PARM: a Paragraph Aggregation Retrieval Model for dense document-to-document retrieval

Althammer, S.; Hofstätter, S.; Sertkan, M.; Verberne, S.; Hanbury, A.; Hagen, M.; ... ; Setty, V.

PARM: A Paragraph Aggregation Retrieval Model for Dense Document-to-Document Retrieval

Sophia Althammer[1(✉)], Sebastian Hofstätter[1], Mete Sertkan[1], Suzan Verberne[2], and Allan Hanbury[1]

[1] Institute for Information Systems Engineering, TU Wien, Vienna, Austria
{sophia.althammer,sebastian.hofstatter,mete.sertkan,
allan.hanbury}@tuwien.ac.at
[2] Leiden University, Leiden, The Netherlands
s.verberne@liacs.leidenuniv.nl

Abstract. Dense passage retrieval (DPR) models show great effectiveness gains in first stage retrieval for the web domain. However in the web domain we are in a setting with large amounts of training data and a query-to-passage or a query-to-document retrieval task. We investigate in this paper dense document-to-document retrieval with limited labelled target data for training, in particular legal case retrieval. In order to use DPR models for document-to-document retrieval, we propose a Paragraph Aggregation Retrieval Model (PARM) which liberates DPR models from their limited input length. PARM retrieves documents on the paragraph-level: for each query paragraph, relevant documents are retrieved based on their paragraphs. Then the relevant results per query paragraph are aggregated into one ranked list for the whole query document. For the aggregation we propose vector-based aggregation with reciprocal rank fusion (VRRF) weighting, which combines the advantages of rank-based aggregation and topical aggregation based on the dense embeddings. Experimental results show that VRRF outperforms rank-based aggregation strategies for dense document-to-document retrieval with PARM. We compare PARM to document-level retrieval and demonstrate higher retrieval effectiveness of PARM for lexical and dense first-stage retrieval on two different legal case retrieval collections. We investigate how to train the dense retrieval model for PARM on limited target data with labels on the paragraph or the document-level. In addition, we analyze the differences of the retrieved results of lexical and dense retrieval with PARM.

1 Introduction

Dense passage retrieval (DPR) models brought substantial effectiveness gains to information retrieval (IR) tasks in the web domain [14,19,39]. The promise of DPR models is to boost the recall of first stage retrieval by leveraging the semantic information for retrieval as opposed to traditional retrieval models [31], which rely on lexical matching. The web domain is a setting with query-to-passage or query-to-document retrieval tasks and a large amount of training data, while training data is much more limited in other domains. Furthermore we see recent advances in neural retrieval remain neglected for document-to-document retrieval despite the task's importance in several, mainly professional, domains [24,28–30].

M. Hagen et al. (Eds.): ECIR 2022, LNCS 13185, pp. 19–34, 2022.
https://doi.org/10.1007/978-3-030-99736-6_2

In this paper we investigate the effectiveness of dense retrieval models for document-to-document tasks, in particular legal case retrieval. We focus on first stage retrieval with dense models and therefore aim for a high recall. The first challenge for DPR models in document-to-document retrieval tasks is the input length of the query documents and of the documents in the corpus. In legal case retrieval the cases tend to be long documents [35] with an average length of 1269 words in the COLIEE case law corpus [29]. However the input length of DPR models is limited to 512 tokens [19] and theoretically bound of how much information of a long text can be compressed into a single vector [25]. Furthermore we reason in accordance with the literature [7,33,37,38] that relevance between two documents is not only determined by the complete text of the documents, but that a candidate document can be relevant to a query document based on one paragraph that is relevant to one paragraph of the query document. In the web domain DPR models are trained on up to $500k$ training samples [6], whereas in most domain-specific collections only a limited amount of hundreds of labelled samples is available [13,15,29].

In this paper we address these challenges by proposing a **paragraph aggregation retrieval model (PARM)** for dense document-to-document retrieval. PARM liberates dense passage retrieval models from their limited input length without increasing the computational cost. Furthermore PARM gives insight on which paragraphs the document-level relevance is based, which is beneficial for understanding and explaining the retrieved results. With PARM the documents are retrieved on the paragraph-level: the query document and the documents in the corpus are split up into their paragraphs and for each query paragraph a ranked list of relevant documents based on their paragraphs is retrieved. The ranked lists of documents per query paragraph need to be aggregated into one ranked list for the whole query document. As PARM provides the dense vectors of each paragraph, we propose **vector-based aggregation with reciprocal rank fusion weighting (VRRF)** for PARM. VRRF combines the merits of rank-based aggregation [10,16] with semantic aggregation with dense embeddings. We investigate:
RQ1 *How does VRRF compare to other aggregation strategies within PARM?*

We find that our proposed aggregation strategy of VRRF for PARM leads to the highest retrieval effectiveness in terms of recall compared to rank-based [10,34] and vector-based aggregation baselines [21]. Furthermore we investigate:
RQ2 *How effective is PARM with VRRF for document-to-document retrieval?*

We compare PARM with VRRF to document-level retrieval for lexical and dense retrieval methods on two different test collections for the document-to-document task of legal case retrieval. We demonstrate that PARM consistently improves the first stage retrieval recall for dense document-to-document retrieval. Furthermore, dense document-to-document retrieval with PARM and VRRF aggregation outperforms lexical retrieval methods in terms of recall at higher cut-off values.

