland McInnes, John Healy, and James Melville. *UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction*. 2020. URL: <https://arxiv.org/abs/1802.03426> (cited on page 40).
- [164] Ricardo JGB Campello, Davoud Moulavi, and Jörg Sander. *Density-based clustering based on hierarchical density estimates*. In: *Pacific-Asia conference on knowledge discovery and data mining*. Springer. 2013, pp. 160–172 (cited on page 40).
- [165] NeMo Data Designer Team. *NeMo Data Designer: A framework for generating synthetic data from scratch or based on your own seed data*. <https://github.com/NVIDIA-NeMo/DataDesigner>. GitHub Repository. 2025 (cited on page 40).
- [166] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibao Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. *Qwen2.5-VL Technical Report*. 2025. URL: <https://arxiv.org/abs/2502.13923> (cited on page 40).

- [167] Frank Liu, Kenneth Enevoldsen, Roman Solomatin, Isaac Chung, Tom Aarsen, and Zoltán Fődi. *Introducing RTEB: A New Standard for Retrieval Evaluation*. 2025. URL: <https://huggingface.co/bl og/rteb> (cited on page 42).
- [168] NVIDIA Ingest Development Team. *NVIDIA Ingest: An accelerated pipeline for document ingestion*. 2024 (cited on page 43).
- [169] Zero Entropy. *Introducing zerank-2*. Accessed: 2025-12-22. 2025. URL: <https://www.zeroentropy.dev/articles/zerank-2-advanced-instruction-following-multilingual-reranker> (cited on page 44).
- [170] Jina AI. *jina-reranker-m0: Multilingual Multimodal Document Reranker*. Accessed: 2025-12-22. 2025. URL: <https://jina.ai/news/jina-reranker-m0-multilingual-multimodal-document-reranker> (cited on page 44).
- [171] Weizhe Lin and Bill Byrne. *Retrieval Augmented Visual Question Answering with Outside Knowledge*. In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Ed. by Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 11238–11254. doi: [10.18653/v1/2022.emnlp-main.772](https://doi.org/10.18653/v1/2022.emnlp-main.772) (cited on page 50).
- [172] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. *Emergent Abilities of Large Language Models*. 2022. URL: <https://arxiv.org/abs/2206.07682> (cited on pages 51, 81, 95, 99).
- [173] Benjamin Clavié. *Towards Better Monolingual Japanese Retrievers with Multi-Vector Models*. 2024. URL: <https://arxiv.org/abs/2312.16144> (cited on page 51).
- [174] Xin Zhang, Yanzhao Zhang, Wen Xie, Mingxin Li, Ziqi Dai, Dingkun Long, Pengjun Xie, Meishan Zhang, Wenjie Li, and Min Zhang. *GME: Improving Universal Multimodal Retrieval by Multimodal LLMs*. 2025. URL: <https://arxiv.org/abs/2412.16855> (cited on pages 51, 58–60, 161).
- [175] Haonan Chen, Hong Liu, Yuping Luo, Liang Wang, Nan Yang, Furu Wei, and Zhicheng Dou. *MoCa: Modality-aware Continual Pre-training Makes Better Bidirectional Multimodal Embeddings*. 2025. URL: <https://arxiv.org/abs/2506.23115> (cited on pages 51, 56, 59, 60, 161).
- [176] Zeyuan Allen-Zhu and Yuanzhi Li. *Physics of Language Models: Part 1, Learning Hierarchical Language Structures*. 2025. URL: <https://arxiv.org/abs/2305.13673> (cited on page 52).
- [177] Orion Weller, Kathryn Ricci, Marc Marone, Antoine Chaffin, Dawn Lawrie, and Benjamin Van Durme. *Seq vs Seq: An Open Suite of Paired Encoders and Decoders*. 2025. URL: <https://arxiv.org/abs/2507.11412> (cited on pages 52, 56, 58, 60).

- [178] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. *Llama 2: Open Foundation and Fine-Tuned Chat Models*. In: (2023). Publisher: arXiv Version Number: 2. doi: [10.48550/ARXIV.2307.09288](https://arxiv.org/abs/2307.09288). (Visited on 09/03/2023) (cited on pages 52, 80, 86, 90, 93, 100).
- [179] Michael Tschanen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, Olivier Hénaff, Jeremiah Harmsen, Andreas Steiner, and Xiaohua Zhai. *SigLIP 2: Multilingual Vision-Language Encoders with Improved Semantic Understanding, Localization, and Dense Features*. 2025. URL: <https://arxiv.org/abs/2502.14786> (cited on pages 52, 58, 59, 162).
- [180] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. *BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation*. 2022. URL: <https://arxiv.org/abs/2201.12086> (cited on page 52).
- [181] An Yang, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoyan Huang, Jiandong Jiang, Jianhong Tu, Jianwei Zhang, Jingren Zhou, Junyang Lin, Kai Dang, Kexin Yang, Le Yu, Mei Li, Minmin Sun, Qin Zhu, Rui Men, Tao He, Weijia Xu, Wenbiao Yin, Wenyuan Yu, Xiafei Qiu, Xingzhang Ren, Xinlong Yang, Yong Li, Zhiying Xu, and Zipeng Zhang. *Qwen2.5-1M Technical Report*. 2025. URL: <https://arxiv.org/abs/2501.15383> (cited on pages 52, 77).
- [182] Andrés Marafioti, Orr Zohar, Miquel Farré, Merve Noyan, Elie Bakouch, Pedro Cuenca, Cyril Zakka, Loubna Ben Allal, Anton Lozhkov, Nouamane Tazi, Vaibhav Srivastav, Joshua Lochner, Hugo Larcher, Mathieu Morlon, Lewis Tunstall, Leandro von Werra, and Thomas Wolf. *SmolVLM: Redefining small and efficient multimodal models*. 2025. URL: <https://arxiv.org/abs/2504.05299> (cited on pages 52, 53, 57, 141).
- [183] Ziyi Lin, Chris Liu, Renrui Zhang, Peng Gao, Longtian Qiu, Han Xiao, Han Qiu, Chen Lin, Wenqi Shao, Keqin Chen, Jiaming Han, Siyuan Huang, Yichi Zhang, Xuming He, Hongsheng Li, and Yu Qiao. *SPHINX: The Joint Mixing of Weights, Tasks, and*

- Visual Embeddings for Multi-modal Large Language Models*. 2023. URL: <https://arxiv.org/abs/2311.07575> (cited on page 52).
- [184] Jiabo Ye, Anwen Hu, Haiyang Xu, Qinghao Ye, Ming Yan, Guohai Xu, Chenliang Li, Junfeng Tian, Qi Qian, Ji Zhang, Qin Jin, Liang He, Xin Alex Lin, and Fei Huang. *UReader: Universal OCR-free Visually-situated Language Understanding with Multimodal Large Language Model*. 2023. URL: <https://arxiv.org/abs/2310.05126> (cited on page 52).
- [185] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P. Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. *Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network*. 2016. URL: <https://arxiv.org/abs/1609.05158> (cited on page 53).
- [186] Brandon McKinzie, Zhe Gan, Jean-Philippe Fauconnier, Sam Dodge, Bowen Zhang, Philipp Dufter, Dhruti Shah, Xianzhi Du, Futang Peng, Floris Weers, Anton Belyi, Haotian Zhang, Karanjeet Singh, Doug Kang, Hongyu Hè, Max Schwarzer, Tom Gunter, Xiang Kong, Aonan Zhang, Jianyu Wang, Chong Wang, Nan Du, Tao Lei, Sam Wiseman, Mark Lee, Zirui Wang, Ruoming Pang, Peter Gräsch, Alexander Toshev, and Yinfei Yang. *MM1: Methods, Analysis & Insights from Multimodal LLM Pre-training*. In: (2024). Publisher: [object Object] Version Number: 1. doi: [10.48550/ARXIV.2403.09611](https://arxiv.org/abs/2403.09611). (Visited on 03/17/2024) (cited on pages 53, 56, 57).
- [187] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. *Improved Baselines with Visual Instruction Tuning*. Version Number: 2. 2023. doi: [10.48550/ARXIV.2310.03744](https://arxiv.org/abs/2310.03744). URL: <https://arxiv.org/abs/2310.03744> (visited on 06/02/2024) (cited on page 53).
- [188] Mathew Huerta-Enochian and Seung Yong Ko. *Instruction Fine-Tuning: Does Prompt Loss Matter?* 2024. URL: <https://arxiv.org/abs/2401.13586> (cited on page 53).
- [189] Zhengyan Shi, Adam X. Yang, Bin Wu, Laurence Aitchison, Emine Yilmaz, and Aldo Lipani. *Instruction Tuning With Loss Over Instructions*. 2024. URL: <https://arxiv.org/abs/2405.14394> (cited on page 53).
- [190] Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Gabriel Martín Blázquez, Guilherme Penedo, Lewis Tunstall, Andrés Marafioti, Hynek Kydlíček, Agustín Piqueres Lajarín, Vaibhav Srivastav, Joshua Lochner, Caleb Fahlgren, Xuan-Son Nguyen, Clémentine Fourrier, Ben Burtenshaw, Hugo Larcher, Haojun Zhao, Cyril Zakka, Mathieu Morlon, Colin Raffel, Leandro von Werra, and Thomas Wolf. *SmolLM2: When Smol Goes Big – Data-Centric Training of a Small Language Model*. 2025. URL: <https://arxiv.org/abs/2502.02737> (cited on page 53).
- [191] Shengding Hu, Yuge Tu, Xu Han, Chaoqun He, Ganqu Cui, Xiang Long, Zhi Zheng, Yewei Fang, Yuxiang Huang, Weilin Zhao, Xinrong Zhang, Zheng Leng Thai, Kaihuo Zhang, Chongyi Wang, Yuan Yao, Chenyang Zhao, Jie Zhou, Jie Cai, Zhongwu Zhai, Ning Ding, Chao Jia, Guoyang Zeng, Dahai Li, Zhiyuan Liu, and Maosong Sun. *MiniCPM: Unveiling the Potential of Small Language Models with Scalable Training Strategies*. 2024. URL: <https://arxiv.org/abs/2404.06395> (cited on pages 53, 101).

