Know When to Fuse: Investigating Non-English Hybrid Retrieval in the Legal Domain

Antoine Louis, Gijs van Dijck, Gerasimos Spanakis
Law & Tech Lab, Maastricht University
{a.louis, gijs.vandijck, jerry.spanakis}@maastrichtuniversity.nl

Abstract

Hybrid search has emerged as an effective strategy to offset the limitations of different matching paradigms, especially in out-of-domain contexts where notable improvements in retrieval quality have been observed. However, existing research predominantly focuses on a limited set of retrieval methods, evaluated in pairs on domain-general datasets exclusively in English. In this work, we study the efficacy of hybrid search across a variety of prominent retrieval models within the unexplored field of law in the French language, assessing both zero-shot and in-domain scenarios. Our findings reveal that in a zero-shot context, fusing different domain-general models consistently enhances performance compared to using a standalone model, regardless of the fusion method. Surprisingly, when models are trained in-domain, we find that fusion generally diminishes performance relative to using the best single system, unless fusing scores with carefully tuned weights. These novel insights, among others, expand the applicability of prior findings across a new field and language, and contribute to a deeper understanding of hybrid search in non-English specialized domains.[1]

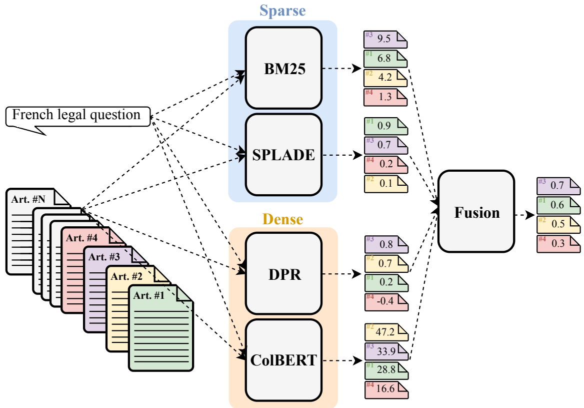


Figure 1: A high-level illustration of the hybrid search workflow based on various sparse and dense retrievers.

1 Introduction

Information retrieval is typically addressed through one of two fundamental matching paradigms: (i) *lexical matching*, which relies on an exact match of terms between queries and documents; and (ii) *semantic matching*, which measures complex relationships between words to capture underlying semantics. Lexical matching is simple, efficient, and generally effective across various domains (Thakur et al., 2021). However, it suffers from the vocabulary gap issue (Berger et al., 2000), where relevant information might not explicitly include query terms yet still fulfills the actual informational needs. Semantic models remedy vocabulary mismatches by learning to model semantic similarity, resulting in significant in-domain performance gains (Qu et al., 2021; Xiong et al., 2021; Hofstätter et al., 2021). Nevertheless, these models tend to exhibit limited generalization across unseen topics (Thakur et al., 2021), which is particularly problematic in highly specialized domains, like law, where high-quality labeled data is both scarce and costly.

Recent works suggest that combining these two paradigms can enhance retrieval quality (Kuzi et al., 2020; Wang et al., 2021; Ma et al., 2021), particularly in out-of-distribution settings (Chen et al., 2022; Bruch et al., 2024), as they tend to mitigate each other's limitations. However, these efforts have mostly been limited to combining no more than two systems – typically pairing BM25 (Robertson et al., 1994) with single-vector dense bi-encoders (Reimers and Gurevych, 2019) – while constraining evaluation to English datasets only.

Our work aims to extend this scope by investigating the potential synergies among a broader range of retrieval models, encompassing both sparse and dense methods, specifically within the uncharted *legal* domain in the *French* language, as illustrated

[1] Our source code and models are available at https://github.com/maastrichtlawtech/fusion.

Proceedings of the 31st International Conference on Computational Linguistics, pages 4293–4312
January 19–24, 2025.

in Figure 1. Our contributions are threefold:

- First, we investigate the efficacy of combining diverse domain-general retrieval models for legal retrieval, assuming *no* domain-specific labeled data is available – a highly usual scenario in specialized domains like law.
- Second, we explore the extent to which specialized retrievers and their fusion can impact in-domain performance, assuming *limited* domain-specific training data is available.
- Finally, we release all our learned retrievers, including the first French SPLADE and ColBERT models for general and legal domains.

2 Methodology

Assuming that different matching paradigms may be complementary in how they model relevance (Chen et al., 2022; Bruch et al., 2024), we aim to explore the potential of combining various systems to enhance performance on French legal retrieval. In this section, we outline the retrieval models (§2.1), fusion techniques (§2.2), and experimental setup (§2.3) employed in our study, with additional comprehensive details available in Appendix A.

2.1 Retrieval Models

We select several prominent retrieval methods representing diverse matching paradigms, all demonstrating high effectiveness in prior studies. Specifically, we explore the unsupervised BM25 weighting scheme (Robertson et al., 1994), our own *single-vector dense* (Lee et al., 2019; Chang et al., 2020; Karpukhin et al., 2020), *multi-vector dense* (Khattab and Zaharia, 2020; Santhanam et al., 2022b), and *single-vector sparse* (Formal et al., 2021a,b) bi-encoder models – respectively dubbed DPR_{FR}, $\text{ColBERT}_{\text{FR}}$, and $\text{SPLADE}_{\text{FR}}$ – and a *cross-attention* model (Nogueira and Cho, 2019; Han et al., 2020; Gao et al., 2021a) termed $\text{monoBERT}_{\text{FR}}$. Following a preliminary comparative analysis of various pretrained French language models in Appendix B.1, we choose $\text{CamemBERT}_{\text{BASE}}$ (Martin et al., 2020) as the backbone encoder for all our supervised neural retrievers. We refer readers to Appendix A.1 for detailed explanations of each method's relevance matching and optimization processes.

2.2 Fusion Techniques

To leverage existing retrieval methods without modifications, our study explores *late* fusion techniques, which aggregate results post-prediction – in contrast to early fusion methods that merge latent representations of distinct retrievers within the feature space prior to making predictions. In this context, the relevance of a candidate can be assessed using two main measures: its position in the ranked list or its predicted score. This distinction underpins the two primary late fusion approaches explored in this study: *score-based* and *rank-based* fusion. Specifically, we investigate *normalized score fusion* (NSF; Lee, 1995) with various scaling techniques, *Borda count fusion* (BCF; Ho et al., 1994), and *reciprocal rank fusion* (RRF; Cormack et al., 2009). See Appendix A.2 for detailed definitions of each method.

2.3 Experimental Setup

Datasets. We exploit two French text ranking datasets: the domain-general mMARCO-fr (Bonifacio et al., 2021) and the domain-specific LLeQA (Louis et al., 2024). The former is a translated version of MS MARCO (Nguyen et al., 2018) in 13 languages, including French. It comprises a corpus of 8.8M passages, 539K training queries, and 6980 development queries. LLeQA targets long-form question answering and information retrieval within the legal domain. It consists of 1,868 French-native questions on various legal topics, distributed across training (1472), development (201), and test (195) sets. Each question is expertly annotated with references to relevant legal provisions drawn from a corpus of 27,942 Belgian law articles.

Evaluation metrics. To measure effectiveness, we use official metrics for each dataset: mean reciprocal rank at cutoff 10 (MRR@10) for mMARCO, and average r-precision (RP) for LLeQA. Both metrics are rank-aware, meaning they are sensitive to variations in the ordering of retrieved results. Additionally, we report the rank-unaware recall measure at various cutoffs (R@k), which is particularly useful for assessing performance of first-stage retrievers. See Appendix A.3 for details.

Baselines. We evaluate our learned retrievers and their hybrid configurations against leading open-source multilingual retrieval models, including BM25 (Robertson et al., 1994), mE5 (Wang et al., 2024) in its small, base, and large variants, and BGE-M3 (Chen et al., 2024) in its dense version.

2.4 Efficiency

To evaluate the practicality of each system for real-world deployment, we assess their computational and memory efficiency during inference.

	Model	mMARCO-fr		Model Size		#Samples		Batch Size		Hardware	
		MRR@10	R@500	#Params	RAM	PF	F	PF	F	Pre-Finetune	Finetune
Baselines											
1	BM25 ($k1$=0.9, b=0.4)	0.143	0.681	–	–	–	–	–	–	–	–
2	mE5$_{\text{SMALL}}$	0.297	0.908	117.7M	0.5GB	1B	1.6M	32k	512	32×V100	8×V100
3	mE5$_{\text{BASE}}$	0.303	<u>0.914</u>	278.0M	1.1GB	1B	1.6M	32k	512	64×V100	8×V100
4	mE5$_{\text{LARGE}}$	<u>0.311</u>	0.909	559.9M	2.2GB	1B	1.6M	32k	512	*Unk.*	8×V100
5	BGE-M3$_{\text{DENSE}}$	0.270	0.891	567.8M	2.3GB	1.2B	1.6M	67k	1.2k	96×A800	24×A800
Learned models (ours)											
6	DPR$_{\text{FR-BASE}}$	0.285	0.891	110.6M	0.4GB	–	0.5M	–	152	–	1×V100
7	SPLADE$_{\text{FR-BASE}}$	0.247	0.860	110.6M	0.4GB	–	0.5M	–	128	–	1×H100
8	ColBERT$_{\text{FR-BASE}}$	0.295†	0.884†	110.6M	0.4GB	–	0.5M	–	128	–	1×H100
9	monoBERT$_{\text{FR-BASE}}$	**0.334***⋆	**0.965**⋆	110.6M	0.4GB	–	0.5M	–	128	–	1×H100

† Evaluated using the PLAID retrieval engine (Santhanam et al., 2022a). ⋆ Evaluated by re-ranking 1k candidates including gold and hard negative passages.

Table 1: Retrieval results on mMARCO-fr small dev set (in-domain). We report each model's training resources.

Index size. We start by calculating the storage footprint of the indexed LLeQA articles, precomputed offline and loaded at inference, noting that the indexing method varies with the retrieval approach. Sparse methods like BM25 and SPLADE use inverted indexes, which store each vocabulary term along lists of articles containing the term and its frequency within those articles. Single-vector dense models, such as DPR$_{\text{FR}}$, mE5, and BGE-M3, rely on flat indexes for brute-force search, sequentially storing vectors on $d \times b \times |\mathcal{C}|$ bits given d-dimensional representations of articles from corpus $\mathcal{C}$ encoded in b bits (with b=32 in our study).[2] Meanwhile, ColBERT uses an advanced centroid-based indexing to store late-interaction token embeddings, with a footprint comparable to dense flat indexes (Santhanam et al., 2022b).

Retrieval latency. We then measure the retrieval latency per query in seconds. We use a query batch size of one to simulate streaming queries and compute the average latency across all queries in the LLeQA dev set. Measurements are conducted on a single NVIDIA H100 for GPU search and an AMD EPYC 7763 for CPU search.

Inference FLOPs. Finally, we estimate the number of floating point operations (FLOPs) per query as a hardware-agnostic measure of compute usage. Details of our estimation methodology across the different systems are provided in Appendix A.4.

3 Zero-Shot Evaluation

In this section, we investigate the out-of-domain generalization capabilities of modern retrieval models trained on a budget and explore the efficacy of their fusion in the specialized domain of law. Specifically, we explore the following question: *Assuming a lack of domain-specific labeled data and limited computational resources, how effectively can hybrid combinations of domain-general retrieval models perform within the legal domain?* To address this, we train the supervised retrieval models presented in Section 2 on the French segment of the domain-general mMARCO dataset. We denote the resulting models with the FR-BASE subscript throughout the rest of the paper.

Main results. When evaluated on mMARCO-fr, our learned French retrievers exhibit competitive, and at times superior, in-domain performance compared to leading multilingual retrieval models. This is particularly notable given their relatively smaller size and the constrained resources used during training, as shown in Table 1. For instance, DPR$_{\text{FR-BASE}}$ surpasses BGE-M3$_{\text{DENSE}}$ with only one-fifth of its parameters, 2400× fewer training samples, and significantly less training compute. Additionally, our cross-encoder consistently outperforms all other retrieval methods, corroborating prior findings on the efficacy of cross-attention (Hofstätter et al., 2020). However, results in Table 2 reveal that, when evaluated in the legal domain, our domain-general French retrievers generally underperform against the multilingual baselines, except for our cross-encoder which remains competitive at smaller cutoffs. This discrepancy is largely due to the baselines' extensive (pre-)finetuning across diverse data with large batch sizes – which proved beneficial for enhanced contrastive learning (Qu et al., 2021). Surprisingly, BM25 outperforms all neural models in this specialized context, reaffirming its robustness when dealing with out-of-distribution data.