The success of DPR relies on the size of labelled training data. As we have a limited amount of labelled data as well as paragraph and document-level labels we investigate:
RQ3 *How can we train dense passage retrieval models for PARM for document-to-document retrieval most effectively?*

For training DPR for PARM we compare training with relevance labels on the paragraph or document-level. We find that despite the larger size of document-level labelled datasets, the additional training data is not always beneficial compared to training DPR on smaller, but more accurate paragraph-level samples. Our contributions are:

- We propose a **paragraph aggregation retrieval model (PARM)** for dense document-to-document retrieval and demonstrate higher retrieval effectiveness for dense retrieval with PARM compared to retrieval without PARM and to lexical retrieval with PARM.
- We propose **vector-based aggregation with reciprocal rank fusion weighting (VRRF)** for dense retrieval with PARM and find that VRRF leads to the highest recall for PARM compared to other aggregation strategies.
- We investigate training DPR for PARM and compare the impact of fewer, more accurate paragraph-level labels to more, potentially noisy document-level labels.
- We publish the code at https://github.com/sophiaalthammer/parm

2 Related Work

Dense Passage Retrieval. Improving the first stage retrieval with DPR models is a rapidly growing area in neural IR, mostly focusing on the web domain. Karpukhin et al. [19] propose dense passage retrieval for open-domain QA using BERT models as bi-encoder for the query and the passage. With ANCE, Xiong et al. [39] train a DPR model for open-domain QA with sampling negatives from the continuously updated index. Efficiently training DPR models with distillation [17] and balanced topic aware sampling [18] has demonstrated to improve the retrieval effectiveness. As opposed to this prior work, we move from dense passage to dense document-to-document retrieval and propose PARM to use dense retrieval for document-to-document tasks.

Document Retrieval. The passage level influence for retrieval of documents has been analyzed in multiple works [7,22,37,38] and shown to be beneficial, but in these works the focus lies on passage-to-document retrieval. Cohan et al. [9] present document-level representation learning strategies for ranking, however the input length remains bounded by 512 tokens and only title and abstract of the document are considered. Abolghasemi et al. [1] present multi-task learning for document-to-document retrieval. Liu et al. [40] propose similar document matching for documents up to a length of 2048 however here the input length is still bounded and the computational cost of training and using the model is increased. Different to this prior work, the input length of PARM is not bounded without increasing the computational complexity of the retrieval.

Aggregation Strategies. Aggregating results from different ranked lists has a long history in IR. Shaw et al. [20,34] investigate the combination of multiple result lists by summing the scores. Different rank aggregation strategies like Condorcet [26] or Borda count [36] are proposed, however it is demonstrated [10,41] that reciprocal rank fusion outperforms them. Ai et al. [2] propose a neural passage model for scoring passages for a passage-to-document retrieval task. Multiple works [3,4,11,42] propose score aggregation for re-ranking with BERT on a passage-to-document task ranging from taking the first passage of a document to the passage of the document with the highest score. Different to rank/score-based aggregation approaches, Li et al. [21] propose vector-based aggregation for re-ranking for a passage-to-document task. Different to our approach they concatenate query and passage and learn a representation for binary classification of the relevance score. The focus of score/rank aggregation is mainly on federated search or passage-to-document tasks, however we focus on document-to-document retrieval. We have not seen a generalization of aggregation strategies for

the query and candidate paragraphs for document-to-document retrieval yet. Different to previous work, we propose to combine rank and vector-based aggregation methods for aggregating the representation of query and candidate documents independently.

3 Paragraph Aggregation Retrieval Model (PARM)

In this section we propose PARM as well as the aggregation strategy VRRF for PARM for dense document-to-document retrieval and training strategies.

3.1 Workflow

We use the DPR model [19] based on BERT [12] bi-encoders, of which one encodes the query passage q, the other one the candidate passage p. After storing the encoded candidate passages $\hat{p}$ in the index, the relevance score between a query q and a candidate passage p is computed by the dot-product between the encoded query passage $\hat{q}$ and $\hat{p}$.

As the input length of BERT [12] is limited to 512 tokens, the input length for the query and the candidate passage for DPR [19] is also limited by that. The length of query and candidate documents for document-to-document tasks exceeds this input length. For example the average length of a document is 1296 words for the legal case retrieval collection COLIEE [29]. We reason that for document-to-document tasks a single paragraph or multiple paragraphs can be decisive for the relevance of a document to another one [7,22,37,38] and that different paragraphs contain different topics of a document. Therefore we propose a **paragraph aggregation retrieval model (PARM)**, in order to use DPR models for dense document-to-document retrieval. PARM retrieves relevant documents based on the paragraph-level relevance.

The workflow of PARM is visualized in Fig. 1. For the documents in the corpus we split each document d into paragraphs $p_1, ..., p_{m_d}$ with m_d the number of paragraphs of document d. We take the paragraphs of the document as passages for DPR. We index each paragraph p_j, $j \in 1, ..., m_d$ of each document d in the corpus and attain a paragraph-level index containing the encoded paragraphs $\hat{p}_j$ for all documents d in the corpus. At query time, the query document q is also split up into paragraphs $q_1, ..., q_{n_q}$,

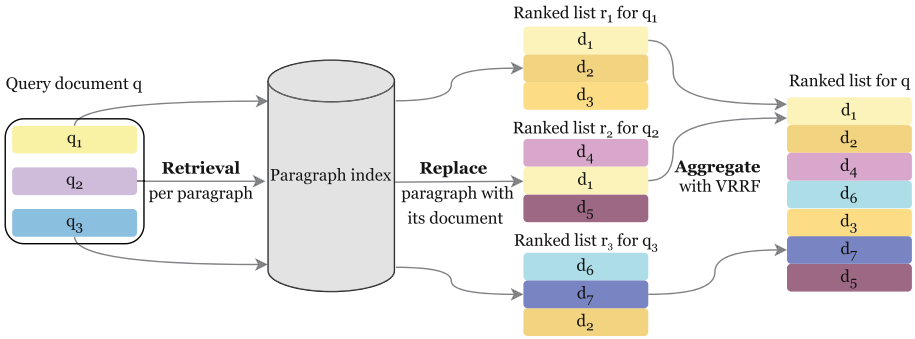

Fig. 1. PARM workflow for query document q and retrieved documents $d_1, .., d_7$

where n_q is the number of paragraphs of q. For each query paragraph q_i with $i \in 1, ..., n_q$ the top N most relevant paragraphs are retrieved from the paragraph-level corpus. The result is a ranked list r_i with $i \in 1, ..., n_q$ per query paragraph q_i with N relevant paragraphs. The paragraphs in the ranked lists r_i with $i \in 1, ..., n_q$ are replaced by the documents that contain the paragraphs. Therefore it is possible that one document occurs multiple times in the list. In order to attain one ranked list for the whole query document q, the ranked paragraph lists of retrieved documents $r_1, ..., r_{n_q}$ of each query paragraph q_i with $i \in 1, ..., n_q$ need to be aggregated to one ranked list.