- [192] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. *Representation Learning with Contrastive Predictive Coding*. Version Number: 2. 2018. DOI: [10.48550/ARXIV.1807.03748](https://arxiv.org/abs/1807.03748). URL: <https://arxiv.org/abs/1807.03748> (visited on 06/04/2024) (cited on pages 54, 72).
- [193] Chenghao Xiao, Isaac Chung, Imene Kerboua, Jamie Stirling, Xin Zhang, Márton Kardos, Roman Solomatin, Noura Al Moubayed, Kenneth Enevoldsen, and Niklas Muennighoff. *MIEB: Massive Image Embedding Benchmark*. 2025. URL: <https://arxiv.org/abs/2504.10471> (cited on pages 54, 55, 162).
- [194] Bryan A. Plummer, Liwei Wang, Chris M. Cervantes, Juan C. Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. *Flickr30k Entities: Collecting Region-to-Phrase Correspondences for Richer Image-to-Sentence Models*. Version Number: 4. 2015. DOI: [10.48550/ARXIV.1505.04870](https://arxiv.org/abs/1505.04870). URL: <https://arxiv.org/abs/1505.04870> (visited on 06/14/2024) (cited on page 54).
- [195] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. *3D Object Representations for Fine-Grained Categorization*. In: *2013 IEEE International Conference on Computer Vision Workshops*. 2013, pp. 554–561. DOI: [10.1109/ICCVW.2013.77](https://arxiv.org/abs/10.1109/ICCVW.2013.77) (cited on page 54).
- [196] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. *Food-101 – Mining Discriminative Components with Random Forests*. In: *Computer Vision – ECCV 2014*. Ed. by David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars. Cham: Springer International Publishing, 2014, pp. 446–461 (cited on page 54).
- [197] Yousif Khairuddin and Zhuofa Chen. *Facial Emotion Recognition: State of the Art Performance on FER2013*. 2021. URL: <https://arxiv.org/abs/2105.03588> (cited on page 54).
- [198] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. *EuroSAT: A Novel Dataset and Deep Learning Benchmark for Land Use and Land Cover Classification*. 2019. URL: <https://arxiv.org/abs/1709.00029> (cited on page 54).
- [199] Anwen Hu, Haiyang Xu, Liang Zhang, Jiabo Ye, Ming Yan, Ji Zhang, Qin Jin, Fei Huang, and Jingren Zhou. *mPLUG-DocOwl2: High-resolution Compressing for OCR-free Multi-page Document Understanding*. 2024. URL: <https://arxiv.org/abs/2409.03420> (cited on page 57).
- [200] Ting Jiang, Minghui Song, Zihan Zhang, Haizhen Huang, Weiwei Deng, Feng Sun, Qi Zhang, Deqing Wang, and Fuzhen Zhuang. *E5-V: Universal Embeddings with Multimodal Large Language Models*. 2024. URL: <https://arxiv.org/abs/2407.12580> (cited on pages 58, 59, 161).
- [201] Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, Nathan Cooper, Griffin Adams, Jeremy Howard, and Iacopo Poli. *Smarter, Better, Faster, Longer: A Modern Bidirectional Encoder for Fast, Memory Efficient, and Long Context Finetuning and Inference*. 2024. URL: <https://arxiv.org/abs/2412.13663> (cited on pages 58, 60, 66, 71, 72, 167, 174).

- [202] Junjie Zhou, Zheng Liu, Shitao Xiao, Bo Zhao, and Yongping Xiong. *VISTA: Visualized Text Embedding For Universal Multi-Modal Retrieval*. 2024. URL: <https://arxiv.org/abs/2406.04292> (cited on pages 59, 162).
- [203] Ahmed Masry and Enamul Hoque. *ColFlor: Towards BERT-Size Vision-Language Document Retrieval Models*. In: (2024) (cited on pages 59, 162).
- [204] Bin Xiao, Haiping Wu, Weijian Xu, Xiyang Dai, Houdong Hu, Yumao Lu, Michael Zeng, Ce Liu, and Lu Yuan. *Florence-2: Advancing a Unified Representation for a Variety of Vision Tasks*. 2023. URL: <https://arxiv.org/abs/2311.06242> (cited on page 59).
- [205] Zilin Xiao, Qi Ma, Mengting Gu, Chun-cheng Jason Chen, Xintao Chen, Vicente Ordonez, and Vijai Mohan. *MetaEmbed: Scaling Multimodal Retrieval at Test-Time with Flexible Late Interaction*. 2025. URL: <https://arxiv.org/abs/2509.18095> (cited on pages 60, 61).
- [206] Lucas Beyer*, Andreas Steiner*, André Susano Pinto*, Alexander Kolesnikov*, Xiao Wang*, Xiaohua Zhai*, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Jeremiah Harmsen, Daniel Keysers, Neil Houlsby, Xi Chen, Emanuele Bugliarello, Thomas Unterthiner, Keran Rong, Matthias Minderer, Ioana Bica, Ivana Balazevic, Joan Puigcerver, Julian Eisenschlos, Manoj Kumar, Matko Bošnjak, Matthias Bauer, Fangyu Liu, Adam Grycner, Alexey Gritsenko, Paul Voigtlaender, Pinelopi Papalampidi, Olivier Henaff, Skanda Koppula, Xi Xiong, Radu Soricut, Model release contributors and general support, Tris Warkentin, Kat Black, Luiz Gustavo Martins, Glenn Cameron, Raj Gundluru, Manvinder Singh, Meg Risdal, Nilay Chauhan, Nate Keating, Nesh Devanathan, Elisa Bandy, Joe Fernandez, Antonia Paterson, Jenny Brennan, Tom Eccles, Pankil Botadra, Ben Bariach, Lav Rai, Minwoo Park, Dustin Luong, Daniel Vlasic, Bo Wu, Wenming Ye, Divyashree Sreepathihalli, Kiranbir Sodhia, Alek Andreev, Armand Joulin, Surya Bhupatiraju, Minh Giang, Joelle Barral, and Zoubin Ghahramani. *PaliGemma*. 2024. DOI: [10.34740/KAGGLE/M/23393](https://doi.org/10.34740/KAGGLE/M/23393). URL: <https://www.kaggle.com/m/23393> (visited on 06/04/2024) (cited on pages 60, 132).
- [207] Zach Nussbaum, John X. Morris, Brandon Duderstadt, and Andriy Mulyar. *Nomic Embed: Training a Reproducible Long Context Text Embedder*. 2024. URL: <https://arxiv.org/abs/2402.01613> (cited on pages 60, 72, 174).
- [208] Jo Bergum. *Scaling ColPali to billions of PDFs with Vespa* — blog.vespa.ai/scaling-colpali-to-billions/. 2025 (cited on pages 60, 61).
- [209] Patrick Fernandes, Behrooz Ghorbani, Xavier Garcia, Markus Freitag, and Orhan Firat. *Scaling Laws for Multilingual Neural Machine Translation*. In: *Proceedings of the 40th International Conference on Machine Learning*. Ed. by Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett. Vol. 202. Proceedings of Machine Learning Research. PMLR, 2023, pp. 10053–10071 (cited on pages 61, 88, 121).