[2]While ANNS indexes such as HNSW (Malkov et al., 2014) enable more efficient retrieval, they introduce significant overhead which makes flat indexes generally preferable for smaller datasets like LLeQA (Milvus, 2022; Redis, 2024).

	Model	LLeQA			Index Storage		Latency (s/q)		FLOPs
		RP	R@10	R@500	Disk♦	Ratio♣	GPU	CPU	
Baselines									
1	BM25 ($k1{=}2.5, b{=}0.2$)	**0.163**	0.367	0.672	6.6MB	×0.2	–	0.142	1.7e+6
2	mE5$_{\text{SMALL}}$	0.081	0.174	0.611	40.9MB	×1.5	0.013	0.028	6.6e+8
3	mE5$_{\text{BASE}}$	0.074	0.157	0.653	81.9MB	×2.9	0.014	0.065	2.6e+9
4	mE5$_{\text{LARGE}}$	0.074	0.194	0.695	109.1MB	×3.9	0.022	0.121	9.2e+9
5	BGE-M3$_{\text{DENSE}}$	0.090	0.325	0.734	109.1MB	×3.9	0.023	0.113	9.2e+9
Learned models (ours)									
6	DPR$_{\text{FR-BASE}}$	0.046	0.146	0.590	81.9MB	×2.9	0.013	0.057	2.6e+9
7	SPLADE$_{\text{FR-BASE}}$	0.045	0.107	0.596	30.2MB	×1.1	0.013	0.609	2.6e+9
8	ColBERT$_{\text{FR-BASE}}$	0.047†	0.148†	0.517†	185.8MB†	×6.7	0.031†	0.142†	2.6e+11
9	monoBERT$_{\text{FR-BASE}}$	0.102	0.290	0.536	–	–	4.472*	184.7*	2.2e+13*
Hybrid combinations									
10	NSF$_{\text{Z-SCORE}}(1,7)$	0.130	0.372	**0.755**	36.8MB	×1.3	–	–	2.6e+9
11	NSF$_{\text{MIN-MAX}}(1,8)$	0.134	0.397	0.746	192.4MB	×6.9	–	–	2.6e+11
12	NSF$_{\text{Z-SCORE}}(1,6,7)$	0.092	0.354	0.742	118.7MB	×4.3	–	–	5.2e+9
13	NSF$_{\text{Z-SCORE}}(1,7,8)$	0.109	<u>0.399</u>	<u>0.753</u>	222.6MB	×8.0	–	–	5.2e+9
14	NSF$_{\text{Z-SCORE}}(1,6,8)$	<u>0.139</u>	**0.407**	0.750	274.3MB	×9.8	–	–	2.6e+11
15	NSF$_{\text{Z-SCORE}}(1,6,7,8)$	0.125	0.388	0.736	304.5MB	×10.9	–	–	2.7e+11

♦ Estimated with 32-bit precision for dense vectors. ♣ Ratio of index size to plain text size.

Table 2: Retrieval results on LLeQA test set (zero-shot). We report performance of the best hybrid configurations obtained after extensive evaluation on LLeQA dev set (see Table 3).

Besides, BM25 is notably efficient at inference, with an index up to 30× smaller and significantly fewer FLOPs than neural retrievers. In contrast, the full interaction mechanism of monoBERT$_{\text{FR-BASE}}$ incurs substantial computational costs, resulting in latencies up to 350× and 2350× higher on GPU and CPU, respectively, than the other learned French models – while assessed to re-rank 1,000 candidates only rather than the whole corpus. ColBERT$_{\text{FR-BASE}}$, with its token-to-token interaction, achieves reasonable latencies on both GPU and CPU due to the low-level optimization of PLAID, but results in a larger index. Meanwhile, SPLADE$_{\text{FR-BASE}}$ stands out among neural methods by using an inverted index nearly 3× smaller than that of its single-vector dense counterpart.

Finally, we observe that fusing BM25 with one or more of our learned domain-general French models consistently and significantly outperforms all individual retrievers in the zero-shot setting (except on RP where BM25 excels) yet at the expense of increased memory – but comparable latencies when using parallelization. This fusion markedly enhances recall at large cutoffs compared to standalone BM25. On recall@10, most fusions improve upon BM25; notably, the BM25+DPR$_{\text{FR-BASE}}$+ColBERT$_{\text{FR-BASE}}$ fusion shows a 4% enhancement and surpasses both DPR$_{\text{FR-BASE}}$ and ColBERT$_{\text{FR-BASE}}$ by around 26%. Surprisingly, the BM25+SPLADE$_{\text{FR-BASE}}$ fusion is the most effective on R@500 while standing out for its efficiency due to both methods' use of inverted indexes.

How do score distributions vary across models? Figure 2 depicts the score distributions of end-to-end retrievers, normalized using both traditional techniques and our proposed percentile normalization. We find that traditional scaling methods lead to misaligned distributions among retrievers, particularly under min-max scaling. Such misalignment impacts score fusion as identical scores may convey different levels of relevance across systems. For example, a min-max normalized score of 0.35 approximates the median for DPR$_{\text{FR-BASE}}$, but corresponds to the 95th percentile for BM25. When these scores are equally combined, the higher relevance indicated by BM25's score is therefore negated. To address this, we explore a new scaling approach that maps scores to their respective percentiles within each system's overall score distribution, estimated using around 5.6 million data points per system. This way, a score of 0.35 is adjusted to 0.5 for DPR$_{\text{FR-BASE}}$ and 0.95 for BM25, leading to a relatively higher fused score that favors high relevance signals. This method requires pre-computing each retriever's score distribution, ideally with a volume matching the corpus size to avoid score collisions. Despite its intuitive appeal, our empirical findings reveal that this percentile-based scaling does not surpass traditional methods, as shown in Table 3.

How complementary are distinct retrievers? We select the two systems that showed the best hybrid sparse-dense performance in Table 3, namely BM25+ColBERT$_{\text{FR-BASE}}$, and analyze their min-max scaled scores across 18.6K query-article pairs from

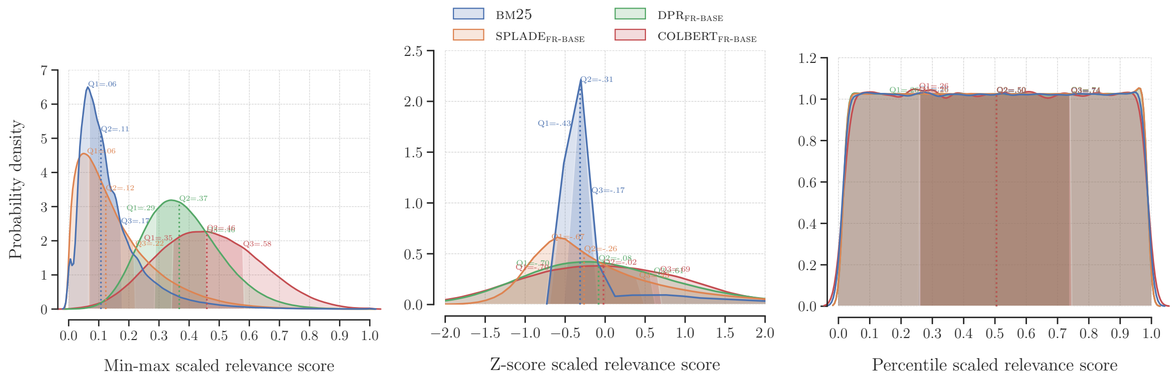


Figure 2: Score distributions of domain-general end-to-end retrievers, normalized using min-max, z-score, and percentile scaling. The distributions are derived from ranking all 27,942 articles in LLeQA's knowledge corpus against the 201 development set queries, resulting in approximately 5.6 million scores per system.

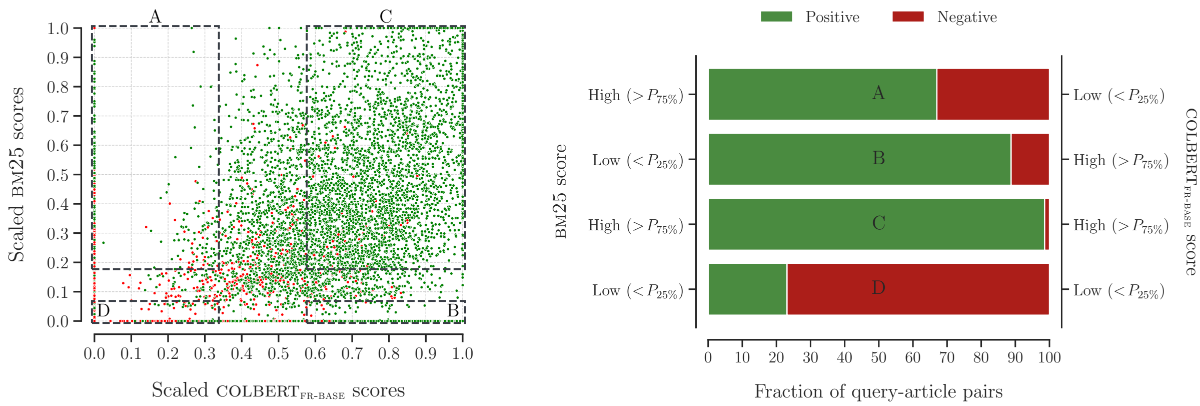


Figure 3: Illustration of the complementary relationship between a sparse (BM25) and a dense (ColBERT$_{\text{FR-BASE}}$) system on out-of-distribution data. Scores have been min-max normalized and categorized into four distinct regions based on each system's global distribution, depicted in Figure 2.

LLeQA, balanced between positive and negative instances. We examine four scenarios: (A) BM25 scores high (above the third quartile of its distribution, depicted in Figure 2) while ColBERT$_{\text{FR-BASE}}$ scores low (below the first quartile of its distribution); (B) BM25 scores low while ColBERT$_{\text{FR-BASE}}$ scores high; (C) both systems score high; (D) both systems score low. Our findings, shown in Figure 3, reveal that when one system scores high while the other does not, the higher-scoring system generally provides the correct signal, effectively compensating for the other's error. Conversely, when both systems concur on the relevance assessment, whether high or low, they are predominantly correct.

Does fusion always help for OOD data? We conduct an exhaustive evaluation across all possible combinations of our learned retrievers (excluding the monoBERT$_{\text{FR-BASE}}$ re-ranker due to its high inefficiency for end-to-end retrieval) and BM25, using the fusion methods presented in Section 2. For NSF, we test both conventional min-max and z-score scaling, as well as our proposed percentile normalization, with either equal or tuned weights. This results in a total of 88 different configurations, whose results are presented in Table 3. Of these, we find that 72 (i.e., 82%) improve performance compared to using the retrievers from the respective combinations individually. Remarkably, nine combinations outperform the extensively trained BGE-M3$_{\text{DENSE}}$ model, which demonstrates the best individual performance by far on LLeQA dev set. Overall, our findings indicate that fusion *almost* always enhances performance on out-of-distribution data, regardless of the fusion technique or normalization approach used – though tuned NSF with z-score scaling seems to deliver optimal results.

