Sentence-level representations for document ranking

Manuel Santos
manuel.s.santos@tecnico.ulisboa.pt
Instituto Superior Técnico, University of Lisbon, Portugal

Abstract

Pre-trained contextual language models based on Transformers have been successful in a number of applications in natural language processing, and more recently also on information retrieval problems. In this paper, we propose the use of sentence-level representations, built through this type of models, for ad-hoc document ranking problems. We predict relevance scores for long documents by aggregating sentence-level scores from a pool of candidate sentences, determined by a RoBERTa-based model. Experiments on the TREC GOV collection show that the proposed approach produces better results than using simpler well known ranking function based on sparse representations, like BM25.

Keywords: Information Retrieval, Ad-Hoc Document Ranking, Pre-Trained Language Models

1. Introduction

The use of neural networks in Information Retrieval (IR), and particularly for document ranking, has been expanding in recent years [Lin et al., 2020]. Pre-trained language models (PLMs), like BERT [Devlin et al., 2019] and RoBERTa [Liu et al., 2019], are achieving state-of-the-art results on retrieval benchmarks and in a number of related natural language processing (NLP) tasks, such as question answering and text summarization. These models are being particularly successful because, unlike traditional word embedding models like word2vec [Mikolov et al., 2013] or unidirectional language models like ELMo [Peters et al., 2018], BERT creates deep bidirectional representations. BERT and RoBERTa rely on a Transformer encoder [Vaswani et al., 2017] to generate a fixed sized length output representation which has a quadratic computational complexity to the input sequence, so the input sequence length is usually limited to 512 tokens. Therefore, when applying PLMs to the task of document ranking, these models often fall short to encode the entirety of most document contents, since their size usually surpasses the model limit. To avoid this problem, several previous studies predict relevance scores over sentences or passages, to be then aggregated into a document relevance score [Yilmaz et al., 2019, Dai and Callan, 2019].

An issue with passage-level approaches is that the majority of ad-hoc collections only have relevance judgments for the whole document, making it difficult to fine-tune a passage-based ranking model in the same domain. Given this problem, models based on BERT are mostly fine-tuned on MSMARCO, i.e. a passage ranking dataset [Nguyen et al., 2016], and to our knowledge no one has yet tried to modify relevance judgements in an ad-hoc collection into a sentence-level weak labeled dataset. Given this problem, one of our motivations is to initially explore an unsupervised approach based on a RoBERTa model, previously trained for the task of semantic similarity between sentences. This way, our model can take advantage of datasets with sentence-labeled pairs for document retrieval. Additionally, we consider the findings from Yang et al. [2019], that demonstrated the increased effectiveness of BERT when fined-tuned on the same task, to further fine-tune RoBERTa with a weak signaled dataset.

In this paper, we analyse how a RoBERTa model can be utilized in the task of document ranking, based on a sentence-level approach. We infer a document's relevance score by aggregating RoBERTa's scores of the document's best sentences. These candidate sentences are chosen based on their position and query term similarity. Furthermore, we adapt document-level relevance judgments into a weak supervised sentence-level dataset, in order to create an environment where RoBERTa can be fine-tuned and tested on the same domain and task. We evaluate the efficiency of our proposal on a TREC ad-hoc collection, concluding that our approach has promising results, outperforming the baseline ranking function BM25. In brief, our work has the following contributions:

- The proposal of a document ranking method based on a sentence-level approach, where only the best sentences are processed by a RoBERTa model, trained for sentence similarity, and aggregated into a final document-level relevance score.
- The creation of a weak-labeled dataset, where the document labels are adapted into sentence-level weak signals, in order to analyse how RoBERTa benefits from being fine-tuned on the same domain and task.

- An evaluation of our proposal on a standard ad-hoc TREC collection, showing the effectiveness of our approaches.

2. Related Work

In this section, we review the most relevant research done in connection to sentence-level document ranking.