- [210] Xin Zhang, Yanzhao Zhang, Dingkun Long, Wen Xie, Ziqi Dai, Jialong Tang, Huan Lin, Baosong Yang, Pengjun Xie, Fei Huang, et al. *mGTE: Generalized Long-Context Text Representation and Reranking Models for Multilingual Text Retrieval*. In: *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*. 2024, pp. 1393–1412 (cited on pages 66, 71).
- [211] Dawei Zhu, Liang Wang, Nan Yang, Yifan Song, Wenhao Wu, Furu Wei, and Sujian Li. *LongEmbed: Extending Embedding Models for Long Context Retrieval*. 2024. URL: <https://arxiv.org/abs/2404.12096> (cited on pages 66, 72).
- [212] Peng Xu, Wei Ping, Xianchao Wu, Lawrence McAfee, Chen Zhu, Zihan Liu, Sandeep Subramanian, Evelina Bakhturina, Mohammad Shoeybi, and Bryan Catanzaro. *Retrieval meets Long Context Large Language Models*. 2024. URL: <https://arxiv.org/abs/2310.03025> (cited on page 66).
- [213] Ziyang Jiang, Xueguang Ma, and Wenhui Chen. *LongRAG: Enhancing Retrieval-Augmented Generation with Long-context LLMs*. 2024. URL: <https://arxiv.org/abs/2406.15319> (cited on page 66).
- [214] Gemini Team et al. *Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context*. 2024. URL: <https://arxiv.org/abs/2403.05530> (cited on page 67).
- [215] Anthropic. *Introducing Contextual Retrieval*. Accessed: 2025-02-10. Sept. 19, 2024. URL: <https://www.anthropic.com/news/contextual-retrieval> (cited on pages 67, 68, 71).
- [216] Jon Saad-Falcon, Daniel Y. Fu, Simran Arora, Neel Guha, and Christopher Ré. *Benchmarking and Building Long-Context Retrieval Models with LoCo and M2-BERT*. 2024. URL: <https://arxiv.org/abs/2402.07440> (cited on page 67).
- [217] Yang Zhou, Hongyi Liu, Zhuoming Chen, Yuandong Tian, and Beidi Chen. *GSM-Infinite: How Do Your LLMs Behave over Infinitely Increasing Context Length and Reasoning Complexity?* 2025. URL: <https://arxiv.org/abs/2502.05252> (cited on page 67).
- [218] Jerry Liu. *LlamaIndex*. Nov. 2022. DOI: [10.5281/zenodo.1234](https://doi.org/10.5281/zenodo.1234) (cited on page 68).
- [219] Zijie Zhong, Hanwen Liu, Xiaoya Cui, Xiaofan Zhang, and Zengchang Qin. *Mix-of-Granularity: Optimize the Chunking Granularity for Retrieval-Augmented Generation*. 2025. URL: <https://arxiv.org/abs/2406.00456> (cited on page 68).
- [220] Harrison Chase. *LangChain*. Oct. 2022. URL: <https://github.com/langchain-ai/langchain> (cited on pages 68, 70).
- [221] Hongjin Qian, Zheng Liu, Kelong Mao, Yujia Zhou, and Zhicheng Dou. *Grounding Language Model with Chunking-Free In-Context Retrieval*. 2024. URL: <https://arxiv.org/abs/2402.09760> (cited on page 68).
- [222] Mykhailo Poliakov and Nadiya Shvai. *Multi-Meta-RAG: Improving RAG for Multi-Hop Queries using Database Filtering with LLM-Extracted Metadata*. 2024. URL: <https://arxiv.org/abs/2406.13213> (cited on page 68).

- [223] Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, and Jonathan Larson. *From Local to Global: A Graph RAG Approach to Query-Focused Summarization*. 2024. URL: <https://arxiv.org/abs/2404.16130> (cited on page 69).
- [224] Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher D. Manning. *RAPTOR: Recursive Abstractive Processing for Tree-Organized Retrieval*. 2024. URL: <https://arxiv.org/abs/2401.18059> (cited on pages 69, 71).
- [225] Wenhan Xiong, Xiang Lorraine Li, Srini Iyer, Jingfei Du, Patrick Lewis, William Yang Wang, Yashar Mehdad, Wen-tau Yih, Sebastian Riedel, Douwe Kiela, and Barlas Oğuz. *Answering Complex Open-Domain Questions with Multi-Hop Dense Retrieval*. 2021. URL: <https://arxiv.org/abs/2009.12756> (cited on page 69).
- [226] Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. *Interleaving Retrieval with Chain-of-Thought Reasoning for Knowledge-Intensive Multi-Step Questions*. 2023. URL: <https://arxiv.org/abs/2212.10509> (cited on page 69).
- [227] Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. *Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection*. 2023. URL: <https://arxiv.org/abs/2310.11511> (cited on page 69).
- [228] Zeyuan Allen-Zhu. *ICML 2024 Tutorial: Physics of Language Models*. Project page: <https://physics.allen-zhu.com/>. July 2024 (cited on page 69).
- [229] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. *SQuAD: 100,000+ Questions for Machine Comprehension of Text*. In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Ed. by Jian Su, Kevin Duh, and Xavier Carreras. Austin, Texas: Association for Computational Linguistics, Nov. 2016, pp. 2383–2392. doi: [10.18653/v1/D16-1264](https://doi.org/10.18653/v1/D16-1264) (cited on page 70).
- [230] Timo Möller, Anthony Reina, Raghavan Jayakumar, and Malte Pietsch. *COVID-QA: A Question Answering Dataset for COVID-19*. In: *Proceedings of the 1st Workshop on NLP for COVID-19 at ACL 2020*. Ed. by Karin Verspoor, Kevin Bretonnel Cohen, Mark Dredze, Emilio Ferrara, Jonathan May, Robert Munro, Cecile Paris, and Byron Wallace. Online: Association for Computational Linguistics, July 2020 (cited on page 70).
- [231] Tomáš Kočiský, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. *The NarrativeQA Reading Comprehension Challenge*. 2017. URL: <https://arxiv.org/abs/1712.07040> (cited on page 70).
- [232] Antoine Chaffin. *ModernBERT-embed-large*. 2025. URL: <https://huggingface.co/lightonai/modernbert-embed-large> (cited on pages 71, 72).
- [233] Antoine Chaffin. *GTE-ModernColBERT*. 2025. URL: <https://huggingface.co/lightonai/GTE-ModernColBERT-v1> (cited on pages 71, 72, 141, 164, 174).

- [234] Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernández Ábrego, Ji Ma, Vincent Y. Zhao, Yi Luan, Keith B. Hall, Ming-Wei Chang, and Yinfei Yang. *Large Dual Encoders Are Generalizable Retrievers*. 2021. URL: <https://arxiv.org/abs/2112.07899> (cited on page 72).
- [235] Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. *Unsupervised Dense Information Retrieval with Contrastive Learning*. Version Number: 4. 2021. DOI: [10.48550/ARXIV.2112.09118](https://arxiv.org/abs/2112.09118). URL: <https://arxiv.org/abs/2112.09118> (visited on 06/07/2024) (cited on page 72).
- [236] Florian Schroff, Dmitry Kalenichenko, and James Philbin. *FaceNet: A Unified Embedding for Face Recognition and Clustering*. In: (2015). Publisher: arXiv Version Number: 3. DOI: [10.48550/ARXIV.1503.03832](https://arxiv.org/abs/1503.03832). (Visited on 06/07/2024) (cited on page 72).
- [237] Mingxin Li, Zhijie Nie, Yanzhao Zhang, Dingkun Long, Richong Zhang, and Pengjun Xie. *Improving General Text Embedding Model: Tackling Task Conflict and Data Imbalance through Model Merging*. 2024. URL: <https://arxiv.org/abs/2410.15035> (cited on pages 76, 77, 165).
- [238] Ke Wang, Nikolaos Dimitriadis, Alessandro Favero, Guillermo Ortiz-Jimenez, Francois Fleuret, and Pascal Frossard. *LiNeS: Post-training Layer Scaling Prevents Forgetting and Enhances Model Merging*. 2025. URL: <https://arxiv.org/abs/2410.17146> (cited on pages 76, 77).
- [239] OpenAI. *GPT-4 Technical Report*. arXiv:2303.08774 [cs]. Mar. 2023. URL: <http://arxiv.org/abs/2303.08774> (visited on 05/24/2023) (cited on page 80).
- [240] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. *Llama: Open and efficient foundation language models*. In: *arXiv preprint arXiv:2302.13971* (2023) (cited on pages 80–83, 85–87, 90).
- [241] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. *Qwen Technical Report*. 2023 (cited on pages 80, 82).
- [242] Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, L  lio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Th  ophile Gerv  t, Thibaut Lavril, Thomas Wang, Timoth  e