3.2 Vector-Based Aggregation with Reciprocal Rank Fusion Weighting (VRRF)

Multiple works have demonstrated the benefit of reciprocal rank fusion [10,16,27] for rank-based aggregation of multiple ranked retrieved lists. Using dense retrieval with PARM we have more information than the ranks and scores of the retrieved paragraphs: we have dense embeddings, which encode the semantic meaning of the paragraphs, for each query paragraph and the retrieved paragraphs. In order to make use of this additional information for aggregation, we propose **vector-based aggregation with reciprocal rank fusion weighting (VRRF)**, which extends reciprocal rank fusion for neural retrieval. VRRF combines the advantages of reciprocal rank fusion with relevance signals of semantic aggregation using the dense vector embeddings.

In **VRRF** we aggregate documents using the dense embeddings $\hat{p_i}$ of the passages p_i, which are from the same document d and which are in the retrieved list r_i with $i \in 1, ..., n_q$, with a weighted sum, taking the reciprocal rank fusion score [10] as weight. The dense embeddings $\hat{q_i}$ of each query paragraph q_i with $i \in 1, ..., n_q$ are aggregated by adding the embeddings without a weighting:

$$\hat{q} = \sum_{i=1}^{n_q} \hat{q_i} \qquad \hat{d} = \sum_{i=1}^{n_q} \sum_{p \in d, d \in r_i} rrf(q_i, p_i)\, \hat{p_i}$$

We compute the relevance score between query and candidate document with the dot-product between the aggregated embedding of query $\hat{q}$ and candidate document $\hat{d}$.

To confirm the viability of VRFF aggregation, we propose simple baselines: **VRanks** and **VScores**, where the paragraph embeddings $\hat{p_i}$ of d are aggregated with the rank or the score of the passage p_i as weight.

3.3 Training Strategies

As we have a limited amount of labelled target data, we examine how to effectively train a DPR model for PARM with the training collections at hand. We assume that we have test collections consisting of documents with clearly identifiable paragraphs, with relevance assessments on either the paragraph or the document-level.

Paragraph-Level Training. For the paragraph-level labelled training we take the relevant paragraphs in the training set as positives and sample random negatives from the paragraphs in the corpus. Here we sample as many negatives as we have positive samples for each query paragraph, thereby balancing the training data.

Document-Level Training. For the document-level labelled training the collection contains query documents and a corpus of documents with relevance assessments for each query document. We sample negative documents randomly from the corpus. In order to use the document-level labelled collection for training the DPR model, we split up the query document as well as the positive documents into its paragraphs and consider each paragraph of the query document relevant to each paragraph of each positive document. Equivalently we consider each paragraph of a negative document irrelevant to each query paragraph. As on average each document in the COLIEE dataset [29] contains 42.44 paragraphs, one relevant document leads to $42 \cdot 20 = 840$ paragraph-level labels containing one positive and one negative sample to a query paragraph. Therefore this method greatly increases the number of paragraph-level annotations, however this comes with the risk of potentially noisy labels [5].

4 Experiment Design

4.1 Training and Test Collections

We focus on the document-to-document task of legal case retrieval because of the importance for the legal domain [23,24,32,33] which facilitates the availability of training collections with relevance annotations on the paragraph and the document-level [29]. For training the DPR models, we introduce paragraph and document-level labelled collections. For the evaluation we use the document-level collections.

Paragraph-Level Labelled Collections. COLIEE [29] is a competition for legal information extraction and retrieval which provides datasets for legal case retrieval and case entailment. Task 2 of COLIEE 2020 [29] provides a training and test collection for legal case entailment. It contains relevance labels on the legal case paragraph level, given a query claim, a set of candidate claims to the query claim as well as relevance labels for the candidate claims. We denote these sets with *COLIEEPara train/test*.

Document-Level Labelled Collections. In Task 1 of COLIEE 2021 [29], the legal case retrieval task, query cases with their relevance judgements on the document-level are provided together with a corpus of candidate documents. We divide the training set of COLIEEDoc into a training and validation set. The validation set contains the last 100 queries of the training set from query case 550 to 650. We will denote the training, validation and test collection with *COLIEEDoc train/val/test*. For a broader evaluation, we evaluate our models additionally on the CaseLaw collection [24]. It contains a corpus of legal cases, query cases and their relevance judgements for legal case retrieval.

Data Pre-processing. For COLIEEDoc, we remove the French versions of the cases, we divide the cases into an introductory part, a summary, if it contains one, and its claims, which are indicated by their numbering. As indicated in Table 1, the paragraphs have an average length of 84 words and 96.2% of the paragraphs are not longer than 512 words. The CaseLaw dataset is split along the line breaks of the text and merged to paragraphs by concatenating sentences until the paragraphs exceed the length of 200 words.

4.2 Baselines

As baseline we use the lexical retrieval model BM25 [31]. For BM25 we use ElasticSearch[1] with parameters $k = 1.3$ and $b = 0.8$, which we optimized on COLIEE-Docval. *VRRF aggregation for PARM (RQ1).* In order to investigate the retrieval effectiveness of our proposed aggregation strategy VRFF for PARM, we compare VRRF to the commonly used score-based aggregation strategy CombSum [34] and rank-based aggregation strategy of reciprocal rank fusion (RRF) [10] for PARM. As baselines for vector-based aggregation, we investigate VSum, VMin, VMax, VAvg, which are originally proposed by Li et al. [21] for re-ranking on a passage-to-document retrieval task. In order to use VSum, VMin, VMax, VAvg in the context of PARM, we aggregate independently the embeddings of both, the query and the candidate document. In contrast to Li et al. [21] we aggregate the query and paragraph embeddings independently and score the relevance between aggregated query and aggregated candidate embedding after aggregation. The learned aggregation methods of CNN and Transformer proposed by Liu et al. [21] are therefore not applicable to PARM, as they learn a classification on the embedding of the concatenated query and paragraph.