4 In-Domain Evaluation

We now investigate the performance enhancement given by specialized retrievers trained in the le-

	Method	BCF	RRF	NSF$_{\text{MIN-MAX}}$		NSF$_{\text{Z-SCORE}}$		NSF$_{\text{PERCENTILE}}$	
				Equal	Tuned	Equal	Tuned	Equal	Tuned
Single baselines	BM25	0.232	0.232	0.232	0.232	0.232	0.232	0.232	0.232
	DPR$_{\text{FR-BASE}}$	0.184	0.184	0.184	0.184	0.184	0.184	0.184	0.184
	SPLADE$_{\text{FR-BASE}}$	0.180	0.180	0.180	0.180	0.180	0.180	0.180	0.180
	ColBERT$_{\text{FR-BASE}}$	0.232	0.232	0.232	0.232	0.232	0.232	0.232	0.232
Sparse / dense	BM25 + SPLADE$_{\text{FR-BASE}}$	0.262	0.279	0.295	0.295	0.286	0.300†	0.282	0.286
	DPR$_{\text{FR-BASE}}$ + ColBERT$_{\text{FR-BASE}}$	0.219	0.230	0.229	0.243	0.227	0.243	0.206	0.228
Dense+sparse w. 2 systems	BM25 + DPR$_{\text{FR-BASE}}$	0.233	0.262	0.268	0.276	0.265	0.286	0.257	0.257
	BM25 + ColBERT$_{\text{FR-BASE}}$	0.249	0.269	0.293	0.303†	0.262	0.294	0.261	0.266
	SPLADE$_{\text{FR-BASE}}$ + DPR$_{\text{FR-BASE}}$	0.188	0.203	0.196	0.217	0.197	0.218	0.195	0.210
	SPLADE$_{\text{FR-BASE}}$ + ColBERT$_{\text{FR-BASE}}$	0.238	0.220	0.225	0.249	0.229	0.243	0.229	0.234
Dense+sparse w. 3 systems	BM25 + SPLADE$_{\text{FR-BASE}}$ + DPR$_{\text{FR-BASE}}$	0.228	0.267	0.297	0.301†	0.296	0.310†	0.263	0.287
	BM25 + SPLADE$_{\text{FR-BASE}}$ + ColBERT$_{\text{FR-BASE}}$	0.260	0.281	0.308†	0.308†	0.300†	0.314†	0.266	0.282
	BM25 + DPR$_{\text{FR-BASE}}$ + ColBERT$_{\text{FR-BASE}}$	0.238	0.289	0.302†	0.308†	0.287	0.314†	0.257	0.263
	SPLADE$_{\text{FR-BASE}}$ + DPR$_{\text{FR-BASE}}$ + ColBERT$_{\text{FR-BASE}}$	0.226	0.232	0.229	0.250	0.229	0.249	0.212	0.233
All	BM25 + SPLADE$_{\text{FR-BASE}}$ + DPR$_{\text{FR-BASE}}$ + ColBERT$_{\text{FR-BASE}}$	0.254	0.275	0.307†	0.315†	0.300†	0.323†	0.260	0.277

Table 3: Out-of-domain recall@10 results on LLeQA dev set. We report performance of normalized score fusion using both equal and tuned weights between systems. Hybrid combinations that improve over each of their constituent systems are highlighted in green, while those that underperform compared to one or more of their systems are marked in red. † indicates competitive performance with state-of-the-art BGE-M3$_{\text{DENSE}}$ (30.6% R@10).

gal domain and assess the effectiveness of fusion techniques in this in-domain context. Specifically, we explore the following question: *Assuming a limited amount of domain-specific labeled data, to what extent can specialized retrievers and their fusion enhance performance within the legal domain?* To address this question, we fine-tune our domain-general neural retrievers, initially trained on mMARCO-fr, on the 1.5K training questions from LLeQA. We denote the resulting models with the FR-LEX subscript in the remainder of the paper.

Main results. Table 4 presents the in-domain performance of our specialized retrieval models. In line with previous findings (Karpukhin et al., 2020; Khattab and Zaharia, 2020; Formal et al., 2021b; Nogueira et al., 2019), we note substantial improvements across all models compared to the zero-shot setting, with each now significantly outperforming the robust BM25 baseline. Interestingly, our single-vector dense retriever, DPR$_{\text{FR-LEX}}$, surpasses all the other approaches, including the more computationally demanding monoBERT$_{\text{FR-LEX}}$ cross-encoder on smaller recall cutoffs. These results underscore the effectiveness of neural methods when trained in-domain, even with relatively limited sample sizes.

Is task-adaptive pre-finetuning beneficial? Here, we study the hypothesis that performing an intermediary finetuning step on a task-related dataset before finetuning on the target dataset can help enhance downstream performance (Dai and Callan, 2019; Li et al., 2020), especially when training samples in the target domain are scarce (Zhang et al., 2020). We therefore compare two learning strategies: the first directly finetunes the pretrained CamemBERT backbone on the specialized LLeQA dataset, while the second (which we adopted as our default approach) incorporates a pre-finetuning step on the domain-general mMARCO-fr dataset. We find this intermediary phase to consistently improve in-domain performance at higher recall cutoffs across all bi-encoder models, as shown in Table 5. However, at lower recall cutoffs, pre-finetuning benefits dense bi-encoders only, with SPLADE$_{\text{FR-LEX}}$ experiencing diminished performance. This strategy does not appear to yield improvements for the monoBERT$_{\text{FR-LEX}}$ cross-encoder.

Does fusion still help with specialized retrievers? Table 6 highlights the in-domain performance of hybrid combinations previously assessed in a zero-shot setting. We now observe a very distinct pattern: around 70% of these combinations lead to deteriorated performance compared to using one of their constituent systems only. Among the 27 (out of 88) configurations that do show improvement, 23 leverage NSF with weights tuned in-domain, while only four combinations (i.e., 5% in total) achieve superior performance without prior tuning. Furthermore, the performance gap between individual systems and their hybrid combinations is considerably narrower within this in-domain context. While a two-system hybrid fusion can yield up to a 7.1%

	Model	R@1k	R@500	R@100	R@10	RP
Dev	BM25	0.634	0.577	0.457	0.232	0.122
	SPLADE$_{\text{FR-LEX}}$	0.925	0.889	0.792	0.535	0.334
	DPR$_{\text{FR-LEX}}$	0.948	0.927	**0.855**	**0.595**	**0.462**
	ColBERT$_{\text{FR-LEX}}$	0.892	0.852	0.747	0.434	0.255
	monoBERT$_{\text{FR-LEX}}$	**0.967**	**0.942**	0.805	0.430	0.219
Test	BM25	0.742	0.672	0.537	0.367	0.163
	SPLADE$_{\text{FR-LEX}}$	0.903	0.857	0.687	0.434	0.102
	DPR$_{\text{FR-LEX}}$	0.937	0.916	**0.801**	**0.558**	**0.244**
	ColBERT$_{\text{FR-LEX}}$	0.841	0.800	0.679	0.432	0.125
	monoBERT$_{\text{FR-LEX}}$	**0.980**	**0.939**	0.746	0.473	0.143

Table 4: In-domain performance on LLeQA dev and test sets. We train each model five times with different seeds and report the best based on the dev set results.

Model	Recall at cut-off k		Δ Avg.
	@1000	@500	
DPR$_{\text{FR-LEX}}$	0.925 / 0.933	0.888 / 0.905	+1.3%
SPLADE$_{\text{FR-LEX}}$	0.863 / 0.878	0.817 / 0.821	+1.0%
ColBERT$_{\text{FR-LEX}}$	0.806 / 0.835	0.777 / 0.806	+2.9%
monoBERT$_{\text{FR-LEX}}$	0.967 / 0.967	0.928 / 0.927	-0.1%
	@50	@10	
DPR$_{\text{FR-LEX}}$	0.685 / 0.706	0.526 / 0.541	+1.8%
SPLADE$_{\text{FR-LEX}}$	0.617 / 0.596	0.402 / 0.403	-1.0%
ColBERT$_{\text{FR-LEX}}$	0.593 / 0.599	0.388 / 0.416	+1.7%
monoBERT$_{\text{FR-LEX}}$	0.632 / 0.629	0.353 / 0.335	-1.2%

Table 5: In-domain recall@k performances on LLeQA test set *without* / *with* pre-finetuning on mMARCO-fr. We report the means across 5 runs with different seeds.

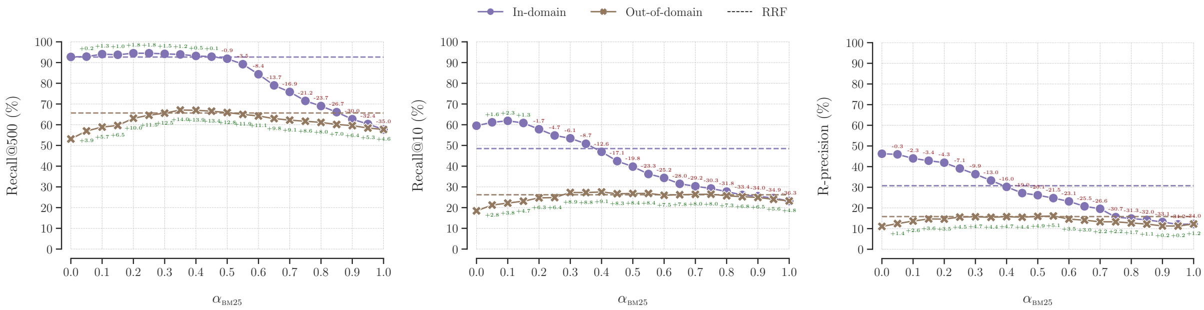


Figure 4: Effect of weight tuning in normalized score fusion between BM25 and DPR$_{\text{FR-\{LEX,BASE\}}}$ on LLeQA dev set.

R@10 improvement over the best single system in zero-shot scenarios, this enhancement does not exceed 1.4% once the models are trained in-domain. Appendix C.1 further discusses that degradation.

How does α in paired NSF affect performance? Finally, we evaluate the impact of weight tuning on the in-domain performance of NSF in a paired configuration, where one system is assigned a weight α and the other $1-\alpha$. We select the best performing two-system combination from Table 6, i.e., BM25+DPR$_{\text{FR-LEX}}$. For comparison, we also report performance of this combination in a zero-shot context and that of RRF in both scenarios, as depicted in Figure 4. We find that integrating BM25 offers minimal benefits once DPR$_{\text{FR}}$ is domain-tuned, with equal weighting between both systems consistently leading to worse performance. This finding contrasts starkly with the out-of-distribution setting, where combining both systems consistently improves performance compared to using one of them alone, regardless of the α weight assigned.

5 Related Work

Statute law retrieval. Returning the relevant legislation to a short legal question is notably challenging due to the linguistic disparity between the specialized jargon of legal statutes (Charrow and Crandall, 1978) and the plain language typically used by laypeople. Research on statute retrieval has traditionally focused on text-level similarity between queries and candidate documents, with earlier methods employing lexical approaches such as TF-IDF (Kim and Goebel, 2017; Dang et al., 2019) or BM25 (Wehnert et al., 2019; Gain et al., 2021). With advancements in representation learning techniques (Vaswani et al., 2017; Devlin et al., 2019), attention has shifted towards dense retrieval to enhance semantic matching capabilities. For instance, Louis and Spanakis (2022) demonstrate that supervised single-vector dense bi-encoders significantly outperform TF-IDF weighting schemes. Su et al. (2024) explore various dense bi-encoder models trained on different domains and reached similar conclusions. Santosh et al. (2024) further push performance of dense bi-encoders by introducing a dynamic negative sampling strategy tailored to law. In parallel, some studies have begun incorporating legal knowledge into the retrieval process. For example, Louis et al. (2023) propose a graph-augmented dense retriever that uses the topological