- Lacroix, and William El Sayed. *Mixtral of Experts*. 2024 (cited on page 80).
- [243] Rishi Bommasani, Kevin Klyman, Shayne Longpre, Sayash Kapoor, Nestor Maslej, Betty Xiong, Daniel Zhang, and Percy Liang. *The Foundation Model Transparency Index*. Oct. 2023. URL: <https://crfm.stanford.edu/fmti/FMTI.pdf> (cited on pages 80, 82, 99).
 - [244] Stephen Casper, Carson Ezell, Charlotte Siegmann, Noam Kolt, Taylor Lynn Curtis, Benjamin Bucknall, Andreas Haupt, Kevin Wei, J  r  my Scheurer, Marius Hobbhahn, Lee Sharkey, Satyapriya Krishna, Marvin Von Hagen, Silas Alberti, Alan Chan, Qinyi Sun, Michael Gerovitch, David Bau, Max Tegmark, David Krueger, and Dylan Hadfield-Menell. *Black-Box Access is Insufficient for Rigorous AI Audits*. 2024 (cited on page 80).
 - [245] Evan Hubinger, Carson Denison, Jesse Mu, Mike Lambert, Meg Tong, Monte MacDiarmid, Tamera Lanham, Daniel M. Ziegler, Tim Maxwell, Newton Cheng, Adam Jermyn, Amanda Askill, Ansh Radhakrishnan, Cem Anil, David Duvenaud, Deep Ganguli, Fazl Barez, Jack Clark, Kamal Ndousse, Kshitij Sachan, Michael Sellitto, Mrinank Sharma, Nova DasSarma, Roger Grosse, Shauna Kravec, Yuntao Bai, Zachary Witten, Marina Favaro, Jan Brauner, Holden Karnofsky, Paul Christiano, Samuel R. Bowman, Logan Graham, Jared Kaplan, S  ren Mindermann, Ryan Greenblatt, Buck Shlegeris, Nicholas Schiefer, and Ethan Perez. *Sleeper Agents: Training Deceptive LLMs that Persist Through Safety Training*. 2024 (cited on page 80).
 - [246] NewYorkTimes. *The Times sues OpenAI and Microsoft over A.I. use of copyrighted work*. <https://www.nytimes.com/2023/12/27/business/media/new-york-times-open-ai-microsoft-lawsuit.html>. 2023 (cited on page 80).
 - [247] Pamela Samuelson. *Generative AI meets copyright*. In: *Science* 381.6654 (2023), pp. 158–161 (cited on page 80).
 - [248] Lucas Bandarkar, Davis Liang, Benjamin Muller, Mikel Artetxe, Satya Narayan Shukla, Donald Husa, Naman Goyal, Abhinandan Krishnan, Luke Zettlemoyer, and Madian Khabsa. *The Belebele Benchmark: a Parallel Reading Comprehension Dataset in 122 Language Variants*. 2023 (cited on pages 81, 93).
 - [249] Emily M Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. *On the dangers of stochastic parrots: Can language models be too big?* In: *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*. 2021, pp. 610–623 (cited on page 81).
 - [250] Shibani Santurkar, Esin Durmus, Faisal Ladhak, Cinoo Lee, Percy Liang, and Tatsunori Hashimoto. *Whose Opinions Do Language Models Reflect?* 2023 (cited on pages 81, 83).
 - [251] Daria Soboleva, Faisal Al-Khateeb, Robert Myers, Jacob R Steeves, Joel Hestness, and Nolan Dey. *SlimPajama: A 627B token cleaned and deduplicated version of RedPajama*. 2023. URL: <https://huggingface.co/datasets/cerebras/SlimPajama-627B> (cited on pages 81, 85).

- [252] Hugo Laurençon, Lucile Saulnier, Thomas Wang, Christopher Akiki, Albert Villanova del Moral, Teven Le Scao, Leandro Von Werra, Chenghao Mou, Eduardo González Ponferrada, Huu Nguyen, Jörg Frohberg, Mario Šaško, Quentin Lhoest, Angelina McMillan-Major, Gerard Dupont, Stella Biderman, Anna Rogers, Loubna Ben allal, Francesco De Toni, Giada Pistilli, Olivier Nguyen, Somaieh Nikpoor, Maraim Masoud, Pierre Colombo, Javier de la Rosa, Paulo Villegas, Tristan Thrush, Shayne Longpre, Sebastian Nagel, Leon Weber, Manuel Muñoz, Jian Zhu, Daniel Van Strien, Zaid Alyafeai, Khalid Almubarak, Minh Chien Vu, Itziar Gonzalez-Dios, Aitor Soroa, Kyle Lo, Manan Dey, Pedro Ortiz Suarez, Aaron Gokaslan, Shamik Bose, David Adelani, Long Phan, Hieu Tran, Ian Yu, Suhas Pai, Jenny Chim, Violette Lepercq, Suzana Ilic, Margaret Mitchell, Sasha Alexandra Luccioni, and Yacine Jernite. *The BigScience ROOTS Corpus: A 1.6TB Composite Multilingual Dataset*. 2023 (cited on pages 81, 96).
- [253] Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. *Bloom: A 176b-parameter open-access multilingual language model*. In: *arXiv preprint arXiv:2211.05100* (2022) (cited on pages 81, 82, 84, 85, 93, 99).
- [254] Oleh Shliazhko, Alena Fenogenova, Maria Tikhonova, Vladislav Mikhailov, Anastasia Kozlova, and Tatiana Shavrina. *mGPT: Few-Shot Learners Go Multilingual*. 2022. doi: [10.48550/ARXIV.2204.07580](https://arxiv.org/abs/2204.07580). URL: <https://arxiv.org/abs/2204.07580> (cited on pages 81, 94).
- [255] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. *Training Compute-Optimal Large Language Models*. 2022 (cited on pages 81, 90, 93, 121).
- [256] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayanan Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel.

- PaLM: Scaling Language Modeling with Pathways*. 2022 (cited on pages 81, 85).
- [257] Nikhil Sardana and Jonathan Frankle. *Beyond Chinchilla-Optimal: Accounting for Inference in Language Model Scaling Laws*. 2023 (cited on pages 81, 90).
 - [258] Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, et al. *Glm-130b: An open bilingual pre-trained model*. In: *arXiv preprint arXiv:2210.02414* (2022) (cited on page 82).
 - [259] Rachel Bawden, Hatim Bourfoune, Bertrand Cabot, Nathan Cassereau, Pierre Cornette, Marco Naguib, Aurélie Névéol, and François Yvon. ‘Les modèles Bloom pour le traitement automatique de la langue française’. working paper or preprint. Feb. 2024 (cited on page 82).
 - [260] Igor Shilov, Matthieu Meeus, and Yves-Alexandre de Montjoye. *Mosaic Memory: Fuzzy Duplication in Copyright Traps for Large Language Models*. 2024. URL: <https://arxiv.org/abs/2405.15523> (cited on page 82).
 - [261] Roberto Navigli, Simone Conia, and Björn Ross. *Biases in Large Language Models: Origins, Inventory, and Discussion*. In: *J. Data and Information Quality* 15.2 (2023). doi: [10.1145/3597307](https://doi.org/10.1145/3597307) (cited on page 83).
 - [262] Niklas Muennighoff, Alexander M. Rush, Boaz Barak, Teven Le Scao, Aleksandra Piktus, Nouamane Tazi, Sampo Pyysalo, Thomas Wolf, and Colin Raffel. *Scaling Data-Constrained Language Models*. 2023 (cited on pages 83, 85, 89).
 - [263] Julien Abadji, Pedro Ortiz Suarez, Laurent Romary, and Benoit Sagot. *Towards a Cleaner Document-Oriented Multilingual Crawled Corpus*. In: *Proceedings of the Thirteenth Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, June 2022, pp. 4344–4355 (cited on page 83).
 - [264] Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. *mT5: A massively multilingual pre-trained text-to-text transformer*. 2021 (cited on page 83).
 - [265] Thuat Nguyen, Chien Van Nguyen, Viet Dac Lai, Hieu Man, Nghia Trung Ngo, Franck Dernoncourt, Ryan A. Rossi, and Thien Huu Nguyen. *CulturaX: A Cleaned, Enormous, and Multilingual Dataset for Large Language Models in 167 Languages*. 2023 (cited on page 83).
 - [266] Michael Hart. *Project gutenber*. 1971. URL: <https://www.gutenberg.org/> (cited on pages 84, 85).
 - [267] Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. *The Secret Sharer: Evaluating and Testing Unintended Memorization in Neural Networks*. In: *28th USENIX Security Symposium (USENIX Security 19)*. Santa Clara, CA: USENIX Association, Aug. 2019, pp. 267–284 (cited on page 85).