Table 1. Statistics of paragraph- and document-level labelled collections.

Labels	**Dataset**	**Train/ Test**	**Statistics**					
			# queries	∅ # docs	∅ # rel docs	∅ para length	% para < 512 words	∅ # para
Para	COLIEEPara	Train	325	32.12	1.12	102	95.5%	-
	COLIEEPara	Test	100	32.19	1.02	117	95.2%	-
Doc	COLIEEDoc	Train	650	4415	5.17	84	96.2%	44.6
	COLIEEDoc	Test	250	4415	3.60	92	97.8%	47.5
	CaseLaw	Test	100	63431	7.2	219	91.3%	7.5

PARM VRRF for Dense Document-to-Document Retrieval (RQ2). In order to investigate the retrieval effectiveness of PARM with VRRF for dense document-to-document retrieval, we compare PARM to document-level retrieval on two document-level collections (COLIEEDoc and CaseLaw). Because of the limited input length, the document-level retrieval either reduces to retrieval based on the First Passage (FirstP) or the passage of the document with the maximum score (MaxP) [3,42]. In order to separate the impact of PARM for lexical and dense retrieval methods, we also use PARM with BM25 as baseline. For PARM with BM25 we also investigate which aggregation strategy leads to the highest retrieval effectiveness in order to have a strong baseline. As BM25 does not provide dense embeddings only rank-based aggregation strategies are applicable.

[1] https://github.com/elastic/elasticsearch.

Paragraph and Document-Level Labelled Training (RQ3). We train a DPR model on a paragraph- and another document-level labelled collection and compare the retrieval performance of PARM for document-to-document retrieval. As bi-encoders for DPR we choose BERT [12] and LegalBERT [8]. We train DPR on the paragraph-level labelled collection COLIEEPara train and additionally on the document-level labelled collection COLIEEDoc train as described in Sect. 3.3. We use the public code[2] and train DPR according to Karpukhin et al. [19]. We sample the negative paragraphs randomly from randomly sampled negative documents and take the 20 paragraphs of a positive document as positive samples, which have the highest BM25 score to the query paragraph. This training procedure lead to the highest recall compared to training with all positive paragraphs or with BM25 sampled negative paragraphs. We also experimented with the DPR model pre-trained on open-domain QA as well as TAS-balanced DPR model [18], but initial experiments did not show a performance improvement. We train each DPR model for 40 epochs and take the best checkpoint according to COLIEEPara test/COLIEEDoc val. We use batch size of 22 and a learning rate of $2 * 10^{-5}$, after comparing three commonly used learning rates ($2 * 10^{-5}$, $1 * 10^{-5}$, $5 * 10^{-6}$) for [19].

5 Results and Analysis

We evaluate the first stage retrieval performance with nDCG@10, recall@100, recall@500 and recall@1k using pytrec_eval. We focus our evaluation on recall because the recall performance of the first stage retrieval bounds the ranking performance after re-ranking the results in the second stage for a higher precision. We do not compare our results to the reported state-of-the-art results as they rely on re-ranked results and do not report evaluation results after the first stage retrieval.

5.1 RQ1: VRRF Aggregation for PARM

As we propose vector-based aggregation with reciprocal rank fusion weighting (VRRF) for PARM, we first investigate:

(**RQ1**) *How does VRRF compare to other aggregation strategies within PARM?*
We compare VRRF, which combines dense-vector-based aggregation with rank-based weighting, to score/rank-based and vector-based aggregation methods for PARM. The results in Table 2 show that VRRF outperforms all rank and vector-based aggregation approaches for the dense retrieval results of DPR PARM with BERT and LegalBERT. For the lexical retrieval BM25 with PARM, only rank-based aggregation approaches are feasible, here RRF shows the best performance, which will be our baseline for RQ2.

[2] https://github.com/facebookresearch/DPR.

Table 2. Aggregation comparison for PARM on COLIEEval, VRRF shows best results for dense retrieval, stat. sig. difference to RRF w/ paired t-test ($p < 0.05$) denoted with †, Bonferroni correction with $n = 7$. For BM25 only rank-based methods applicable.

Aggregation	BM25			DPR BERT			DPR LegalBERT		
	R@100	R@500	R@1K	R@100	R@500	R@1K	R@100	R@500	R@1K
Rank-based									
CombSum [34]	.5236	.7854	.8695	.4460	.7642	.8594	.5176	.7975	.8882
RRF [10]	**.5796**	**.8234**	**.8963**	.5011	.8029	.8804	**.5830**	.8373	.9049
Vector-based									
VAvg [21]	-	-	-	.1908†	.4668†	.6419†	.2864†	.4009†	.7466†
VMax [21]	-	-	-	.3675†	.6992†	.8273†	.4071†	.6587†	.8418†
VMin [21]	-	-	-	.3868†	.6869†	.8295†	.4154†	.6423†	.8465†
VSum [21]	-	-	-	.4807	.7496†	.8742	.5182†	.8069	.8882
Vector-based with rank-based weights (Ours)									
VScores	-	-	-	.4841	.7616†	.8709	.5195†	.8075†	.8882†
VRanks	-	-	-	.4826	.7700†	.8804	.5691†	.8212	.8980
VRRF	-	-	-	**.5035**	**.8062**†	**.8806**	**.5830**†	**.8386**†	**.9091**†

Table 3. Document-to-document retrieval results for PARM and Document-level retrieval. No comparison to results reported in prior work as those rely on re-ranking, while we evaluate only first stage retrieval evaluation. *nDCG cutoff at 10, stat. sig. difference to BM25 Doc w/ paired t-test ($p < 0.05$) denoted with † and Bonferroni correction with $n = 12$, effect size > 0.2 denoted with ‡.*