Method	BCF	RRF	NSF$_{\text{MIN-MAX}}$		NSF$_{\text{Z-SCORE}}$		NSF$_{\text{PERCENTILE}}$	
			Equal	Tuned	Equal	Tuned	Equal	Tuned
BM25	0.232	0.232	0.232	0.232	0.232	0.232	0.232	0.232
DPR$_{\text{FR-LEX}}$	0.595	0.595	0.595	0.595	0.595	0.595	0.595	0.595
SPLADE$_{\text{FR-LEX}}$	0.535	0.535	0.535	0.535	0.535	0.535	0.535	0.535
ColBERT$_{\text{FR-LEX}}$	0.434	0.434	0.434	0.434	0.434	0.434	0.434	0.434
BM25 + SPLADE$_{\text{FR-LEX}}$	0.385	0.457	0.417	0.570	0.350	0.561	0.369	0.450
DPR$_{\text{FR-LEX}}$ + ColBERT$_{\text{FR-LEX}}$	0.546	0.541	0.577	0.609†	0.592	0.608†	0.464	0.555
BM25 + DPR$_{\text{FR-LEX}}$	0.391	0.485	0.398	0.619†	0.326	0.618†	0.351	0.452
BM25 + ColBERT$_{\text{FR-LEX}}$	0.363	0.412	0.360	0.470	0.288	0.473	0.383	0.437
SPLADE$_{\text{FR-LEX}}$ + DPR$_{\text{FR-LEX}}$	0.573	0.586	0.582	0.613†	0.586	0.612†	0.587	0.604
SPLADE$_{\text{FR-LEX}}$ + ColBERT$_{\text{FR-LEX}}$	0.514	0.509	0.537	0.557	0.543	0.553	0.464	0.519
BM25 + SPLADE$_{\text{FR-LEX}}$ + DPR$_{\text{FR-LEX}}$	0.431	0.606†	0.533	0.629†	0.447	0.625†	0.395	0.472
BM25 + SPLADE$_{\text{FR-LEX}}$ + ColBERT$_{\text{FR-LEX}}$	0.427	0.535	0.505	0.575	0.402	0.578	0.412	0.475
BM25 + DPR$_{\text{FR-LEX}}$ + ColBERT$_{\text{FR-LEX}}$	0.429	0.564	0.481	0.624†	0.372	0.623†	0.402	0.468
SPLADE$_{\text{FR-LEX}}$ + DPR$_{\text{FR-LEX}}$ + ColBERT$_{\text{FR-LEX}}$	0.548	0.579	0.579	0.617†	0.587	0.620†	0.480	0.560
BM25 + SPLADE$_{\text{FR-LEX}}$ + DPR$_{\text{FR-LEX}}$ + ColBERT$_{\text{FR-LEX}}$	0.457	0.603†	0.561	0.628†	0.485	0.627†	0.418	0.477

Table 6: In-domain recall@10 results on LLeQA dev set. The red region highlights hybrid combinations that perform worse than one or more of their systems, while the green region emphasizes combinations that outperform each of their constituent systems. † indicates improved performance over DPR$_{\text{FR-LEX}}$ alone.

structure of legislation to enrich article content information. Meanwhile, Qin et al. (2024) develop a generative model that learns to represent legal documents as hierarchical semantic IDs before associating queries with their relevant document IDs. Despite this progress, no studies have explored the potential of combining diverse retrieval approaches in the legal domain, especially in zero-shot settings using domain-general models, which may individually struggle due to the specialized nature of law.

French language representation. Existing research in NLP predominantly focuses on English-centric directions (ARR, 2024). In French, efforts have been made in developing monolingual pretrained language models in various configurations: encoder-only (Martin et al., 2020; Le et al., 2020; Antoun et al., 2023), seq2seq (Eddine et al., 2021), and decoder-only (Louis, 2020; Simoulin and Crabbé, 2021; Müller and Laurent, 2022; Launay et al., 2022). Despite these advancements, specialized models for French remain scarce, largely due to the limited availability of high-quality labeled data. This scarcity is particularly pronounced in the field of retrieval, with few exceptions (Arbarétier, 2023). As a result, practitioners typically rely on larger multilingual models (Wang et al., 2024; Chen et al., 2024) that distribute tokens and parameters across various languages, often leading to sub-optimal downstream performance due to the curse of multilinguality (Conneau et al., 2020).

6 Conclusion

Our work explores the potential of combining distinct retrieval methods in a non-English specialized domain, specifically French statute laws. Our findings reveal that supervised domain-general monolingual models, trained with limited resources, can rival leading multilingual retrieval models, though are more vulnerable to out-of-distribution data. However, combining these monolingual models almost consistently enhances their zero-shot performance, regardless of the fusion technique employed, with certain combinations achieving state-of-the-art results in the legal domain. We show the complementary nature of these models and find they can effectively compensate for each other's mistakes, explaining the performance boost. Moreover, we confirm that in-domain training significantly enhances the effectiveness of neural retrieval models, while pre-finetuning can help with dense bi-encoders. Finally, our results indicate that fusion generally does not benefit specialized retrievers and only improves performance when scores are fused with carefully tuned weights, as equal weighting consistently leads to reduced performance. Overall, these insights suggest that for specialized domains, finetuning a single bi-encoder generally yields optimal results when (even limited) high-quality domain-specific data is available, whereas fusion should be preferred when such data is not accessible and domain-general retrievers are used.

Limitations

We identify three core limitations in our research.

Firstly, our analysis specifically targets two underexplored areas – the legal domain and the French language – and is therefore confined to the only dataset available in this niche (LLeQA; Louis et al., 2024). This raises questions about the generalizability of our findings across broader French legal resources, such laws from different French-speaking jurisdictions (e.g., France, Switzerland, or Canada) or across legal topics beyond those covered in LLeQA.

Secondly, our study focuses solely on end-to-end retrievers – i.e., systems that identify and fetch all potentially relevant items from an entire knowledge corpus – as opposed to ranking methods that take the output of retrievers and sort it. Specifically, we deliberately omit the monoBERT$_{\text{FR}}$ ranker due to its prohibitive inference costs for end-to-end retrieval – a brute-force search across all 28K articles in LLeQA requires about two minutes per query on GPU, a latency 9500× higher than that of single-vector retrieval, making it impractical for real-world retrieval. We let the exploration of fusion with re-rankers for future work.

Lastly, although beyond the scope of our work, it remains an open question whether the present findings are applicable to other non-English languages within different highly specialized domains.

Ethical Considerations

The scope of this work is to drive research forward in legal information retrieval by uncovering novel insights on fusion strategies. We believe this is an important application field where more research could improve legal aid services and access to justice for all. We do not foresee major situations where our methodology and findings would lead to harm (Tsarapatsanis and Aletras, 2021). Nevertheless, we emphasize that the premature deployment of prominent retrieval models not tailored for the legal domain poses a tangible risk to laypersons, who may uncritically rely on the provided information when faced with a legal issue and inadvertently worsen their personal situations.

Acknowledgments

This research is partially supported by the Sector Plan Digital Legal Studies of the Dutch Ministry of Education, Culture, and Science. In addition, this research was made possible, in part, using the Data Science Research Infrastructure (DSRI) hosted at Maastricht University.

References

Wissam Antoun, Benoît Sagot, and Djamé Seddah. 2023. Data-efficient french language modeling with camemberta. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 5174–5185. Association for Computational Linguistics. [Pages 8 and 17]

Baudouin Arbarétier. 2023. Solon-embeddings-0.1. Ordalie. Accessed: 2024-07-13. [Page 8]

ARR. 2024. Linguistic diversity statistics. ACL Rolling Review. Accessed: 2024-07-13. [Page 8]

Adam L. Berger, Rich Caruana, David Cohn, Dayne Freitag, and Vibhu O. Mittal. 2000. Bridging the lexical chasm: statistical approaches to answer-finding. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 192–199. Association for Computing Machinery. [Page 1]

Lukas Biewald. 2020. Experiment tracking with weights and biases. Wandb. Accessed: 2024-07-13. [Page 18]

Luiz Henrique Bonifacio, Israel Campiotti, Roberto de Alencar Lotufo, and Rodrigo Nogueira. 2021. mmarco: A multilingual version of MS MARCO passage ranking dataset. *CoRR*, abs/2108.13897. [Pages 2 and 17]

Sebastian Bruch, Siyu Gai, and Amir Ingber. 2024. An analysis of fusion functions for hybrid retrieval. *ACM Transactions on Information Systems*, 42(1):20:1–20:35. [Pages 1 and 2]

Wei-Cheng Chang, Felix X. Yu, Yin-Wen Chang, Yiming Yang, and Sanjiv Kumar. 2020. Pre-training tasks for embedding-based large-scale retrieval. In *Proceedings of the 8th International Conference on Learning Representations*. OpenReview. [Page 2]

Veda R Charrow and Jo Ann Crandall. 1978. Legal language: What is it and what can we do about it? In *Proceedings of the 7th New Wave Conference of the American Dialect Society*. ERIC. [Page 7]

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. BGE m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *CoRR*, abs/2402.03216. [Pages 2 and 8]

Tao Chen, Mingyang Zhang, Jing Lu, Michael Bendersky, and Marc Najork. 2022. Out-of-domain semantics to the rescue! zero-shot hybrid retrieval models. In *Proceedings of the 44th European Conference on IR Research*, pages 95–110. Springer. [Pages 1 and 2]

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. 2020. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning*, pages 1597–1607. PMLR. [Page 13]

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: pre-training text encoders as discriminators rather than generators. In *Proceedings of the 8th International Conference on Learning Representations*. OpenReview. [Page 17]

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451. Association for Computational Linguistics. [Page 8]

Gordon V. Cormack, Charles L. A. Clarke, and Stefan Büttcher. 2009. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 758–759. Association for Computing Machinery. [Pages 2 and 16]

Zhuyun Dai and Jamie Callan. 2019. Deeper text understanding for IR with contextual neural language modeling. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 985–988. Association for Computing Machinery. [Page 6]

Tran-Binh Dang, Thao Nguyen, and Le-Minh Nguyen. 2019. An approach to statute law retrieval task in coliee-2019. *Proceedings of the 6th Competition on Legal Information Extraction/Entailment*. [Page 7]

Jean-Charles de Borda. 1781. *Mémoire sur les élections au scrutin*. Histoire de l'Académie royale des sciences. [Page 15]

Cyrile Delestre and Abibatou Amar. 2022. DistilCamemBERT: Une Distillation du Modèle Français CamemBERT. In *Actes de la Conférence 2021 sur l'Apprentissage Automatique*. [Page 17]

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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*, pages 4171–4186. Association for Computational Linguistics. [Pages 7 and 17]

Moussa Kamal Eddine, Antoine J.-P. Tixier, and Michalis Vazirgiannis. 2021. Barthez: a skilled pretrained french sequence-to-sequence model. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9369–9390. Association for Computational Linguistics. [Page 8]

Thibault Formal. 2023. *Towards Effective, Efficient and Explainable Neural Information Retrieval*. Ph.D. thesis, Sorbonne University, Paris, France. [Page 15]

Thibault Formal, Carlos Lassance, Benjamin Piwowarski, and Stéphane Clinchant. 2021a. SPLADE v2: Sparse lexical and expansion model for information retrieval. *CoRR*, abs/2109.10086. [Pages 2 and 14]

Thibault Formal, Benjamin Piwowarski, and Stéphane Clinchant. 2021b. 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*, pages 2288–2292. Association for Computing Machinery. [Pages 2 and 6]

Baban Gain, Dibyanayan Bandyopadhyay, Tanik Saikh, and Asif Ekbal. 2021. IITP in coliee@icail 2019: Legal information retrieval using BM25 and BERT. *CoRR*, abs/2104.08653. [Page 7]

Luyu Gao, Zhuyun Dai, and Jamie Callan. 2021a. Rethink training of BERT rerankers in multi-stage retrieval pipeline. In *Proceedings of the 43rd European Conference on IR Research*, pages 280–286. Springer. [Page 2]

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021b. Simcse: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910. Association for Computational Linguistics. [Page 13]

Daniel Gillick, Alessandro Presta, and Gaurav Singh Tomar. 2018. End-to-end retrieval in continuous space. *CoRR*, abs/1811.08008. [Page 13]

Shuguang Han, Xuanhui Wang, Mike Bendersky, and Marc Najork. 2020. Learning-to-rank with BERT in tf-ranking. *CoRR*, abs/2004.08476. [Page 2]

Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531. [Page 14]

Tin Kam Ho, Jonathan J. Hull, and Sargur N. Srihari. 1994. Decision combination in multiple classifier systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(1):66–75. [Pages 2 and 15]

Sebastian Hofstätter, Sophia Althammer, Michael Schröder, Mete Sertkan, and Allan Hanbury. 2020. Improving efficient neural ranking models with cross-architecture knowledge distillation. *CoRR*, abs/2010.02666. [Pages 3 and 15]

Sebastian Hofstätter, Sheng-Chieh Lin, Jheng-Hong Yang, Jimmy Lin, and Allan Hanbury. 2021. Efficiently teaching an effective dense retriever with balanced topic aware sampling. In *Proceedings of the 44th International ACM SIGIR Conference on*