- [268] Raymond Li, Loubna Ben Allal, Yangtian Zi, Niklas Muennighoff, Denis Kocetkov, Chenghao Mou, Marc Marone, Christopher Akiki, Jia Li, Jenny Chim, Qian Liu, Evgenii Zheltonozhskii, Terry Yue Zhuo, Thomas Wang, Olivier Dehaene, Mishig Davaadorj, Joel Lamy-Poirier, João Monteiro, Oleh Shliazhko, Nicolas Gontier, Nicholas Meade, Armel Zebaze, Ming-Ho Yee, Logesh Kumar Umapathi, Jian Zhu, Benjamin Lipkin, Muhtasham Oblokulov, Zhiruo Wang, Rudra Murthy, Jason Stillerman, Siva Sankalp Patel, Dmitry Abulkhanov, Marco Zocca, Manan Dey, Zhihan Zhang, Nour Fahmy, Urvashi Bhattacharyya, Wenhao Yu, Swayam Singh, Sasha Luccioni, Paulo Villegas, Maxim Kunakov, Fedor Zhdanov, Manuel Romero, Tony Lee, Nadav Timor, Jennifer Ding, Claire Schlesinger, Hailey Schoelkopf, Jan Ebert, Tri Dao, Mayank Mishra, Alex Gu, Jennifer Robinson, Carolyn Jane Anderson, Brendan Dolan-Gavitt, Danish Contractor, Siva Reddy, Daniel Fried, Dzmitry Bahdanau, Yacine Jernite, Carlos Muñoz Ferrandis, Sean Hughes, Thomas Wolf, Arjun Guha, Leandro von Werra, and Harm de Vries. *StarCoder: may the source be with you!* 2023 (cited on page 85).
- [269] Denis Kocetkov, Raymond Li, Loubna Ben Allal, Jia Li, Chenghao Mou, Carlos Muñoz Ferrandis, Yacine Jernite, Margaret Mitchell, Sean Hughes, Thomas Wolf, Dzmitry Bahdanau, Leandro von Werra, and Harm de Vries. *The Stack: 3 TB of permissively licensed source code*. 2022 (cited on page 85).
- [270] Yujia Li, David Choi, Junyoung Chung, Nate Kushman, Julian Schrittwieser, Rémi Leblond, Tom Eccles, James Keeling, Felix Gimeno, Agustin Dal Lago, Thomas Hubert, Peter Choy, Cyprien de Masson d’Autume, Igor Babuschkin, Xinyun Chen, Po-Sen Huang, Johannes Welbl, Sven Gowal, Alexey Cherepanov, James Molloy, Daniel J. Mankowitz, Esme Sutherland Robson, Pushmeet Kohli, Nando de Freitas, Koray Kavukcuoglu, and Oriol Vinyals. *Competition-level code generation with AlphaCode*. In: *Science* 378.6624 (2022), pp. 1092–1097. doi: [10.1126/science.abq1158](https://doi.org/10.1126/science.abq1158) (cited on page 85).
- [271] Yuanzhi Li, Sébastien Bubeck, Ronen Eldan, Allie Del Giorno, Suriya Gunasekar, and Yin Tat Lee. *Textbooks Are All You Need II: phi-1.5 technical report*. 2023 (cited on page 85).
- [272] Rohan Anil et al. *PaLM 2 Technical Report*. 2023 (cited on pages 86, 90).
- [273] Eleftheria Briakou, Colin Cherry, and George Foster. *Searching for Needles in a Haystack: On the Role of Incidental Bilingualism in PaLM’s Translation Capability*. In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki. Toronto, Canada: Association for Computational Linguistics, July 2023, pp. 9432–9452. doi: [10.18653/v1/2023.acl-long.524](https://doi.org/10.18653/v1/2023.acl-long.524) (cited on page 86).
- [274] Jörg Tiedemann. *Parallel Data, Tools and Interfaces in OPUS*. In: *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC’12)*. Ed. by Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Mehmet Ugur Dogan, Bente Maegaard, Joseph Mariani, Jan Odijk, and Stelios Piperidis.

Istanbul, Turkey: European Language Resources Association (ELRA), 2012 (cited on page 86).

- [275] Gema Ramírez-Sánchez, Jaume Zaragoza-Bernabeu, Marta Bañón, and Sergio Ortiz Rojas. *Bifixer and Bicleaner: two open-source tools to clean your parallel data*. In: *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*. Ed. by André Martins, Helena Moniz, Sara Fumega, Bruno Martins, Fernando Batista, Luisa Coheur, Carla Parra, Isabel Trancoso, Marco Turchi, Arianna Bisazza, Joss Moorkens, Ana Guerbero, Mary Nurminen, Lena Marg, and Mikel L. Forcada. Lisboa, Portugal: European Association for Machine Translation, Nov. 2020, pp. 291–298 (cited on page 86).
- [276] Ricardo Rei, Marcos Treviso, Nuno M. Guerreiro, Chrysoula Zerva, Ana C Farinha, Christine Maroti, José G. C. de Souza, Taisiya Glushkova, Duarte Alves, Luisa Coheur, Alon Lavie, and André F. T. Martins. *CometKiwi: IST-Unbabel 2022 Submission for the Quality Estimation Shared Task*. In: *Proceedings of the Seventh Conference on Machine Translation (WMT)*. Ed. by Philipp Koehn, Loic Barrault, Ondřej Bojar, Fethi Bougares, Rajen Chatterjee, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Alexander Fraser, Markus Freitag, Yvette Graham, Roman Grundkiewicz, Paco Guzman, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Tom Kocmi, André Martins, Makoto Morishita, Christof Monz, Masaaki Nagata, Toshiaki Nakazawa, Matteo Negri, Aurélie Névél, Mariana Neves, Martin Popel, Marco Turchi, and Marcos Zampieri. Abu Dhabi, United Arab Emirates (Hybrid): Association for Computational Linguistics, Dec. 2022, pp. 634–645 (cited on page 86).
- [277] Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. *RoFormer: Enhanced Transformer with Rotary Position Embedding*. 2023 (cited on page 86).
- [278] Rico Sennrich, Barry Haddow, and Alexandra Birch. *Neural Machine Translation of Rare Words with Subword Units*. 2016 (cited on page 87).
- [279] Taku Kudo. *Subword Regularization: Improving Neural Network Translation Models with Multiple Subword Candidates*. 2018 (cited on page 87).
- [280] Phillip Rust, Jonas Pfeiffer, Ivan Vulić, Sebastian Ruder, and Iryna Gurevych. *How Good is Your Tokenizer? On the Monolingual Performance of Multilingual Language Models*. 2021 (cited on pages 87, 121).
- [281] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. *Scaling Laws for Neural Language Models*. In: *CoRR abs/2001.08361* (2020) (cited on pages 88, 90, 102, 121).
- [282] Risto Luukkainen, Ville Komulainen, Jouni Luoma, Anni Eskelinen, Jenna Kanerva, Hanna-Mari Kupari, Filip Ginter, Veronika Laippala, Niklas Muennighoff, Aleksandra Piktus, Thomas Wang, Nouamane Tazi, Teven Scao, Thomas Wolf, Osmo Suominen, Samuli Sairanen, Mikko Merioksa, Jyrki Heinonen, Aija Vahtola, Samuel Antao, and Sampo Pyysalo. *FinGPT: Large Generative Models for a Small Language*. In: *Proceedings of the 2023 Conference on*