Model	Retrieval	COLIEEDoc test				CaseLaw			
		nDCG	R@100	R@500	R@1K	nDCG	R@100	R@500	R@1K
BM25									
BM25	Doc	**.2435**	.6231	.7815	.8426	**.2653**	.4218	.5058	.5438
	PARM RRF	.1641†‡	**.6497**†‡	.8409†‡	.8944†‡	.0588†‡	.3362†‡	.5716†‡	.6378†‡
DPR									
BERT para	Doc FirstP	.0427†‡	.3000†‡	.5371†‡	.6598†‡	.0287†‡	.0871†‡	.1658†‡	.2300†‡
	Doc MaxP	.0134†‡	.1246†‡	.5134†‡	.6201†‡	.0000†‡	.0050†‡	.4813†‡	.4832†‡
	PARM RRF	.0934†‡	.5765†‡	.8153†‡	.8897†‡	.0046†‡	.1720†‡	.5019†‡	.5563†
	PARM VRRF	.0952†‡	.5786†‡	.8132†‡	.8909†‡	.1754†‡	.3855†‡	.5328†‡	.5742†‡
LegalBERT para	Doc FirstP	.0553†‡	.2447†‡	.4598†‡	.5657†‡	.0397†‡	.0870†‡	.1844†‡	.2248†‡
	Doc MaxP	.0073†‡	.0737†‡	.3970†‡	.5670†‡	.0000†‡	.0050†‡	.4846†‡	.4858†‡
	PARM RRF	.1280†‡	.6370	.8308†‡	.8997†‡	.0177†‡	.2595†‡	.5446†‡	.6040†‡
	PARM VRRF	.1280†‡	.6396	.8310†‡	.9023†‡	.0113†‡	**.4986**†‡	.5736†‡	.6340†‡
LegalBERT doc	Doc FirstP	.0682†‡	.3881†‡	.6187†‡	.7361†‡	.0061†‡	.0050†‡	.4833†‡	.4866†‡
	Doc MaxP	.0008†‡	.0302†‡	.2069†‡	.2534†‡	.0022†‡	.0050†‡	.4800†‡	.4833†‡
	PARM RRF	.1248†‡	.6086†	.8394†‡	.9114†‡	.0117†‡	.2277†‡	.5637†‡	.6265†‡
	PARM VRRF	.1256†‡	.6127†	**.8426**†‡	**.9128**†‡	.2284†‡	.4620†‡	**.5847**†‡	**.6402**†‡

5.2 RQ2: PARM VRRF vs Document-Level Retrieval

As we propose PARM VRRF for document-to-document retrieval, we investigate:
(**RQ2**) *How effective is PARM with VRRF for document-to-document retrieval?*
We evaluate and compare PARM and document-level retrieval for lexical and dense retrieval methods on the two test collections (COLIEEDoc and CaseLaw) for document-to-document retrieval in Table 3. For BM25 we find that PARM-based retrieval outperforms document-level retrieval at each recall stage, except for R@100 on CaseLaw.

For dense retrieval we evaluate DPR models with BERT trained solely on the paragraph-level labels and with LegalBERT trained on the paragraph-level labels (denoted with LegalBERT para) and with additional training on the document-level labels (denoted with LegalBERT doc). For dense document-to-document retrieval PARM consistently outperforms document-level retrieval for all performance metrics for both test collections. Furthermore PARM aggregation with VRRF outperforms PARM RRF in nearly all cases. Overall we find that LegalBERTdoc-based dense retrieval with PARM VRRF achieves the highest recall at high ranks. When comparing the nDCG@10 evaluation we find that PARM lowers the nDCG@10 score for BM25 as well as for dense retrieval. Therefore we suggest that PARM is beneficial for first stage retrieval, so that in the re-ranking stage the overall ranking can be improved.

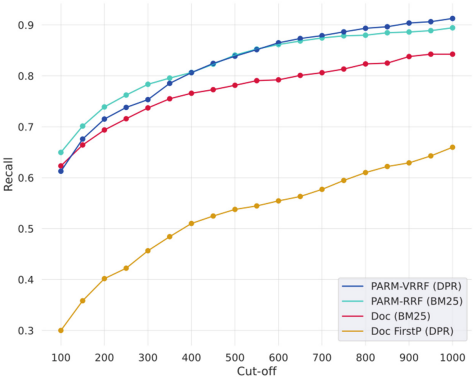

Fig. 2. Recall at different cut-off values for PARM-VRRF (DPR) and PARM-RRF (BM25) and Document-level retrieval with BM25 and DPR for COLIEEDoc test.

	COLIEEDoc		CaseLaw	
	BM25	DPR	BM25	DPR
Total				
relevant	900	900	720	720
PARM	892	896	578	545
Doc	751	662	419	199
Sets				
PARM ∩ Doc	750	661	417	196
PARM\Doc	142	235	161	349
Doc\PARM	1	1	2	3

Fig. 3. Number of relevant documents retrieved in comparison between PARM and Doc-level retrieval for COLIEEDoc and CaseLaw with BM25 or LegalBERT_doc-based DPR.

In Fig. 2 we show the recall at different cut-off values for PARM-VRRF with DPR (based on LegalBERTdoc) and PARM-RRRF with BM25 compared to document-level retrieval (Doc FirstP) of BM25/DPR. When comparing PARM to document-retrieval, we can see a clear gap between the performance of document-level retrieval and PARM for BM25 and for DPR. Furthermore we see that dense retrieval (PARM-VRRF DPR) outperforms lexical retrieval (PARM-RRF BM25) at cut-off values above 500.