Research and Development in Information Retrieval, pages 113–122. Association for Computing Machinery. [Page 1]

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick S. H. Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 6769–6781. Association for Computational Linguistics. [Pages 2, 6, and 14]

Omar Khattab and Matei Zaharia. 2020. 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*, pages 39–48. Association for Computing Machinery. [Pages 2, 6, 15, and 18]

Mi-Young Kim and Randy Goebel. 2017. Two-step cascaded textual entailment for legal bar exam question answering. In *Proceedings of the 16th edition of the International Conference on Artificial Intelligence and Law*, pages 283–290. Association for Computing Machinery. [Page 7]

Saar Kuzi, Mingyang Zhang, Cheng Li, Michael Bendersky, and Marc Najork. 2020. Leveraging semantic and lexical matching to improve the recall of document retrieval systems: A hybrid approach. *CoRR*, abs/2010.01195. [Page 1]

Julien Launay, E. L. Tommasone, Baptiste Pannier, François Boniface, Amélie Chatelain, Alessandro Cappelli, Iacopo Poli, and Djamé Seddah. 2022. Pagnol: An extra-large french generative model. In *Proceedings of the 13th Language Resources and Evaluation Conference*, pages 4275–4284. European Language Resources Association. [Page 8]

Hang Le, Loïc Vial, Jibril Frej, Vincent Segonne, Maximin Coavoux, Benjamin Lecouteux, Alexandre Allauzen, Benoît Crabbé, Laurent Besacier, and Didier Schwab. 2020. Flaubert: Unsupervised language model pre-training for french. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 2479–2490. European Language Resources Association. [Page 8]

Joon Ho Lee. 1995. Combining multiple evidence from different properties of weighting schemes. In *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 180–188. Association for Computing Machinery. [Pages 2 and 16]

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. In *Proceedings of the 57th Conference of the Association for Computational Linguistics*, pages 6086–6096. Association for Computational Linguistics. [Page 2]

Canjia Li, Andrew Yates, Sean MacAvaney, Ben He, and Yingfei Sun. 2020. PARADE: passage representation aggregation for document reranking. *CoRR*, abs/2008.09093. [Page 6]

Antoine Louis. 2020. Belgpt-2: a gpt-2 model pre-trained on french corpora. Accessed: 2024-07-13. [Page 8]

Antoine Louis and Gerasimos Spanakis. 2022. A statutory article retrieval dataset in french. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, pages 6789–6803. Association for Computational Linguistics. [Page 7]

Antoine Louis, Gijs van Dijck, and Gerasimos Spanakis. 2023. Finding the law: Enhancing statutory article retrieval via graph neural networks. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2753–2768. Association for Computational Linguistics. [Page 7]

Antoine Louis, Gijs van Dijck, and Gerasimos Spanakis. 2024. Interpretable long-form legal question answering with retrieval-augmented large language models. In *Proceedings of the 38th AAAI Conference on Artificial Intelligence*, pages 22266–22275. Association for the Advancement of Artificial Intelligence. [Pages 2 and 9]

Xueguang Ma, Kai Sun, Ronak Pradeep, and Jimmy Lin. 2021. A replication study of dense passage retriever. *CoRR*, abs/2104.05740. [Page 1]

Yury Malkov, Alexander Ponomarenko, Andrey Logvinov, and Vladimir Krylov. 2014. Approximate nearest neighbor algorithm based on navigable small world graphs. *Information Systems*, 45:61–68. [Page 3]

Louis Martin, Benjamin Müller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric de la Clergerie, Djamé Seddah, and Benoît Sagot. 2020. Camembert: a tasty french language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7203–7219. Association for Computational Linguistics. [Pages 2, 8, and 17]

Milvus. 2022. Vector index. Accessed: 2024-07-09. [Page 3]

Martin Müller and Florian Laurent. 2022. Cedille: A large autoregressive french language model. *CoRR*, abs/2202.03371. [Page 8]

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2018. MS MARCO: A human generated machine reading comprehension dataset. *CoRR*, abs/1611.09268v3. [Page 2]

Rodrigo Frassetto Nogueira and Kyunghyun Cho. 2019. Passage re-ranking with BERT. *CoRR*, abs/1901.04085. [Pages 2 and 15]

Rodrigo Frassetto Nogueira, Wei Yang, Kyunghyun Cho, and Jimmy Lin. 2019. Multi-stage document ranking with BERT. *CoRR*, abs/1910.14424. [Page 6]

Biswajit Paria, Chih-Kuan Yeh, Ian En-Hsu Yen, Ning Xu, Pradeep Ravikumar, and Barnabás Póczos. 2020. Minimizing flops to learn efficient sparse representations. In *Proceedings of the 8th International Conference on Learning Representations*. OpenReview. [Page 14]

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 32:8024–8035. [Page 18]

Weicong Qin, Zelin Cao, Weijie Yu, Zihua Si, Sirui Chen, and Jun Xu. 2024. Explicitly integrating judgment prediction with legal document retrieval: A law-guided generative approach. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2210–2220. Association for Computing Machinery. [Page 8]

Yingqi Qu, Yuchen Ding, Jing Liu, Kai Liu, Ruiyang Ren, Wayne Xin Zhao, Daxiang Dong, Hua Wu, and Haifeng Wang. 2021. Rocketqa: An optimized training approach to dense passage retrieval for open-domain question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5835–5847. Association for Computational Linguistics. [Pages 1 and 3]

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding with unsupervised learning. [Page 17]

Redis. 2024. Vectors: Flat index. Accessed: 2024-07-09. [Page 3]

Nils Reimers and Iryna Gurevych. 2019. 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*, pages 3980–3990. Association for Computational Linguistics. [Pages 1 and 18]

Stephen E. Robertson, Steve Walker, Susan Jones, Micheline Hancock-Beaulieu, and Mike Gatford. 1994. Okapi at TREC-3. In *Proceedings of the 3rd Text REtrieval Conference*, pages 109–126. National Institute of Standards and Technology. [Pages 1, 2, and 13]

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108. [Page 17]

Keshav Santhanam, Omar Khattab, Christopher Potts, and Matei Zaharia. 2022a. PLAID: an efficient engine for late interaction retrieval. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 1747–1756. Association for Computing Machinery. [Pages 3 and 15]

Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. 2022b. Colbertv2: Effective and efficient retrieval via lightweight late interaction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3715–3734. Association for Computational Linguistics. [Pages 2, 3, and 15]

T. Y. S. S. Santosh, Kristina Kaiser, and Matthias Grabmair. 2024. Cusines: Curriculum-driven structure induced negative sampling for statutory article retrieval. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics*, pages 4266–4272. European Language Resources Association. [Page 7]

Stefan Schweter. 2020. Europeana bert and electra models. Zenodo. Accessed: 2024-07-02. [Page 17]

Joseph A. Shaw and Edward A. Fox. 1994. Combination of multiple searches. In *Proceedings of the 3rd Text REtrieval Conference*, pages 105–108. National Institute of Standards and Technology. [Page 16]

Antoine Simoulin and Benoit Crabbé. 2021. Un modèle transformer génératif pré-entrainé pour le______ français. In *Actes de la 28e Conférence sur le Traitement Automatique des Langues Naturelles*, pages 246–255. Association pour le Traitement Automatique des Langues. [Page 8]

Weihang Su, Yiran Hu, Anzhe Xie, Qingyao Ai, Zibing Que, Ning Zheng, Yun Liu, Weixing Shen, and Yiqun Liu. 2024. STARD: A chinese statute retrieval dataset with real queries issued by non-professionals. *CoRR*, abs/2406.15313. [Page 7]

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *Proceedings of the 35th Conference on Neural Information Processing Systems: Datasets and Benchmarks Track*. [Pages 1, 13, 14, and 15]

Dimitrios Tsarapatsanis and Nikolaos Aletras. 2021. On the ethical limits of natural language processing on legal text. In *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021*, pages 3590–3599. Association for Computational Linguistics. [Page 9]

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*, 30:5998–6008. [Page 7]

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024. Multilingual E5 text embeddings: A technical report. *CoRR*, abs/2402.05672. [Pages 2 and 8]

Shuai Wang, Shengyao Zhuang, and Guido Zuccon. 2021. Bert-based dense retrievers require interpolation with BM25 for effective passage retrieval. In *Proceedings of the 2021 ACM SIGIR International Conference on the Theory of Information Retrieval*, pages 317–324. Association for Computing Machinery. [Page 1]

Sabine Wehnert, Sayed Anisul Hoque, Wolfram Fenske, and Gunter Saake. 2019. Threshold-based retrieval and textual entailment detection on legal bar exam questions. *CoRR*, abs/1905.13350. [Page 7]

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45. Association for Computational Linguistics. [Page 18]

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. 2021. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *Proceedings of the 9th International Conference on Learning Representations*. OpenReview. [Page 1]

Xinyu Zhang, Andrew Yates, and Jimmy Lin. 2020. A little bit is worse than none: Ranking with limited training data. In *Proceedings of SustaiNLP: Workshop on Simple and Efficient Natural Language Processing*, pages 107–112. Association for Computational Linguistics. [Page 6]

A Methodology Details

Formally speaking, a statutory article retrieval system takes as input a question q along with a corpus of law articles $\mathcal{C}$, and returns a ranked list $\mathcal{R}_q \subset \mathcal{C}$ of the supposedly relevant articles, sorted by decreasing order of relevance.

A.1 Retrieval Models

BM25 (Robertson et al., 1994) is an unsupervised probabilistic weighting scheme that estimates relevance based on term-matching between high-dimensional sparse vectors using statistical properties such as term frequencies, document frequencies, and document lengths. Specifically, it calculates a relevance score $s(q, a) : \mathcal{V}^{|q|} \times \mathcal{V}^{|a|} \rightarrow \mathbb{R}_+$ between query q and article a as a sum of contributions of each query term t from vocabulary $\mathcal{V}$ appearing in the article, i.e.,

$$s_{\text{BM25}}(q,a) = \sum_{t \in q} \log\left(\frac{|\mathcal{C}| - \text{df}(t) + 0.5}{\text{df}(t) + 0.5}\right) \cdot \frac{\text{tf}(t,a)\cdot(k_1+1)}{\text{tf}(t,a)+k_1\cdot\left(1-b+b\cdot\frac{|a|}{avgal}\right)}, \tag{1}$$

where the term frequency $\text{tf}(t,a) : \mathcal{V}^1 \times \mathcal{V}^{|a|} \rightarrow \mathbb{Z}_+$ is the number of occurrences of term t in article a, the document frequency $\text{df}(t) : \mathcal{V}^1 \rightarrow \mathbb{Z}_+$ is the number of articles within the corpus $\mathcal{C}$ that contain term t, $k_1 \in \mathbb{R}_+$ and $b \in [0,1]$ are constant parameters, and $avgal$ is the average article length. BM25 remains widely used due to its balance between simplicity and robustness, often competing with modern retrieval methods (Thakur et al., 2021) while being extremely efficient and requiring no training. However, its reliance on exact-term matching restricts its ability to understand semantics, capture contextual relationships, and handle synonyms or rare terms.