- Empirical Methods in Natural Language Processing*. Ed. by Houda Bouamor, Juan Pino, and Kalika Bali. Singapore: Association for Computational Linguistics, Dec. 2023, pp. 2710–2726. doi: [10.18653/v1/2023.emnlp-main.164](https://doi.org/10.18653/v1/2023.emnlp-main.164) (cited on page 89).
- [283] Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng-Xin Yong, Hailey Schoelkopf, et al. *Crosslingual generalization through multitask finetuning*. In: *arXiv preprint arXiv:2211.01786* (2022) (cited on page 89).
 - [284] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. *OPT: Open Pre-trained Transformer Language Models*. 2022 (cited on pages 90, 93).
 - [285] Alexandra Sasha Luccioni, Sylvain Viguier, and Anne-Laure Ligozat. *Estimating the Carbon Footprint of BLOOM, a 176B Parameter Language Model*. 2022 (cited on page 90).
 - [286] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. *HellaSwag: Can a Machine Really Finish Your Sentence?* 2019 (cited on page 91).
 - [287] Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. *PIQA: Reasoning about Physical Commonsense in Natural Language*. 2019 (cited on page 91).
 - [288] Johannes Welbl, Nelson F. Liu, and Matt Gardner. *Crowdsourcing Multiple Choice Science Questions*. 2017 (cited on page 91).
 - [289] Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. *Think you have Solved Question Answering? Try ARC, the AI2 Reasoning Challenge*. 2018 (cited on page 91).
 - [290] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. *Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena*. 2023 (cited on page 91).
 - [291] Martin d’Hoffschmidt, Wacim Belblidia, Tom Brendlé, Quentin Heinrich, and Maxime Vidal. *FQuAD: French Question Answering Dataset*. 2020 (cited on page 92).
 - [292] Chin-Yew Lin. *ROUGE: A Package for Automatic Evaluation of Summaries*. In: *Text Summarization Branches Out*. Barcelona, Spain: Association for Computational Linguistics, July 2004, pp. 74–81 (cited on page 92).
 - [293] Moussa Kamal Eddine, Antoine J-P Tixier, and Michalis Vazirgiannis. *BARThez: a Skilled Pretrained French Sequence-to-Sequence Model*. 2020 (cited on page 93).
 - [294] Duarte M Alves, Nuno M Guerreiro, João Alves, José Pombal, Ricardo Rei, José GC de Souza, Pierre Colombo, and André FT Martins. *Steering Large Language Models for Machine Translation with Finetuning and In-Context Learning*. In: *arXiv preprint arXiv:2310.13448* (2023) (cited on page 93).

- [295] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. *Bleu: a Method for Automatic Evaluation of Machine Translation*. In: *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. Philadelphia, Pennsylvania, USA: Association for Computational Linguistics, July 2002, pp. 311–318. doi: [10.3115/1073083.1073135](https://doi.org/10.3115/1073083.1073135) (cited on page 93).
- [296] Matt Post. *A Call for Clarity in Reporting BLEU Scores*. In: *Proceedings of the Third Conference on Machine Translation: Research Papers*. Belgium, Brussels: Association for Computational Linguistics, Oct. 2018, pp. 186–191 (cited on page 93).
- [297] Ricardo Rei, José G. C. de Souza, Duarte Alves, Chrysoula Zerva, Ana C Farinha, Taisiya Glushkova, Alon Lavie, Luisa Coheur, and André F. T. Martins. *COMET-22: Unbabel-IST 2022 Submission for the Metrics Shared Task*. In: *Proceedings of the Seventh Conference on Machine Translation (WMT)*. Ed. by Philipp Koehn, Loic Barrault, Ondřej Bojar, Fethi Bougares, Rajen Chatterjee, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Alexander Fraser, Markus Freitag, Yvette Graham, Roman Grundkiewicz, Paco Guzman, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Tom Kocmi, André Martins, Makoto Morishita, Christof Monz, Masaaki Nagata, Toshiaki Nakazawa, Matteo Negri, Aurélie Névél, Mariana Neves, Martin Popel, Marco Turchi, and Marcos Zampieri. Abu Dhabi, United Arab Emirates (Hybrid): Association for Computational Linguistics, Dec. 2022, pp. 578–585 (cited on page 93).
- [298] NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, Sermarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. *No Language Left Behind: Scaling Human-Centered Machine Translation*. 2022 (cited on pages 93, 97).
- [299] Antonios Anastasopoulos, Alessandro Cattelan, Zi-Yi Dou, Marcello Federico, Christian Federman, Dmitriy Genzel, Francisco Guzmán, Junjie Hu, Macduff Hughes, Philipp Koehn, Rosie Lazar, Will Lewis, Graham Neubig, Mengmeng Niu, Alp Öktem, Eric Paquin, Grace Tang, and Sylwia Tur. *TICO-19: the Translation Initiative for Covid-19*. 2020 (cited on page 93).
- [300] Stella Biderman, Hailey Schoelkopf, Quentin Anthony, Herbie Bradley, Kyle O’Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. *Pythia: A Suite for Analyzing Large Language Models Across Training and Scaling*. 2023 (cited on page 93).
- [301] Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. *TinyLlama: An Open-Source Small Language Model*. 2024 (cited on page 93).

- [302] Antoine Simoulin and Benoit Crabbé. *Un modèle Transformer Génératif Pré-entraîné pour le _____ français*. In: *Traitement Automatique des Langues Naturelles*. Ed. by Pascal Denis, Natalia Grabar, Amel Fraisse, Rémi Cardon, Bernard Jacquemin, Eric Kergosien, and Antonio Balvet. Lille, France: ATALA, 2021, pp. 246–255 (cited on page 93).
- [303] Julien Launay, Elena Tommasone, Baptiste Pannier, François Boniface, Amélie Chatelain, Alessandro Cappelli, Iacopo Poli, and Djamé Seddah. *PAGnol: An Extra-Large French Generative Model*. 2021 (cited on page 93).
- [304] Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. *Finetuned Language Models Are Zero-Shot Learners*. 2022 (cited on page 96).
- [305] Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Zhi Zheng, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. *Enhancing Chat Language Models by Scaling High-quality Instructional Conversations*. In: *arXiv preprint arXiv:2305.14233* (2023) (cited on page 96).
- [306] Wenting Zhao, Xiang Ren, Jack Hessel, Claire Cardie, Yejin Choi, and Yuntian Deng. *(InThe)WildChat: 570K ChatGPT Interaction Logs In The Wild*. In: *The Twelfth International Conference on Learning Representations*. 2024 (cited on page 96).
- [307] Cyrille Delestre. *BloomZ SFT Chat*. 2023. URL: <https://huggingface.co/cmarkea/bloomz-3b-sft-chat> (cited on page 96).
- [308] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. *Survey of Hallucination in Natural Language Generation*. In: *ACM Computing Surveys* 55.12 (Mar. 2023), pp. 1–38. DOI: [10.1145/3571730](https://doi.org/10.1145/3571730) (cited on page 99).
- [309] Nuno M Guerreiro, Duarte M Alves, Jonas Waldendorf, Barry Haddow, Alexandra Birch, Pierre Colombo, and André FT Martins. *Hallucinations in large multilingual translation models*. In: *Transactions of the Association for Computational Linguistics* 11 (2023), pp. 1500–1517 (cited on page 99).
- [310] Gemma Team et al. *Gemma 2: Improving Open Language Models at a Practical Size*. 2024. URL: <https://arxiv.org/abs/2408.00118> (cited on pages 101, 186).
- [311] Marah Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Harkirat Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Caio César Teodoro Mendes, Weizhu Chen, Vishrav Chaudhary, Parul Chopra, Allie Del Giorno, Gustavo de Rosa, Matthew Dixon, Ronen Eldan, Dan Iter, Amit Garg, Abhishek Goswami, Suriya Gunasekar, Emman Haider, Junheng Hao, Russell J. Hewett, Jamie Huynh, Mojan Javaheripi, Xin Jin, Piero Kauffmann, Nikos Karampatziakis, Dongwoo Kim, Mahoud Khademi, Lev Kurilenko, James R. Lee, Yin Tat Lee, Yuanzhi Li, Chen Liang, Weishung Liu, Eric Lin, Zeqi Lin, Piyush Madan, Arindam Mitra, Hardik Modi, Anh Nguyen, Brandon Norick, Barun Patra, Daniel Perez-Becker, Thomas Portet, Reid Pryzant, Heyang Qin, Marko Radmilac,