2.1. Passage-Level Relevance Ranking

In document ranking benchmarks, relevance judgments are almost always associated to the whole document, and hence traditional retrieval models calculate relevance scores based on document-level signals. However, of all the sentences that compose a document, only a select few are perhaps relevant for a given query. Given the increase of document lengths in full-text collections, Callan [1994] first proposed to consider passage relevance for retrieval tasks. He defined passages by splitting a document into three different ways: paragraph passages, bounded-paragraph passages, and window passages. After a document is split into passages, we can obtain passage relevance signals that can be used to calculate a document-level relevance score, e.g. by averaging or taking the maximum score [Liu and Croft, 2002].

More recently, Wu et al. [2019] studied the relation between passage-level relevance and document-level relevance judgments. They showed that position and query similarity of passages play a significant role in the determination of document-level relevance. These authors also demonstrated that on the THUCNews[1] dataset, in average, a relevant document only has 23% of highly relevant passages. In subsequent work, Wu et al. [2020] proposed a model that uses a passage-level representation based on a cumulative gain, where the last passage cumulative gain represents the document-level cumulative gain. Unlike our work, they deal with a dataset with a passage-level ground truth.

In our work, we take into account the aforementioned findings in order to select a pool of candidate sentences to build a document relevance score. With this approach, we differ from most passage-level representation models, as we only aggregate the relevance scores of the most relevant sentences. This reduction of sentences processed drastically decreases the computational costs.

2.2. Neural Ranking Models for IR

There is a large variety of ranking models, including vector space models (e.g., classic TF-IDF), probabilistic models (e.g., BM25 [Robertson et al., 1996]), feature-based learning to rank models (e.g., LambdaMART [Burges, 2010]) and neural ranking models (e.g., DSSM [Huang et al., 2013] or DRMM [Guo et al., 2016]). However, the contextual capacity of the aforementioned models is much more limited than a BERT-based model, pretrained on a large-scale corpus. Recent work has shown that PLMs achieve state-of-the-art results in many NLP tasks, and also in IR problems [Lin et al., 2020]. Nogueira and Cho [2019] first utilized BERT as a passage reranker, using the MSMARCO passage ranking dataset for fine-tuning the model. The authors use BERT's [CLS] vector as input to a single layer neural network, to obtain a final probability score. In subsequent work Nogueira et al. [2019] developed a multi-stage document ranking architecture with BERT. In the first stage, the top-k_0 documents retrieved by a standard ranking function are reranked by a first BERT model. After that, the top-k_1 documents are then reranked by duoBERT, a second BERT model trained through a pair-wise classification approach. This design has the ability to trade off quality against latency by controlling the number of documents that enter each stage. Previous studies have also shown that ensembles of BERT models can be used to improve results on passage re-ranking, e.g. aggregating the scores of several snapshots taken during model training through approaches such as MAPFuse [Borges et al., 2021].

Birch [Yilmaz et al., 2019] is another recent approach which started to utilize sentence-level labels, using BERT to create a document reranker. The authors estimate a document relevance score from the combination of the document's original score (e.g., obtained through a model like BM25) with the aggregation of the top-n most relevant sentences according to BERT. BERT-MaxP [Dai and Callan, 2019] is also a document reranker that instead explores passage-level signals. The authors adopt a simple passage-level approach by splitting the document into overlapping passages. BERT is then used to predict the relevance of each passage independently, and the final score is obtained with the best passage.

CEDR [MacAvaney et al., 2019] corresponds to a joint approach that incorporates BERT's vector representation into existing neural models, such as DRMM. The paper's method is to use BERT's [CLS] vector, benefiting from deep semantic information, as well as individual contextualized token matches. PARADE [Li et al., 2020] is an end-to-end document reranking model that aggregates passage-level representations, overcoming the problem of performing inference over passages independently. The first step of PARADE is to represent a document as passages. To do so, a sliding window of 150 words is applied to the document with a stride of 100 words. In the next step, each passage is represented by BERT's [CLS] token, built from the concatenation between the query and the passage. In the passage aggregation phase, all passage representations are con-