DPR$_{\text{FR-\{BASE,LEX\}}}$ are based on the widely-adopted siamese bi-encoder architecture (Gillick et al., 2018), which consists of a learnable text embedding function $E(i; \mathbf{\Omega}) : \mathcal{V}^n \mapsto \mathbb{R}^{n \times d}$ that maps an input text sequence i of n terms from vocabulary $\mathcal{V}$ to d-dimensional real-valued term vectors, i.e.,

$$E(i; \mathbf{\Omega}) = \mathbf{H}_i = [\mathbf{h}_{i,\text{CLS}}, \mathbf{h}_{i,1}, \cdots, \mathbf{h}_{i,n}], \tag{2}$$

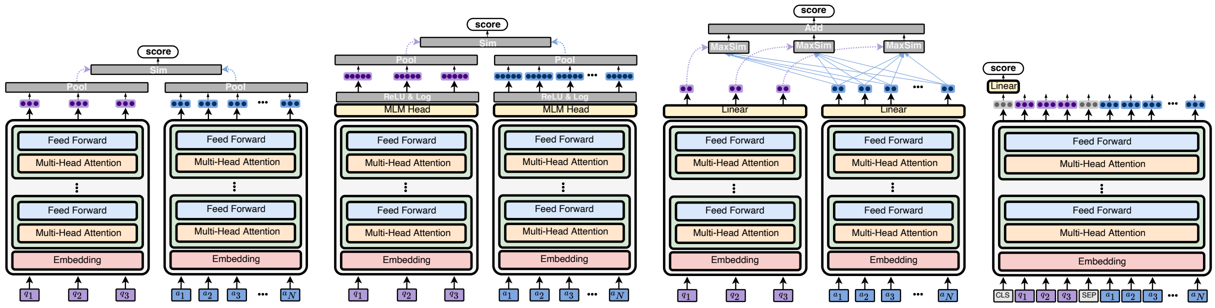
and calculates a relevance score between query q and article a by operating on their independently computed bags of contextualized term embeddings $\mathbf{H}_i \in \mathbb{R}^{n \times d}$. Our single-vector dense representation models obtain this score by performing

$$s_{\text{SINGLE}}(q,a) = \mathbf{h}_q^* \cdot \mathbf{h}_a^*, \tag{3}$$

where $\mathbf{h}_i^* \in \mathbb{R}^d$ is the global representation of sequence i, derived by mean pooling across the sequence term embeddings, i.e.,

$$\mathbf{h}_i^* = \text{AvgP}(\mathbf{H}_i) = \frac{1}{|i|}\mathbf{H}_i^\top \mathbf{1}_{|i|}. \tag{4}$$

The models are trained via optimization of the contrastive NT-Xent loss (Chen et al., 2020; Gao et al., 2021b), which aims to learn a high-quality embedding function so that relevant query-article pairs



(a) Single-vector dense bi-encoder (b) Single-vector sparse bi-encoder (c) Multi-vector dense bi-encoder (d) Cross-encoder

Figure 5: High-level illustration of the four prominent neural retrieval architectures explored in this study.

achieve higher similarity than irrelevant ones. Let $\mathcal{B} = \{(q_i, a_i^+, a_{\text{H},i}^-)\}_{i=1}^N$ be a batch of N training instances, each comprising a query q_i associated with a positive article a_i^+ and a hard negative article $a_{\text{H},i}^-$. By considering the articles paired with all other queries within the same batch, we can enrich each training triple with an additional set of $2(N-1)$ in-batch negatives $\mathcal{A}_{\text{IB},i}^- = \{a_j^+, a_{\text{H},j}^-\}_{j\neq i}^N$. Given these augmented training samples, we contrastively optimize the negative log-likelihood of each positive article such that

$$\mathcal{L}_{\text{NT-XENT}} = -\log \frac{e^{s(q_i,a_i^+)/\tau}}{\sum_{a\in\{a_i^+,a_{\text{H},i}^-\}\cup\mathcal{A}_{\text{IB},i}^-} e^{s(q_i,a)/\tau}}, \tag{5}$$

where $\tau \in \mathbb{R}_+$ is a temperature hyper-parameter that controls the concentration level of the distribution (Hinton et al., 2015). We enforce $\|\mathbf{h}_i^*\| = 1$ via a ℓ_2-normalization layer such that Equation (3) computes the cosine similarity.

Single-vector dense models proved to effectively model language nuances and contextual information (Karpukhin et al., 2020). Furthermore, the independent encoding enables offline precomputation of article embeddings, resulting in low latency query-time retrieval. However, its effectiveness can be limited by the quality and diversity of its training data, potentially leading to sub-optimal performance with out-of-distribution content (Thakur et al., 2021).

SPLADE$_{\text{FR-\{BASE,LEX\}}}$ follow SPLADE-max (Formal et al., 2021a), which uses the same single-vector scoring mechanism as its dense representation counterpart, outlined in Equation (3), but operates on different global sequence representations derived as follows:

$$\mathbf{h}_i^* = \text{MaxP}\Big(\text{sat}\Big(\text{transf}(\mathbf{H}_i)\mathbf{W}_{\text{MLM}}^\intercal + \mathbf{b}_{\text{MLM}}\Big)\Big), \tag{6}$$

where $\text{transf}(\cdot;\boldsymbol{\gamma}) : \mathbb{R}^{n\times d} \rightarrow \mathbb{R}^{n\times d}$ first transforms the contextualized term embeddings using

$$\text{transf}(\cdot;\boldsymbol{\gamma}) = \text{LayerNorm}(\text{GELU}(\text{Linear}(\cdot))), \tag{7}$$

preparing them for subsequent projection onto the vocabulary space via the MLM classification head $\mathbf{W}_{\text{MLM}} \in \mathbb{R}^{|\mathcal{V}|\times d}$, with bias $\mathbf{b}_{\text{MLM}} \in \mathbb{R}^{|\mathcal{V}|}$. The function $\text{sat}(\cdot) : \mathbb{R}^{n\times|\mathcal{V}|} \rightarrow \mathbb{R}^{n\times|\mathcal{V}|}$ then applies ReLU to ensure positive token activations, before performing log-saturation to maintain sparsity and prevent some tokens from dominating:

$$\text{sat}(\cdot) = \log\left(1 + \text{ReLU}(\cdot)\right). \tag{8}$$

Finally, a max pooling operation $\text{MaxP}(\cdot) : \mathbb{R}^{n\times|\mathcal{V}|} \rightarrow \mathbb{R}^{|\mathcal{V}|}$ is applied to distill the global sequence representation. The model is trained by jointly optimizing the contrastive NT-Xent objective, presented in Equation (5), and the FLOPS regularization loss (Paria et al., 2020), which aims to impose sparsity on the produced embeddings while encouraging an even distribution of the non-zero elements across all the dimensions to ensure maximal speedup. This is achieved by minimizing a smooth relaxation of the average number of floating-point operations necessary to compute the dot product between two embeddings (as outlined in Equation (3)), defined as follows:

$$\ell_{\text{FLOPS}} = \sum_{j=1}^{|\mathcal{V}|} \bar{p}_j^2 = \sum_{j=1}^{|\mathcal{V}|} \left(\frac{1}{|\mathcal{B}|}\sum_{i=1}^{|\mathcal{B}|} \mathbf{h}_{ij}^*\right)^2 \tag{9}$$

where $\bar{p}_j \approx |\mathcal{B}|^{-1}\sum_{i=1}^{|\mathcal{B}|} \mathbb{1}[\mathbf{h}_{ij}^* \neq 0]$ is the empirical estimation of the activation probability for token $t_j \in \mathcal{V}$ over a batch $\mathcal{B}$. The overall loss is given by

$$\mathcal{L}_{\text{SPLADE}} = \mathcal{L}_{\text{NT-XENT}} + \lambda_q \ell_{\text{FLOPS}}^q + \lambda_a \ell_{\text{FLOPS}}^a, \tag{10}$$

where λ_i controls the strength of the regularization, with higher values typically encouraging the

model to learn sparser representations, therefore enhancing efficiency yet often at the expense of effectiveness. By applying separate regularization weights for queries and articles, greater emphasis can be placed on sparsity for queries, which is critical for fast inference with inverted indexes.
As its representations are grounded in the encoder's vocabulary, SPLADE enhances interpretability and facilitates explanations of observed rankings. It also exhibits strong generalization capabilities on out-of-distribution data and the sparsity of its vectors enables the use of inverted indexes for fast inference. Nevertheless, learning sparse representations in high-dimensional spaces poses specific challenges: factors such as the tokenization type or the initial distribution of MLM weights can lead to model divergence (Formal, 2023).

ColBERT$_{\text{FR-\{BASE,LEX\}}}$ use the fine-granular late interaction scoring mechanism of ColBERT (Khattab and Zaharia, 2020), which calculates the similarity across all pairs of query and article token embeddings, applies max-pooling across the resulting scores for each query term, and then sum the maximum values across query terms to derive the overall relevance estimate, i.e.,

$$s_{\text{MULTI}}(q,a) = \sum_{i=1}^{|q|} \max_{j=1}^{|a|} \mathbf{h}_{q,i} \cdot \mathbf{h}_{a,j}. \tag{11}$$

We train the model by jointly optimizing two contrastive objectives, namely the pairwise softmax cross-entropy loss used in ColBERTv1, defined as

$$\mathcal{L}_{\text{PAIRSM-CE}} = -\log \frac{e^{s(q_i,a_i^+)}}{e^{a(q_i,a_i^+)} + e^{s(q_i,a_{\text{H},i}^-)}}, \tag{12}$$

and the NT-Xent loss, added as an enhancement for optimizing ColBERTv2 (Santhanam et al., 2022b).
ColBERT's fine-grained late interaction between term embeddings demonstrates greater effectiveness and robustness to out-of-distribution data compared to single-vector dense bi-encoders (Thakur et al., 2021), while enabling result interpretability. However, its computational complexity requires sophisticated engineering schemes and low-level optimizations for efficient large-scale deployment (Santhanam et al., 2022a).

monoBERT$_{\text{FR-\{BASE,LEX\}}}$ exploit the encoder-only cross-attention model structure (Nogueira and Cho, 2019), which uses a text embedding model similar to the one defined in Equation (2) to perform all-to-all interactions across terms from concatenated query-article pairs, before deriving a relevance score through binary classification on the pair representation, i.e.,

$$s_{\text{MONO}}(q,a) = \sigma\Big(\text{transf}\Big(\mathbf{h}^*_{[q;a]}\Big)\,\mathbf{W}_{\text{out}}^{\mathsf{T}} + \mathbf{b}_{\text{out}}\Big), \tag{13}$$

where $\mathbf{h}^*_{[q;a]} \in \mathbb{R}^d$ is obtained through a first token pooling operation $\text{FirstP}(\cdot) : \mathbb{R}^{n\times d} \to \mathbb{R}^d$, which extracts the special CLS token representation of the concatenated sequence:

$$\mathbf{h}^*_{[q;a]} = \text{FirstP}\big(\mathbf{H}_{[q;a]}\big) = \mathbf{h}_{[q;a],\text{CLS}}. \tag{14}$$

The CLS token embedding is then transformed with $\text{transf}(\cdot;\boldsymbol{\theta}) : \mathbb{R}^d \to \mathbb{R}^d$ such that

$$\text{transf}(\cdot;\boldsymbol{\theta}) = \tanh(\text{Linear}(\cdot)), \tag{15}$$

before being projected to a real-valued score via a linear layer $\mathbf{W}_{\text{out}} \in \mathbb{R}^{1\times d}$ with bias $\mathbf{b}_{\text{out}} \in \mathbb{R}^d$. Finally, the sigmoid function σ bounds the resulting score to the interval $[0,1]$. The model is optimized via the binary cross-entropy training objective

$$\begin{aligned}\mathcal{L}_{\text{BCE}} = &- y_i \cdot \log\left(s(q_i,a_i)\right) \\ &- (1-y_i)\cdot \log\left(1 - s(q_i,a_i)\right),\end{aligned} \tag{16}$$

where y_i is the ground-truth relevance label for query-article pair (q_i,a_i).
The rich interaction mechanism of such a model allows to capture complex relationships and often achieve state-of-the-art performance in retrieval tasks (Hofstätter et al., 2020). However, its high computational complexity makes it impractical for large-scale or real-time retrieval scenarios, limiting its use to re-ranking small candidate sets only.

A.2 Late Fusion Techniques

A late fusion function $f(q,a,\mathcal{M}) : \mathcal{V}^{|q|} \times \mathcal{V}^{|a|} \times \mathcal{M} \to \mathbb{R}_+$ computes a relevance score between query q and article a by combining the ranked lists of articles $\mathcal{R}_m \subset \mathcal{C}$ returned separately by a set of retrieval models $\mathcal{M}$.