- Corby Rosset, Sambudha Roy, Olatunji Ruwase, Olli Saarikivi, Amin Saied, Adil Salim, Michael Santacrose, Shital Shah, Ning Shang, Hiteshi Sharma, Xia Song, Masahiro Tanaka, Xin Wang, Rachel Ward, Guanhua Wang, Philipp Witte, Michael Wyatt, Can Xu, Jiahang Xu, Sonali Yadav, Fan Yang, Ziyi Yang, Donghan Yu, Chengruidong Zhang, Cyril Zhang, Jianwen Zhang, Li Lyna Zhang, Yi Zhang, Yue Zhang, Yunan Zhang, and Xiren Zhou. *Phi-3 Technical Report: A Highly Capable Language Model Locally on Your Phone*. Version Number: 2. 2024. DOI: [10.48550/ARXIV.2404.14219](https://arxiv.org/abs/2404.14219). URL: <https://arxiv.org/abs/2404.14219> (visited on 05/21/2024) (cited on pages 101, 102, 161).
- [312] Reka Team, Aitor Ormazabal, Che Zheng, Cyprien de Masson d’Autume, Dani Yogatama, Deyu Fu, Donovan Ong, Eric Chen, Eugenie Lamprecht, Hai Pham, Isaac Ong, Kaloyan Aleksiev, Lei Li, Matthew Henderson, Max Bain, Mikel Artetxe, Nishant Relan, Piotr Padlewski, Qi Liu, Ren Chen, Samuel Phua, Yazheng Yang, Yi Tay, Yuqi Wang, Zhongkai Zhu, and Zhihui Xie. *Reka Core, Flash, and Edge: A Series of Powerful Multimodal Language Models*. 2024. URL: <https://arxiv.org/abs/2404.12387> (cited on page 101).
- [313] An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. *Qwen2 Technical Report*. 2024. URL: <https://arxiv.org/abs/2407.10671> (cited on pages 101, 102).
- [314] Abhimanyu Dubey et al. *The Llama 3 Herd of Models*. 2024. URL: <https://arxiv.org/abs/2407.21783> (cited on pages 101, 102, 186).
- [315] Dirk Groeneveld, Iz Beltagy, Pete Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya Harsh Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, Shane Arora, David Atkinson, Russell Authur, Khyathi Chandu, Arman Cohan, Jennifer Dumas, Yanai Elazar, Yuling Gu, Jack Hessel, Tushar Khot, William Merrill, Jacob Morrison, Niklas Muennighoff, Aakanksha Naik, Crystal Nam, Matthew E. Peters, Valentina Pyatkin, Abhilasha Ravichander, Dustin Schwenk, Saurabh Shah, Will Smith, Nishant Subramani, Mitchell Wortsman, Pradeep Dasigi, Nathan Lambert, Kyle Richardson, Jesse Dodge, Kyle Lo, Luca Soldaini, Noah A. Smith, and Hannaneh Hajishirzi. *OLMo: Accelerating the Science of Language Models*. In: *Preprint* (2024) (cited on page 101).
- [316] Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Alessandro Cappelli, Hamza Alobeidli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. *The RefinedWeb*

Dataset for Falcon LLM: Outperforming Curated Corpora with Web Data, and Web Data Only. In: *arXiv preprint arXiv:2306.01116* (2023) (cited on page 101).

- [317] Loubna Ben Allal, Anton Lozhkov, Guilherme Penedo, Thomas Wolf, and Leandro von Werra. *Cosmopedia*. 2024 (cited on page 101).
- [318] Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. *Scaling Vision Transformers*. Version Number: 2. 2021. DOI: [10.48550/ARXIV.2106.04560](https://arxiv.org/abs/2106.04560). URL: <https://arxiv.org/abs/2106.04560> (visited on 05/19/2024) (cited on page 101).
- [319] Alexander Hägele, Elie Bakouch, Atli Kosson, Loubna Ben Allal, Leandro Von Werra, and Martin Jaggi. *Scaling Laws and Compute-Optimal Training Beyond Fixed Training Durations*. 2024. URL: <https://arxiv.org/abs/2405.18392> (cited on page 101).
- [320] Shu Zhong, Mingyu Xu, Tenglong Ao, and Guang Shi. *Understanding Transformer from the Perspective of Associative Memory*. 2025. URL: <https://arxiv.org/abs/2505.19488> (cited on page 107).
- [321] Sabri Eyuboglu, Ryan Ehrlich, Simran Arora, Neel Guha, Dylan Zinsley, Emily Liu, Will Tennien, Atri Rudra, James Zou, Azalia Mirhoseini, and Christopher Re. *Cartridges: Lightweight and general-purpose long context representations via self-study*. 2025. URL: <https://arxiv.org/abs/2506.06266> (cited on page 108).
- [322] Yuri Kuratov, Mikhail Arkhipov, Aydar Bulatov, and Mikhail Burtsev. *Cramming 1568 Tokens into a Single Vector and Back Again: Exploring the Limits of Embedding Space Capacity*. 2025. URL: <https://arxiv.org/abs/2502.13063> (cited on page 108).
- [323] Xiang Lisa Li and Percy Liang. *Prefix-Tuning: Optimizing Continuous Prompts for Generation*. 2021. URL: <https://arxiv.org/abs/2101.00190> (cited on page 108).
- [324] Alex L. Zhang, Tim Kraska, and Omar Khattab. *Recursive Language Models*. 2026. URL: <https://arxiv.org/abs/2512.24601> (cited on pages 108, 110).
- [325] Jin Xu, Zhifang Guo, Jinzheng He, Hangrui Hu, Ting He, Shuai Bai, Keqin Chen, Jialin Wang, Yang Fan, Kai Dang, Bin Zhang, Xiong Wang, Yunfei Chu, and Junyang Lin. *Qwen2.5-Omni Technical Report*. 2025. URL: <https://arxiv.org/abs/2503.20215> (cited on page 109).
- [326] Mengyao Xu, Wenfei Zhou, Yauhen Babakhin, Gabriel Moreira, Ronay Ak, Radek Osmulski, Bo Liu, Even Oldridge, and Benedikt Schifferer. *Omni-Embed-Nemotron: A Unified Multimodal Retrieval Model for Text, Image, Audio, and Video*. 2025. URL: <https://arxiv.org/abs/2510.03458> (cited on page 109).
- [327] Haonan Chen, Sicheng Gao, Radu Timofte, Tetsuya Sakai, and Zhicheng Dou. *e5-omni: Explicit Cross-modal Alignment for Omni-modal Embeddings*. 2026. URL: <https://arxiv.org/abs/2601.03666> (cited on page 109).
- [328] Nicolas Boizard, Théo Deschamps-Berger, Hippolyte Gisserot-Boukhlef, Céline Hudelot, and Pierre Colombo. *BidirLM: From Text to Omnimodal Bidirectional Encoders by Adapting and Composing Causal LLMs*. 2026. URL: <https://arxiv.org/abs/2604.02045> (cited on page 109).