In order to analyze the differences between PARM and document-level retrieval further, we analyze in Fig. 3, how many relevant documents are retrieved with PARM or with document-level retrieval with lexical (BM25) or dense methods (DPR). Furthermore we investigate how many relevant documents are retrieved by both PARM and document-level retrieval (PARM ∩ Doc), and how many relevant documents are retrieved only with PARM and not with document-level retrieval (PARM\Doc) and vice versa (Doc\PARM). When comparing the performance of PARM and document-level retrieval, we find that PARM retrieves more relevant documents in total for both test collections. PARM retrieves 142–380 of the relevant documents that did not get retrieved with document-level retrieval (PARM\Doc), which are 15–52% of the total number of relevant documents. This analysis demonstrates that PARM largely retrieves many of relevant documents that are not retrieved with document-level retrieval. We conclude that PARM is not only beneficial for dense but also for lexical retrieval.

5.3 RQ3: Paragraph-Level vs Document-level Labelled Training

As labelled in-domain data for document-to-document retrieval tasks is limited, we ask: (**RQ3**) *How can we train dense passage retrieval models for PARM for document-to-document retrieval most effectively?* We compare the retrieval performance for BERT-based and LegalBERT-based dense retrieval models in Table 4, which are either trained solely on the paragraph-level labelled collection or additionally trained on the document-level labelled collection. The upper part of the table shows that for BERT the additional training data on document-level improves the retrieval performance for document-level retrieval, but harms the performance for PARM RRF and PARM VRRF. For LegalBERT the additional document-level training data highly improves the performance of document-retrieval. For PARM the recall is improved at higher cut-off values (@500, @1000) for a cut-off. Therefore we consider the training on document-level labelled data beneficial for dense retrieval based on LegalBERT. This reveals that it is not always better to have more, potentially noisy data, for BERT-based dense retrieval the training with fewer, but accurate paragraph-level labels is more beneficial for overall document-to-document retrieval with PARM.

5.4 Analysis of Paragraph Relations

With our proposed paragraph aggregation retrieval model for dense document-to-document retrieval we can analyze on which paragraphs the document-level relevance is based. To gain more insight in what the dense retrieval model learned to retrieve on the paragraph-level with PARM, we analyze which query paragraph retrieves which paragraphs from relevant documents with dense retrieval with PARM and compare it to lexical retrieval with PARM. In Fig. 4, a heatmap visualizes which query paragraph how often retrieves which paragraph from a relevant document with PARM BM25 or PARM DPR on the COLIEEDoc test set. As introduced in Sect. 4.1, the legal cases in COLIEEDoc contain an introduction, a summary and claims as paragraphs. For the introduction (I) and the summary (S) we see the paragraph relation for lexical and dense retrieval that both methods retrieve also more introductions and summaries from the relevant documents. We reason this is due to the special structure of the introduction and

Table 4. Paragraph- and document-level labelled training of DPR. Document-level labelled training improves performance at high ranks for LegalBERT.

Model	Retrieval	Train Labels	COLIEEDoc val R@100	R@200	R@300	R@500	R@1K
DPR Retrieval							
BERT	Doc FirstP	para	.3000	.4018	.4566	.5371	.6598
	Doc FirstP	+ doc	.3800	.4641	.5160	.6054	.7211
	PARM RRF	para	.5765	.6879	.7455	.8153	.8897
	PARM RRF	+ doc	.5208	.6502	.7100	.7726	.8660
	PARM VRRF	para	.5786	.6868	.7505	.8132	.8909
	PARM VRRF	+ doc	.5581	.6696	.7298	.7970	.8768
LegalBERT	Doc FirstP	para	.2447	.3286	.3853	.4598	.5657
	Doc FirstP	+ doc	.3881	.4665	.5373	.6187	.7361
	PARM RRF	para	.6350	.7323	.7834	.8308	.8997
	PARM RRF	+ doc	.6086	.7164	.7561	.8394	.9114
	PARM VRRF	para	**.6396**	**.7325**	**.7864**	.8310	.9023
	PARM VRRF	+ doc	.6098	.7152	.7520	**.8396**	**.9128**

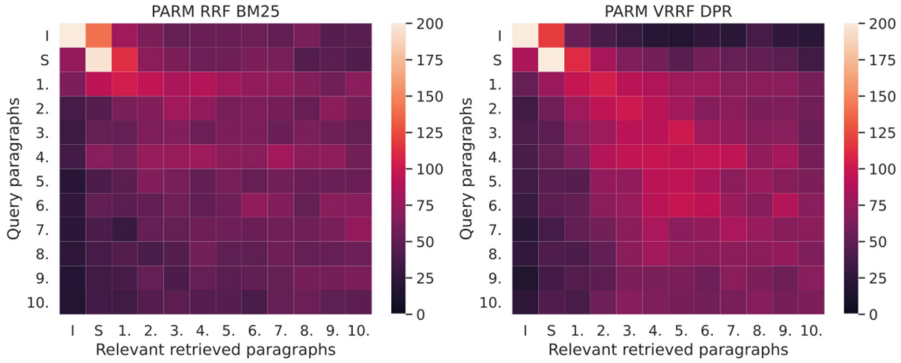

Fig. 4. Heatmap for PARM retrieval with BM25 or DPR visualizing which query paragraph how often retrieves which paragraph from a relevant document. I denotes the introduction, S the summary, 1.–10. denote the claims 1.–10. of COLIEEDoc test.

the summary which is distinct to the claims. For the query paragraphs 1.–10. we see that PARM DPR seems to focus on to the diagonal different to PARM BM25. This means for example that the first paragraph retrieves more first paragraphs from relevant documents than they retrieve other paragraphs. As the claim numbers are removed in the data pre-processing, this focus relies on the textual content of the claims. This paragraph relation suggests that there is a topical or hierarchical structure in the claims of legal cases, which is learned by DPR and exhibited with PARM. This structural component can not be exhibited with document-level retrieval.