[1] `http://thuctc.thunlp.org`

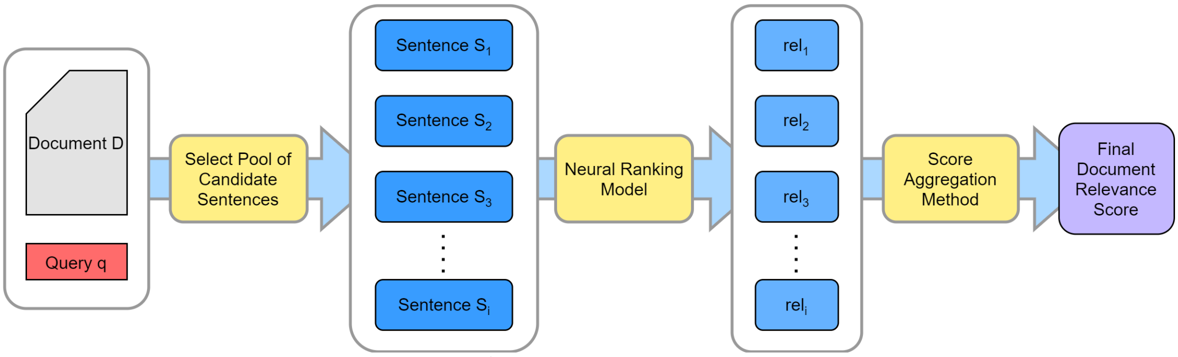

Figure 1: Illustration of our general document ranking architecture.

catenated and the resulting vector is given as input to Transformer [Vaswani et al., 2017] layers, enabling interaction between passages and exploiting the ordering and dependencies between them. Finally, the [CLS] vector of the last Transformer output layer is given as input to a single-layer feed-forward network to generate the final document relevance score. BERT-QE [Zheng et al., 2020] outperforms standard BERT-based models by adding a phase of contextualized query expansion in their three phased approach. In phase one, a BERT model is used to re-rank a list of documents based on an unsupervised ranking model. In phase two, the top-k_d documents from the previous phase are selected to return the most relevant chunks of text, to serve as feedback information. In phase three, the selected chunks are used in combination with the query and the document to compute a final relevance score. For a deeper understanding about the evolution of text ranking, Lin et al. [2020] presented an overview on modern techniques.

3. Methodology

In this section, we present the proposed method for document ranking using a sentence-level approach. For a given query q and a document D, we calculate a relevance score $\mathrm{rel}(q, D)$ that determines the importance of document D for the query q. This relevance is performed by aggregating the top-n best sentence-level scores, obtained by a neural model such as RoBERTa trained on sentence similarity tasks, [Reimers and Gurevych, 2019] into a document-level score. Figure 1 illustrates the general architecture of our proposal.

3.1. Choosing Candidate Sentences

We do not aim to use the neural ranking model to encode every single sentence in a document. Instead, we calculate a document relevance score based on a specific pool of candidate sentences, formally expressed as $D_P = \{S_1, \ldots, S_n\}$, where n is the number of sentences. This approach will lead to a significant reduction of computational costs, since each document can have a very large number of sentences.

We tested three different approaches to choose a pool of candidate sentences, based on two criteria: (i) the position of a sentence in the document; or (ii) the number of shared terms between a query and a sentence.

The approach named **FIRST** picks the first sentences of a document, exploring the fact that the most relevant information of a document tends to be near the beginning.

The approach named **TERMF** contains the sentences that have the highest raw term frequency score, denoted as follows:

$$\mathrm{tf}(q, S) = \sum_{t_i \in q} f_{i,S} \tag{1}$$

In the previous expression, $f_{i,S}$ is the raw count of query term t_i in sentence S. In the experiments, we ignored all terms that were either stop words or punctuation.

Finally, the approach named **FIRST+TERMF** corresponds to an aggregation of both sets. If the same sentence is in both groups, that sentence is not repeated and another is chosen from TERMF.