- [329] Madhuri Shanbhogue, Zhe Li, Shanfeng Zhang, Gustavo Hernández Ábrego, Shih-Cheng Huang, Aashi Jain, Daniel Salz, Sonam Goenka, Chaitra Hegde, Ji Ma, et al. *Gemini Embedding 2: A Native Multimodal Embedding Model from Gemini*. 2026. URL: <https://arxiv.org/abs/2605.27295> (cited on page 109).
- [330] Jason Wei, Zhiqing Sun, Spencer Papay, Scott McKinney, Jeffrey Han, Isa Fulford, Hyung Won Chung, Alex Tachard Passos, William Fedus, and Amelia Glaese. *BrowseComp: A Simple Yet Challenging Benchmark for Browsing Agents*. 2025. URL: <https://arxiv.org/abs/2504.12516> (cited on pages 110, 111).
- [331] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. *ReAct: Synergizing Reasoning and Acting in Language Models*. In: *arXiv preprint arXiv:2210.03629* (2022) (cited on page 110).
- [332] Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A. Smith, and Mike Lewis. *Measuring and Narrowing the Compositionality Gap in Language Models*. In: *arXiv preprint arXiv:2210.03350* (2022) (cited on page 110).
- [333] Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. *Precise Zero-Shot Dense Retrieval without Relevance Labels*. In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*. 2023, pp. 1762–1777 (cited on page 110).
- [334] Bowen Jin, Hansi Zeng, Zhenrui Yue, Dong Wang, Hamed Zamani, and Jiawei Han. *Search-R1: Training LLMs to Reason and Leverage Search Engines with Reinforcement Learning*. In: *arXiv preprint arXiv:2503.09516* (2025) (cited on page 110).
- [335] Huatong Song, Jinhao Jiang, Yingqian Min, Jie Chen, Zhipeng Chen, Wayne Xin Zhao, Lei Fang, and Ji-Rong Wen. *R1-Searcher: Incentivizing the Search Capability in LLMs via Reinforcement Learning*. In: *arXiv preprint arXiv:2503.05592* (2025) (cited on page 110).
- [336] Mingyang Chen, Linzhuang Sun, Tianpeng Li, Haoze Sun, Yijie Zhou, Chenzheng Zhu, Haofen Wang, Jeff Z. Pan, Wen Zhang, Huajun Chen, Fan Yang, Zenan Zhou, and Weipeng Chen. *ReSearch: Learning to Reason with Search for LLMs via Reinforcement Learning*. In: *arXiv preprint arXiv:2503.19470* (2025) (cited on page 110).
- [337] Alex L. Zhang, Tim Kraska, and Omar Khattab. *Recursive Language Models*. In: *arXiv preprint arXiv:2512.24601* (2025) (cited on page 110).
- [338] Diane Tchuindjo, Devavrat Shah, and Omar Khattab. *OBLIQ-Bench: Exposing Overlooked Bottlenecks in Modern Retrievers with Latent and Implicit Queries*. In: *arXiv preprint arXiv:2605.06235* (2026) (cited on pages 110, 111).
- [339] Friedrich A. Hayek. *The Use of Knowledge in Society*. In: *The American Economic Review* 35.4 (1945), pp. 519–530 (cited on page 111).
- [340] Justin Wolfers and Eric Zitzewitz. *Prediction Markets*. In: *Journal of Economic Perspectives* 18.2 (2004), pp. 107–126. DOI: [10.1257/0895330041371321](https://doi.org/10.1257/0895330041371321) (cited on page 111).
- [341] Justin Wolfers and Eric Zitzewitz. *Prediction Markets in Theory and Practice*. Tech. rep. 12083. National Bureau of Economic Research, 2006 (cited on page 111).

- [342] Joyce E. Berg, Forrest D. Nelson, and Thomas A. Rietz. *Prediction Market Accuracy in the Long Run*. In: *International Journal of Forecasting* 24.2 (2008), pp. 285–300 (cited on page 111).
- [343] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. *Attention Is All You Need*. Version Number: 7. 2017. doi: [10.48550/ARXIV.1706.03762](https://arxiv.org/abs/1706.03762). URL: <https://arxiv.org/abs/1706.03762> (visited on 06/12/2024) (cited on page 118).
- [344] Anna Rogers and Alexandra Sasha Luccioni. *Position: Key Claims in LLM Research Have a Long Tail of Footnotes*. 2024. URL: <https://arxiv.org/abs/2308.07120> (cited on page 119).
- [345] Burton H. Bloom. *Space/time trade-offs in hash coding with allowable errors*. In: *Commun. ACM* 13.7 (July 1970). Place: New York, NY, USA Publisher: Association for Computing Machinery, pp. 422–426. doi: [10.1145/362686.362692](https://doi.org/10.1145/362686.362692) (cited on page 126).
- [346] Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. *Vision Transformers Need Registers*. In: (2023). Publisher: [object Object] Version Number: 2. doi: [10.48550/ARXIV.2309.16588](https://arxiv.org/abs/2309.16588). (Visited on 05/10/2024) (cited on page 131).
- [347] Liquid AI. *LFM2 Technical Report*. In: *arXiv preprint arXiv:2511.23404* (2025) (cited on page 141).
- [348] Xing Han Lü. *BM25S: Orders of magnitude faster lexical search via eager sparse scoring*. 2024. URL: <https://arxiv.org/abs/2407.03618> (cited on page 141).
- [349] Nomic Team. *Nomic Embed Multimodal: Interleaved Text, Image, and Screenshots for Visual Document Retrieval*. 2025. URL: <https://nomic.ai/blog/posts/nomic-embed-multimodal> (cited on page 141).
- [350] Rikiya Takehi, Benjamin Clavié, Sean Lee, and Aamir Shakir. *Fantastic (small) Retrievers and How to Train Them: mxbai-edge-colbert-v0 Tech Report*. 2025. URL: <https://arxiv.org/abs/2510.14880> (cited on page 141).
- [351] Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. *ZeRO: Memory Optimizations Toward Training Trillion Parameter Models*. 2020. URL: <https://arxiv.org/abs/1910.02054> (cited on page 159).
- [352] Tianqi Chen, Bing Xu, Chiyuan Zhang, and Carlos Guestrin. *Training Deep Nets with Sublinear Memory Cost*. 2016. URL: <https://arxiv.org/abs/1604.06174> (cited on page 159).
- [353] Nandan Thakur, Crystina Zhang, Xueguang Ma, and Jimmy Lin. *Fixing Data That Hurts Performance: Cascading LLMs to Relabel Hard Negatives for Robust Information Retrieval*. 2025. URL: <https://arxiv.org/abs/2505.16967> (cited on page 161).
- [354] Yi-Lin Sung, Linjie Li, Kevin Lin, Zhe Gan, Mohit Bansal, and Lijuan Wang. *An Empirical Study of Multimodal Model Merging*. 2023. URL: <https://arxiv.org/abs/2304.14933> (cited on page 165).

- [355] Sebastian Dziadzio, Vishaal Udandaraao, Karsten Roth, Ameya Prabhu, Zeynep Akata, Samuel Albanie, and Matthias Bethge. *How to Merge Your Multimodal Models Over Time?* 2024. URL: <https://arxiv.org/abs/2412.06712> (cited on page 165).
- [356] Gabriel Ilharco, Mitchell Wortsman, Samir Yitzhak Gadre, Shuran Song, Hannaneh Hajishirzi, Simon Kornblith, Ali Farhadi, and Ludwig Schmidt. *Patching open-vocabulary models by interpolating weights*. 2022. URL: <https://arxiv.org/abs/2208.05592> (cited on page 165).
- [357] Ken Shoemake. *Animating rotation with quaternion curves*. In: 19.3 (July 1985), pp. 245–254. doi: [10.1145/325165.325242](https://doi.org/10.1145/325165.325242) (cited on page 165).
- [358] Antoine Chaffin and Raphael Sourty. *PyLate: Flexible Training and Retrieval for Late Interaction Models*. 2024. URL: <https://github.com/lightonai/pylate> (cited on page 174).
- [359] Leo Gao, Jonathan Tow, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Kyle McDonnell, Niklas Muennighoff, Jason Phang, Laria Reynolds, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. *A framework for few-shot language model evaluation*. Version v0.0.1. Sept. 2021. doi: [10.5281/zenodo.5371628](https://doi.org/10.5281/zenodo.5371628) (cited on page 185).
- [360] Nikita Nangia, Clara Vania, Rasika Bhalerao, and Samuel R. Bowman. *CrowS-Pairs: A Challenge Dataset for Measuring Social Biases in Masked Language Models*. 2020 (cited on page 186).
- [361] Aurélie Névél, Yoann Dupont, Julien Bezançon, and Karën Fort. *French CrowS-Pairs: Extension à une langue autre que l’anglais d’un corpus de mesure des biais sociétaux dans les modèles de langue masqués (French CrowS-Pairs : Extending a challenge dataset for measuring social bias in masked language models to a language other than English)*. French. In: *Actes de la 29e Conférence sur le Traitement Automatique des Langues Naturelles. Volume 1 : conférence principale*. Ed. by Yannick Estève, Tania Jiménez, Titouan Parcollet, and Marcelly Zanon Boito. Avignon, France: ATALA, June 2022, pp. 355–364 (cited on page 186).
- [362] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. *Measuring Massive Multitask Language Understanding*. Version Number: 3. 2020. doi: [10.48550/ARXIV.2009.03300](https://doi.org/10.48550/ARXIV.2009.03300). URL: <https://arxiv.org/abs/2009.03300> (visited on 06/04/2024) (cited on page 186).