Borda count fusion (BCF) uses a straightforward approach – originally developed as a voting mechanism (de Borda, 1781) – which combines the ranks from different systems linearly (Ho et al., 1994) such that

$$f_{\text{BCF}}(q,a,\mathcal{M}) = \sum_{m\in\mathcal{M}} |\mathcal{R}_m| - \pi_m(q,a) + 1, \tag{17}$$

where $\pi_m(q, a) \in [1, |\mathcal{R}_m|]$ denotes the rank of article a in the list of results returned by model m for query q, i.e.,

$$\pi_m(q, a) = 1 + \sum_{a_i \in \mathcal{C}} \mathbb{1}[s_m(q, a_i) > s_m(q, a)]. \tag{18}$$

Reciprocal rank fusion (RRF) refines the previous approach by introducing a non-linear weighting scheme that gives more emphasis to top-ranked documents (Cormack et al., 2009), i.e.,

$$f_{\text{RRF}}(q, a, \mathcal{M}) = \sum_{m \in \mathcal{M}} \frac{1}{k + \pi_m(q, a)}, \tag{19}$$

where $k > 0$ is a constant set to 60 by default.

Normalized score fusion (NSF) linearly combines the output relevance scores from distinct retrieval models (Lee, 1995) such that

$$f_{\text{NSF}}(q, a, \mathcal{M}) = \sum_{m \in \mathcal{M}} \alpha_m \hat{s}_m(q, a), \tag{20}$$

where the scalars α_m, controlling the relative importance of each model m in the fused score, are non-negative and sum to one. These weights can be varied or uniformly distributed, as in CombSUM (Shaw and Fox, 1994). Given that the original model-specific scores can be unbounded, they are generally normalized prior to fusion, using either min-max scaling where

$$\hat{s}_m(q, a) = \frac{s_m(q, a) - \min_{i=1}^{|\mathcal{C}|} s_m(q, a_i)}{\max_{i=1}^{|\mathcal{C}|} s_m(q, a_i) - \min_{i=1}^{|\mathcal{C}|} s_m(q, a_i)}, \tag{21}$$

or z-score scaling such that

$$\hat{s}_m(q, a) = \frac{s_m(q, a) - \mu_m(q)}{\sigma_m(q)}, \tag{22}$$

where $\mu_m(q)$ is the mean score across all candidate articles in the ranked list for query q returned by model m, and $\sigma_m(q)$ denotes the standard deviation of these scores. Beyond these conventional scaling methods, we also investigate a percentile-based normalization, the rationale and specifics of which are elaborated in Section 3.

A.3 Evaluation Metrics

Let $\text{rel}(q, a) : \mathcal{V}^m \times \mathcal{V}^n \to \{0, 1\}$ be a binary relevance function, indicating whether an article a from the corpus $\mathcal{C}$ is relevant to a query q. Assume that $\mathcal{R}_q = \{(i, a)\}_{i=1}^{k}$ denotes the ranked list of articles returned by a retrieval system, truncated at the top-k results. We define the metrics mentioned in Section 2.3 as follows.

Recall@k. The metric quantifies the proportion of relevant articles retrieved within the top-k ranked results for query q, compared to the total number of relevant articles in the corpus $\mathcal{C}$, i.e.,

$$\text{R@}k(q, \mathcal{R}_q) = \frac{\sum_{(i,a) \in \mathcal{R}_q} \text{rel}(q, a)}{\sum_{a \in \mathcal{C}} \text{rel}(q, a)}. \tag{23}$$

Reciprocal rank@k. The metric takes the inverse of the position at which the first relevant article appears within the top-k results for query q, i.e.,

$$\text{RR@}k(q, \mathcal{R}_q) = \max_{(i,a) \in \mathcal{R}_q} \frac{\text{rel}(q, a)}{i}. \tag{24}$$

R-precision. The metric computes the ratio of relevant articles within the top-N retrieved results for query q, where N represents the total number of relevant articles for that query, i.e.,

$$\text{RP}(q, \mathcal{R}_q) = \frac{\sum_{(i,a) \in \{\mathcal{R}_q\}_{i=1}^{N}} \text{rel}(q, a)}{N}. \tag{25}$$

For all metrics, we report the average scores over a set of Q queries.

A.4 Counting FLOPs

Below, we detail our methodology to estimate the inference complexity per query in terms of floating point operations (FLOPs). Except for BM25, the main computational cost derives from the Transformer encoder's forward pass, executed once with bi-encoder models to encode the query and repeatedly in cross-encoders to process each query-article pair. We leverage DeepSpeed's profiler to measure the forward pass cost of each neural retriever.[3] Queries are assumed to be 15 tokens and articles 157 tokens, as per their respective average lengths in LLeQA.

BM25. In the BM25 scoring formula, outlined in Equation (1), several elements can be pre-computed and cached to streamline computations during inference. These include the inverse document frequency (IDF) for each term, the normalized document lengths adjusted by the parameters $k1$ and b, and the constant $(k1+1)$. For each query term and candidate document, the process involves four primary operations. First, the term frequency (TF), retrieved via a simple lookup, is multiplied by the pre-computed IDF and $(k1+1)$. The result is then added to the stored normalized document

[3] https://www.deepspeed.ai/tutorials/flops-profiler/

French PLM Backbone	#Params	Architecture	#L	Pre-training	MRR@10	R@100	R@500
DistilCamemBERT (Delestre and Amar, 2022)	68.1M	BERT	6	MLM+KL+COS	<u>0.268</u>	<u>0.764</u>	<u>0.879</u>
ELECTRA-fr$_{\text{BASE}}$ (Schweter, 2020)	110.0M	BERT	12	RTD	0.234	0.690	0.816
CamemBERT$_{\text{BASE}}$ (Martin et al., 2020)	110.6M	BERT	12	MLM	**0.285**	**0.778**	**0.891**
CamemBERTa$_{\text{BASE}}$ (Antoun et al., 2023)	111.8M	DeBERTa	12	RTD	0.248	0.696	0.822

Table 7: In-domain retrieval performances on mMARCO-fr small dev set (Bonifacio et al., 2021) for single-vector dense representation models trained using various French pretrained autoencoding language models as their text embedding backbone. MLM, RTD, KL, and COS denote the masked language modeling (Devlin et al., 2019), replaced token detection (Clark et al., 2020), Kullback-Leibler divergence (Radford et al., 2018), and negative cosine embedding (Sanh et al., 2019) training objectives, respectively. #L indicates the number of encoder layers.

length. Finally, this sum is used as the denominator in dividing the product of TF, IDF, and $(k1+1)$. These four operations – two multiplications, one addition, and one division – per term-article pair lead to an overall computational cost of $4\overline{|q|}|\mathcal{C}|$ FLOPs for searching across the whole corpus.

SPLADE$_{\text{FR-BASE}}$. At indexing time, this model creates a pseudo-TF for each token t in the vocabulary by scaling and rounding the corresponding activation weights in sparse article representations. This enables the construction of a pseudo text collection where each term t is repeated TF(t, a) times for article a. During inference, obtaining the query representation requires a single forward pass. For each non-zero term in that representation, the search process involves three core steps: accessing the inverted list for the term (a negligible lookup operation), multiplying the query term weight by each article term weight from that list, and adding each result to the corresponding article's score accumulator. Consequently, for each term-article pair, the operations include one multiplication and one addition. With C_{FW} representing the cost of the encoder's forward pass, $\overline{|\mathbf{h}_q^+|}$ the average number of non-zero terms in the query representation, and $\overline{|\mathcal{L}_{\text{IV}}|}$ the average length of the inverted lists for these terms, the total computational complexity is estimated as $C_{\text{FW}} + 2\overline{|\mathbf{h}_q^+|}\,\overline{|\mathcal{L}_{\text{IV}}|}$ FLOPs.[4]

Single-vector dense bi-encoders. With these models, a brute-force search across all articles from corpus $\mathcal{C}$ necessitates $|\mathcal{C}|$ inner products between d-dimensional article representations – each involving d multiplications and $d-1$ additions. Consequently, the total inference cost amounts to $C_{\text{FW}} + (2d-1)|\mathcal{C}|$ operations.

[4]On LLeQA, the FR-BASE model activates an average of 178 tokens per query, and the associated index features inverted lists of 378 elements on average.

ColBERT$_{\text{FR-BASE}}$. For each candidate article, this model computes Equation (11) with the query and candidate token representations of d dimensions. For each query term, this computation requires $2d\overline{|q|}\,\overline{|a|}$ operations for token-level inner products, $\overline{|q|}\,\overline{|a|}$ to identify the row-wise max, and $\overline{|q|}$ for the final average. When performing brute-force search across the entire corpus, the inference complexity is estimated as $C_{\text{FW}} + \overline{|q|}^2(2d\overline{|a|}+\overline{|a|}+1)|\mathcal{C}|$ FLOPs.

monoBERT$_{\text{FR-BASE}}$. This model requires one forward pass per article to assess, incurring a high computational cost that typically limits their use to re-ranking a set of candidates returned by a cheaper retrieval model. To reflect that practice, we report the number of operations needed to score a fixed set of 1000 articles, resulting in $10^3 C_{\text{FW}}$ FLOPs.

B Implementation Details

B.1 Embedding Backbone

To ensure a fair comparison between the different matching paradigms detailed in Section 2.1, irrespective of the underlying text embedding model's capacity, we choose to exploit the same pretrained autoencoding language model across all our neural retrievers. To explore the efficacy of existing French embedding models for text retrieval, we finetune four prominent pretrained models on mMARCO-fr, including CamemBERT$_{\text{BASE}}$ (Martin et al., 2020), ELECTRA-fr$_{\text{BASE}}$ (Schweter, 2020), DistilCamemBERT (Delestre and Amar, 2022), and CamemBERTa$_{\text{BASE}}$ (Antoun et al., 2023). We limit our investigation to the performance of single-vector dense bi-encoders to minimize environmental impact. Table 7 presents the results on the mMARCO-fr small dev set, revealing that CamemBERT$_{\text{BASE}}$ significantly outperforms the other French text encoders. Following these findings, we select this model as the common backbone encoder for all our neural retrievers.

Training data (→)	mMARCO-fr				LLeQA			
Learned model (→)	**DPR$_{\text{FR-BASE}}$**	**SPLADE$_{\text{FR-BASE}}$**	**ColBERT$_{\text{FR-BASE}}$**	**monoBERT$_{\text{FR-BASE}}$**	**DPR$_{\text{FR-LEX}}$**	**SPLADE$_{\text{FR-LEX}}$**	**ColBERT$_{\text{FR-LEX}}$**	**monoBERT$_{\text{FR-LEX}}$**
Configuration								
Max query length	128	32	32	-	512	64	64	-
Max article length	128	128	128	$256 - \|q\|$	512	512	512	$512 - \|q\|$
Pooling strategy	mean	max	-	cls	mean	max	-	cls
Similarity function	cos	cos	cos	-	cos	cos	cos	-
Hyperparameters								
Steps	66k	100k	200k	20k	1k	2k	1k	2k
Batch size	152	128	128	128	64	32	64	64
Optimizer	AdamW	AdamW	AdamW	AdamW	AdamW	AdamW	AdamW	AdamW
Weight decay	0.01	0.01	0.0	0.01	0.01	0.01	0.0	0.01
Peak learning rate	2e-5	2e-5	5e-6	2e-5	2e-5	2e-5	5e-6	2e-5
Learning rate decay	linear	linear	linear	constant	constant	constant	constant	constant
Warm-up ratio	0.01	0.04	0.1	0.0	0.0	0.0	0.0	0.0
Gradient clipping	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Softmax temperature	0.05	0.05	1.0	-	0.05	0.05	1.0	-
Energy								
Hardware	V100	H100	H100	H100	H100	H100	H100	H100
Thermal design power (W)	300	310	310	310	310	310	310	310
Training time (h)	14.1	12.9	18.4	1.5	0.22	0.30	0.18	0.17
Power consumption (kWh)	4.2	4.0	5.7	0.5	0.07	0.09	0.06	0.05
Carbon emission (kgCO_2eq)	1.8	1.7	2.5	0.2	0.03	0.04	0.03	0.02

Table 8: Implementation details for our learned domain-general (FR-BASE) and domain-specific (FR-LEX) retrievers.