6 Conclusion

In this paper we address the challenges of using dense passage retrieval (DPR) in first stage retrieval for document-to-document tasks with limited labelled data. We propose the paragraph aggregation retrieval model (PARM), which liberates dense passage retrieval models from their limited input length and which takes the paragraph-level relevance for document retrieval into account. We demonstrate on two test collections higher first stage recall for dense document-to-document retrieval with PARM than with document-level retrieval. We also show that dense retrieval with PARM outperforms lexical retrieval with BM25 in terms of recall at higher cut-off values. As part of PARM we propose the novel vector-based aggregation with reciprocal rank fusion weighting (VRFF), which combines the advantages of rank-based aggregation with RRF [10] and topical aggregation with dense embeddings. We demonstrate the highest retrieval effectiveness for PARM with VRRF aggregation compared to rank and vector-based aggregation baselines. Furthermore we investigate how to train dense retrieval models for dense document-to-document retrieval with PARM. We find the interesting result that training DPR models on more, but noisy document-level data does not always lead to overall higher retrieval performance compared to training on less, but more accurate paragraph-level labelled data. Finally, we analyze how PARM retrieves relevant paragraphs and find that the dense retrieval model learns a structural paragraph relation which it exhibits with PARM and therefore benefits the retrieval effectiveness.

References

1. Abolghasemi, A., Verberne, S., Azzopardi, L.: Improving BERT-based query-by-document retrieval with multi-task optimization. In: Hagen, M. et al. (Eds.) ECIR 2022. LNCS, vol. 13185, pp. xx–yy. Springer, Heidelberg (2022). https://doi.org/10.1007/978-3-030-99736-6_2
2. Ai, Q., O'Connor, B., Croft, W.B.: A neural passage model for ad-hoc document retrieval. In: Pasi, G., Piwowarski, B., Azzopardi, L., Hanbury, A. (eds.) ECIR 2018. LNCS, vol. 10772, pp. 537–543. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-76941-7_41
3. Akkalyoncu Yilmaz, Z., Wang, S., Yang, W., Zhang, H., Lin, J.: Applying BERT to document retrieval with birch. 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): System Demonstrations, Hong Kong, China, November 2019, pp. 19–24. Association for Computational Linguistics (2019). https://doi.org/10.18653/v1/D19-3004. https://aclanthology.org/D19-3004
4. Akkalyoncu Yilmaz, Z., Yang, W., Zhang, H., Lin, J.: Cross-domain modeling of sentence-level evidence for document retrieval. 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, November 2019, pp. 3490–3496. Association for Computational Linguistics (2019). https://doi.org/10.18653/v1/D19-1352. https://aclanthology.org/D19-1352

5. Akkalyoncu Yilmaz, Z., Yang, W., Zhang, H., Lin, J.: Cross-domain modeling of sentence-level evidence for document retrieval. 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, November 2019, pp. 3490–3496. Association for Computational Linguistics (2019). https://doi.org/10.18653/v1/D19-1352. https://www.aclweb.org/anthology/D19-1352
6. Bajaj, P., et al.: MS MARCO: a human generated MAchine Reading COmprehension dataset. In: Proceedings of the NIPS (2016)
7. Bendersky, M., Kurland, O.: Utilizing passage-based language models for document retrieval. In: Macdonald, C., Ounis, I., Plachouras, V., Ruthven, I., White, R.W. (eds.) ECIR 2008. LNCS, vol. 4956, pp. 162–174. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-78646-7_17
8. Chalkidis, I., Fergadiotis, M., Malakasiotis, P., Aletras, N., Androutsopoulos, I.: LEGAL-BERT: the Muppets straight out of law school. In: Findings of the Association for Computational Linguistics, EMNLP 2020, Online, November 2020, pp. 2898–2904. Association for Computational Linguistics (2020). https://doi.org/10.18653/v1/2020.findings-emnlp.261. https://www.aclweb.org/anthology/2020.findings-emnlp.261
9. Cohan, A., Feldman, S., Beltagy, I., Downey, D., Weld, D.: SPECTER: document-level representation learning using citation-informed transformers. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, July 2020, pp. 2270–2282. Association for Computational Linguistics (2020). https://doi.org/10.18653/v1/2020.acl-main.207. https://aclanthology.org/2020.acl-main.207
10. Cormack, G.V., Clarke, C.L.A., Buettcher, S.: Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In: Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2009, pp. 758–759. Association for Computing Machinery, New York (2009). https://doi.org/10.1145/1571941.1572114
11. Dai, Z., Callan, J.: 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, SIGIR 2019, pp. 985–988. Association for Computing Machinery, New York (2019). https://doi.org/10.1145/3331184.3331303
12. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: 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, June 2019, pp. 4171–4186. Association for Computational Linguistics (2019). https://doi.org/10.18653/v1/N19-1423. https://www.aclweb.org/anthology/N19-1423
13. Gao, J., et al.: FIRE 2019@AILA: legal retrieval based on information retrieval model. In: Proceedings of the Forum for Information Retrieval Evaluation, FIRE 2019 (2019)
14. Gao, L., Dai, Z., Chen, T., Fan, Z., Durme, B.V., Callan, J.: Complementing lexical retrieval with semantic residual embedding. arXiv arXiv:2004.13969 (April 2020)
15. Hedin, B., Zaresefat, S., Baron, J., Oard, D.: Overview of the TREC 2009 legal track. In: Proceedings of the 18th Text REtrieval Conference, TREC 2009 (January 2009)
16. García Seco de Herrera, A., Schaer, R., Markonis, D., Müller, H.: Comparing fusion techniques for the ImageCLEF 2013 medical case retrieval task. Comput. Med. Imaging Graph. **39**, 46–54 (2014). http://publications.hevs.ch/index.php/attachments/single/676
17. Hofstätter, S., Althammer, S., Schröder, M., Sertkan, M., Hanbury, A.: Improving efficient neural ranking models with cross-architecture knowledge distillation (2021)
18. Hofstätter, S., Lin, S.C., Yang, J.H., Lin, J., Hanbury, A.: Efficiently teaching an effective dense retriever with balanced topic aware sampling (2021)