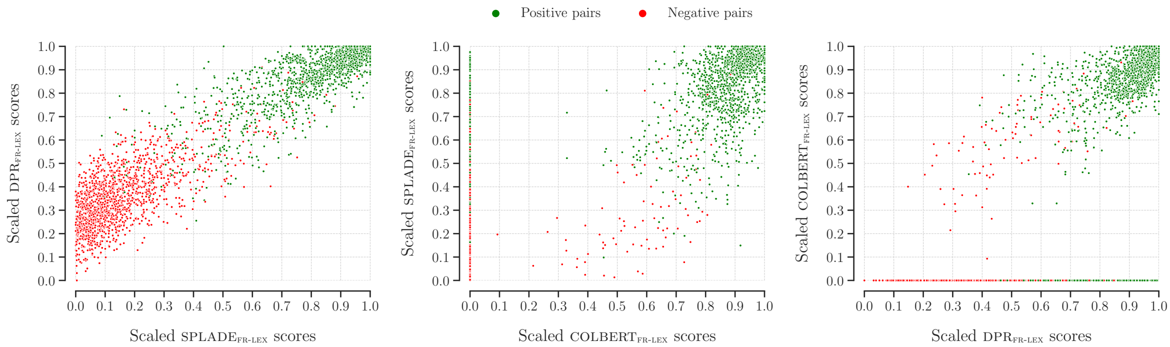


Figure 6: Distribution of paired relevance scores from our learned specialized retrievers on around 3,000 query-article pairs from the LLeQA dev set, evenly balanced between positive and negative instances.

B.2 Optimization

Table 8 provides details on our models' configuration, training hyperparameters, and energy consumption. Training and GPU-based experiments are conducted on a single 80GB NVIDIA H100, while CPU-based evaluations are performed on a server with a 64-core AMD EPYC 7763 CPU at 3.20GHz and 500GB of RAM. We implement, train, tune, and monitor our models using the following Python libraries: `pytorch` (Paszke et al., 2019), `transformers` (Wolf et al., 2020), `sentence-transformers` (Reimers and Gurevych, 2019), `colbert-ai` (Khattab and Zaharia, 2020), and `wandb` (Biewald, 2020).

C Additional Results

C.1 Complementarity of Specialized Models

To understand why fusion does not enhance the performance of specialized retrievers, we examine the complementarity of their relevance signals in Figure 6. We sample approximately 1,500 positive query-article pairs from the LLeQA dev set, along with an equal number of random negatives, and gather the scores assigned by the different models to each pair. Contrary to the zero-shot context, we find that the output scores from the specialized models align closely, as shown by the linear distribution of paired scores in Figure 6. Pairs that receive high relevance scores from one system typically receive similar scores from others, and the same applies to lower scores. We hypothesize that since all retrieval models were trained on a limited number of the exact same domain-specific data with the same primary contrastive learning objective, they converged towards learning similar relevance signals, with some models like DPR$_{\text{FR-LEX}}$ developing more nuanced ones. Consequently, fusing models that have learned related signals, but with varying levels of accuracy, generally results in

#	BM25	DPR$_{\text{FR-BASE}}$	SPLADE$_{\text{FR-BASE}}$	ColBERT$_{\text{FR-BASE}}$
		Min-max scaling		
5	.50	0	.50	0
6	0	.25	0	.75
7	.40	.60	0	0
8	.40	0	0	.60
9	0	.70	.30	0
10	0	0	.20	.80
11	.25	.25	.50	0
12	.35	0	.40	.25
13	.35	.25	0	.40
14	0	.10	.20	.70
15	.30	.35	.10	.25
		Z-score scaling		
5	.40	0	.60	0
6	0	.25	0	.75
7	.30	.70	0	0
8	.25	0	0	.75
9	0	.80	.20	0
10	0	0	.20	.80
11	.20	.40	.40	0
12	.20	0	.40	.40
13	.20	.30	0	.50
14	0	.40	.10	.50
15	.15	.45	.10	.30
		Percentile scaling		
5	.60	0	.40	0
6	0	.05	0	.95
7	.50	.50	0	0
8	.40	0	0	.60
9	0	.85	.15	0
10	0	0	.20	.80
11	.45	.05	.50	0
12	.55	0	.35	.10
13	.50	.40	0	.10
14	0	.05	.70	.25
15	.50	.05	.40	.05

Table 9: Optimally tuned weights for the normalized score fusion results presented in Table 3 (zero-shot).

#	BM25	DPR$_{\text{FR-LEX}}$	SPLADE$_{\text{FR-LEX}}$	ColBERT$_{\text{FR-LEX}}$
		Min-max scaling		
5	.15	0	.85	0
6	0	.85	0	.15
7	.10	.90	0	0
8	.15	0	0	.85
9	0	.70	.30	0
10	0	0	.85	.15
11	.05	.60	.35	0
12	.15	0	.75	.10
13	.10	.80	0	.10
14	0	.60	.25	.15
15	.05	.60	.30	.05
		Z-score scaling		
5	.10	0	.90	0
6	0	.65	0	.35
7	.05	.95	0	0
8	.05	0	0	.95
9	0	.70	.30	0
10	0	0	.75	.25
11	.05	.55	.40	0
12	.05	0	.75	.20
13	.05	.80	0	.15
14	0	.60	.25	.15
15	.05	.80	.05	.10
		Percentile scaling		
5	.05	0	.95	0
6	0	.95	0	.05
7	.05	.95	0	0
8	.10	0	0	.90
9	0	.85	.15	0
10	0	0	.95	.05
11	.05	.45	.50	0
12	.05	0	.90	.05
13	.05	.75	0	.20
14	0	.85	.10	.05
15	.05	.40	.50	.05

Table 10: Optimally tuned weights for the normalized score fusion results presented in Table 6 (in-domain).

degraded performance compared to using the best model alone.

C.2 Weight Tuning in NSF

Table 9 an Table 10 present the optimal weights assigned to each retrieval system in zero-shot and in-domain contexts, respectively, when using normalized score fusion (NSF). These weights were meticulously determined through extensive tuning on the LLeQA dev set. Additionally, Figures 7 to 11 illustrate the variation in performance based on the weights assigned to pairs of retrieval systems.

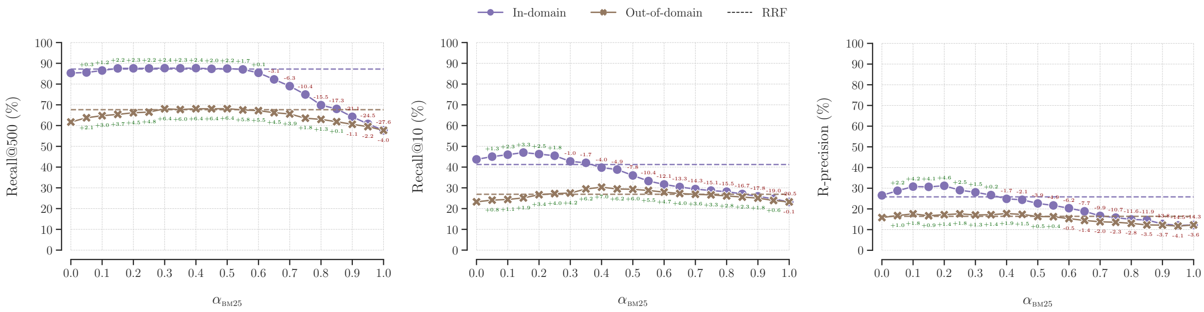


Figure 7: Effect of weight tuning in NSF between BM25 & ColBERT$_{\text{FR-\{LEX,BASE\}}}$ on LLeQA dev set.

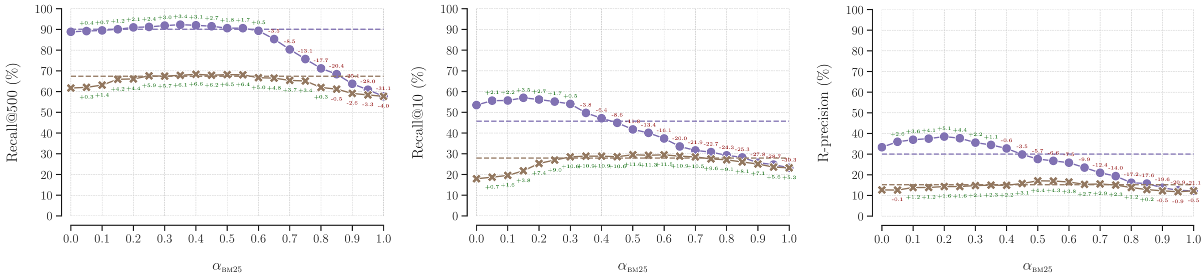


Figure 8: Effect of weight tuning in NSF between BM25 & SPLADE$_{\text{FR-\{LEX,BASE\}}}$ on LLeQA dev set.

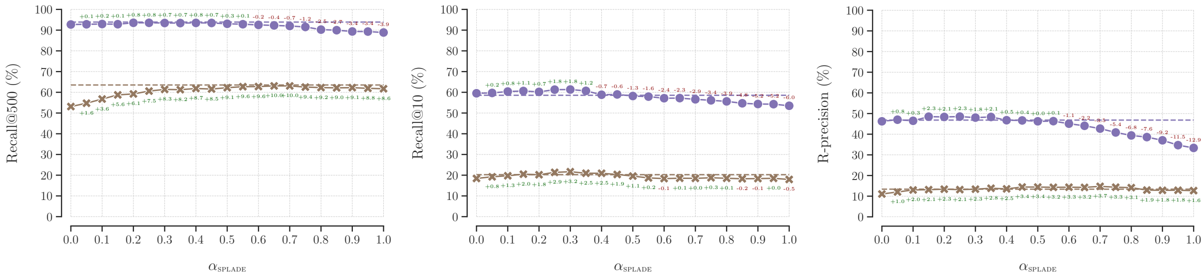


Figure 9: Effect of weight tuning in NSF between SPLADE$_{\text{FR-\{LEX,BASE\}}}$ & DPR$_{\text{FR-\{LEX,BASE\}}}$ on LLeQA dev set.

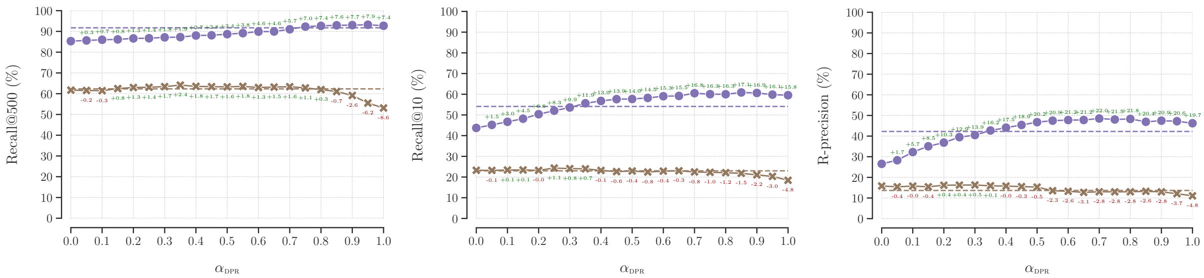


Figure 10: Effect of weight tuning in NSF between DPR$_{\text{FR-\{LEX,BASE\}}}$ & ColBERT$_{\text{FR-\{LEX,BASE\}}}$ on LLeQA dev set.

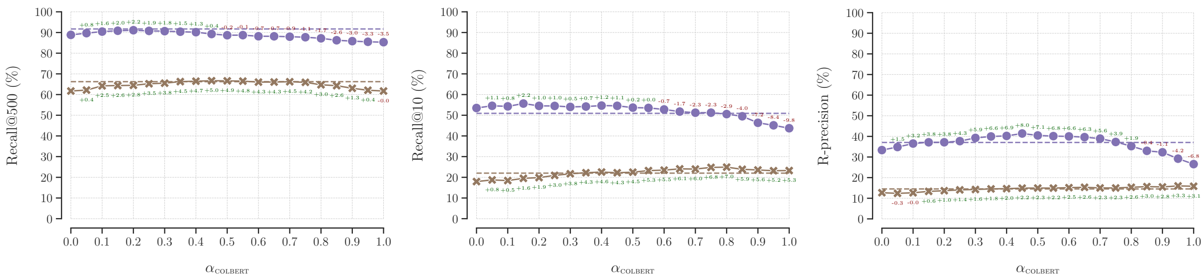


Figure 11: Effect of weight tuning in NSF between ColBERT$_{\text{FR-\{LEX,BASE\}}}$ & SPLADE$_{\text{FR-\{LEX,BASE\}}}$ on LLeQA dev set.