19. Karpukhin, V., et al.: Dense passage retrieval for open-domain question answering. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 6769–6781. Association for Computational Linguistics, Online, November 2020 (2020). https://doi.org/10.18653/v1/2020.emnlp-main.550. https://www.aclweb.org/anthology/2020.emnlp-main.550
20. Lee, J.H.: Analyses of multiple evidence combination. SIGIR Forum **31**(SI), 267–276 (1997). https://doi.org/10.1145/278459.258587
21. Li, C., Yates, A., MacAvaney, S., He, B., Sun, Y.: Parade: passage representation aggregation for document reranking. arXiv preprint arXiv:2008.09093 (2020)
22. Liu, X., Croft, W.B.: Passage retrieval based on language models. In: Proceedings of the 11th International Conference on Information and Knowledge Management, CIKM 2002, pp. 375–382. Association for Computing Machinery, New York (2002). https://doi.org/10.1145/584792.584854
23. Locke, D., Zuccon, G.: A test collection for evaluating legal case law search. In: 41st International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2018, pp. 1261–1264. Association for Computing Machinery, Inc. (June 2018). https://doi.org/10.1145/3209978.3210161
24. Locke, D., Zuccon, G., Scells, H.: Automatic query generation from legal texts for case law retrieval. In: Sung, W.-K., et al. (eds.) Information Retrieval Technology, pp. 181–193. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-70145-5_14
25. Luan, Y., Eisenstein, J., Toutanova, K., Collins, M.: Sparse, dense, and attentional representations for text retrieval. arXiv preprint arXiv:2005.00181 (2020)
26. Montague, M., Aslam, J.A.: Condorcet fusion for improved retrieval. In: Proceedings of the 11th International Conference on Information and Knowledge Management, CIKM 2002, pp. 538–548. Association for Computing Machinery, New York (2002). https://doi.org/10.1145/584792.584881
27. Mourão, A., Martins, F., Magalhães, J.: Multimodal medical information retrieval with unsupervised rank fusion. Comput. Med. Imaging Graph. **39**, 35–45 (2015). Medical visual information analysis and retrieval. https://doi.org/10.1016/j.compmedimag.2014.05.006. https://www.sciencedirect.com/science/article/pii/S0895611114000664
28. Piroi, F., Tait, J.: CLEF-IP 2010: retrieval experiments in the intellectual property domain. In: Proceedings of CLEF 2010 (2010)
29. Rabelo, J., Kim, M.-Y., Goebel, R., Yoshioka, M., Kano, Y., Satoh, K.: A summary of the COLIEE 2019 competition. In: Sakamoto, M., Okazaki, N., Mineshima, K., Satoh, K. (eds.) JSAI-isAI 2019. LNCS (LNAI), vol. 12331, pp. 34–49. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58790-1_3
30. Risch, J., Alder, N., Hewel, C., Krestel, R.: PatentMatch: a dataset for matching patent claims & prior art (2020)
31. Robertson, S., Zaragoza, H.: The probabilistic relevance framework: BM25 and beyond. Found. Trends Inf. Retr. **3**(4), 333–389 (2009). https://doi.org/10.1561/1500000019
32. Shao, Y., Liu, B., Mao, J., Liu, Y., Zhang, M., Ma, S.: THUIR@COLIEE-2020: leveraging semantic understanding and exact matching for legal case retrieval and entailment. CoRR abs/2012.13102 (2020). https://arxiv.org/abs/2012.13102
33. Shao, Y., et al.: BERT-PLI: modeling paragraph-level interactions for legal case retrieval. In: Bessiere, C. (ed.) Proceedings of the 29th International Joint Conference on Artificial Intelligence, IJCAI-20, pp. 3501–3507. International Joint Conferences on Artificial Intelligence Organization (July 2020). Main track. https://doi.org/10.24963/ijcai.2020/484
34. Shaw, J.A., Fox, E.A.: Combination of multiple searches. In: The 2nd Text Retrieval Conference, TREC-2, pp. 243–252 (1994)

35. Van Opijnen, M., Santos, C.: On the concept of relevance in legal information retrieval. Artif. Intell. Law **25**(1), 65–87 (2017). https://doi.org/10.1007/s10506-017-9195-8
36. Wu, S.: Ranking-based fusion. In: Data Fusion in Information Retrieval. Adaptation, Learning, and Optimization, vol. 13, pp 135–147. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-28866-1_7
37. Wu, Z., et al.: Leveraging passage-level cumulative gain for document ranking. In: Proceedings of the Web Conference 2020, WWW 2020, pp. 2421–2431. Association for Computing Machinery, New York (2020). https://doi.org/10.1145/3366423.3380305
38. Wu, Z., Mao, J., Liu, Y., Zhang, M., Ma, S.: Investigating passage-level relevance and its role in document-level relevance judgment. In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, pp. 605–614. Association for Computing Machinery, New York (2019). https://doi.org/10.1145/3331184.3331233
39. Xiong, L., et al.: Approximate nearest neighbor negative contrastive learning for dense text retrieval. In: International Conference on Learning Representations (2021). https://openreview.net/forum?id=zeFrfgyZln
40. Yang, L., Zhang, M., Li, C., Bendersky, M., Najork, M.: Beyond 512 tokens: Siamese multi-depth transformer-based hierarchical encoder for long-form document matching. In: Proceedings of the 29th ACM International Conference on Information & Knowledge Management, CIKM 2020, pp. 1725–1734. Association for Computing Machinery, New York (2020). https://doi.org/10.1145/3340531.3411908
41. Zhang, X., Yates, A., Lin, J.: Comparing score aggregation approaches for document retrieval with pretrained transformers. In: Hiemstra, D., Moens, M.-F., Mothe, J., Perego, R., Potthast, M., Sebastiani, F. (eds.) ECIR 2021. LNCS, vol. 12657, pp. 150–163. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-72240-1_11
42. Zhang, X., Yates, A., Lin, J.J.: Comparing score aggregation approaches for document retrieval with pretrained transformers. In: ECIR (2021)