3.2. Creating Sentence Scores

For query q and sentence S_i, we use a RoBERTa model to generate a fixed sized vector representation for both query and sentence. This output is computed by calculating the mean of all vectors produced for the individual word pieces generated during tokenization, having q^{avg} and s_i^{avg}, denoted as follows:

$$q^{avg} = \mathrm{RoBERTa}(q) \tag{2}$$
$$s_i^{avg} = \mathrm{RoBERTa}(S_i) \tag{3}$$

Note that we do not use the traditional inference method of selecting the output token [CLS], given the concatenation of the two strings as input to a RoBERTa cross-encoder. In our experiments, this setup becomes too expensive because we are dealing with too many possible combination pairs. Since our focus is to efficiently find the most similar sentences given a query, it can be more beneficial to build a model properly trained to find semantic similarity between sentences. We follow the work done by Reimers and Gurevych [2019], which adds a mean-pooling operation to the output of RoBERTa (i.e., computing the mean of all output token vectors), in order to derive a fixed sized sentence embedding. With this approach, the authors designed a bi-encoder that maps each input independently, and then determines matching scores with the cosine similarity between the two vectors. In our experiments, we use as base model their version of RoBERTa fine-tuned on the combination of the SNLI [Bowman et al., 2015] and Multi-Genre NLI [Williams et al., 2018] datasets, and then on the Semantic Textual Search benchmark (STS-b) [Cer et al., 2017], since this model achieved state-of-the-art results for sentence similarity tasks.

The relevance score is then obtained by calculating the cosine similarity between the two vectors.

$$\mathrm{rel}(q, S_i) = \cos(\theta) = \frac{q^{avg} \cdot s_i^{avg}}{\|q^{avg}\| \times \|s_i^{avg}\|} \tag{4}$$

In the previous equation, $q^{avg} \cdot s_i^{avg}$ corresponds to the dot product between the vectors and $\| * \|$ is the vector norm.

3.3. Aggregating Sentence Relevance Scores

Given the pool of sentence relevance scores $D_{P_{rel}} = \{rel_1, \ldots, rel_n\}$, we can obtain a document relevance score in three different ways.

Max calculates a document relevance score by choosing the sentence with the highest score.

$$\mathrm{rel}(q, D) = \max(rel_1, \ldots, rel_n) \tag{5}$$

Sum assumes that all candidate sentences must contribute equally in scoring a document, thus summing all relevance scores.

$$\mathrm{rel}(q, D) = \sum_{i=1}^{n} rel_i \tag{6}$$

Weighted Mean considers that sentences with a higher query term frequency must have a higher weight on a document relevance score.

$$\mathrm{rel}(q, D) = \frac{\sum_{i=1}^{n} w_i \times rel_i}{\sum_{i=1}^{n} w_i} \tag{7}$$

In the previous equation, w_i is the raw count of query q terms in the correspondent sentence S_i.

3.4. Combining Ranking Systems

Similarly to the work done by Yilmaz et al. [2019], we decided to combine the scores of two ranking systems (i.e., the initial ranking function and RoBERTa), in order to take advantage of both approaches. To do so, we used the fusion algorithm named MAPFuse [Lillis et al., 2010, Borges et al., 2021] to create a new ranking system, by combining the document scores given by the baseline ranking function and RoBERTa, with the help of their correspondent MAP scores over a held-out set of queries. The MAPFuse formula is denoted as follows; where S is the set of input systems that returned document D, MAP_s is the MAP score associated with system s, and $p_s(D)$ is the position of document D ranked by system s.

$$\mathrm{rel}(q, D) = \sum_{s \in S} \frac{MAP_s}{p_s(D)} \tag{8}$$

4. Experiments

In this section, we explain the experiments that were made to in order to test our methodology.

4.1. Dataset

We analysed our method with the ad-hoc retrieval collection named GOV[2]. This is a TREC Web collection crawled from government websites, with approximately 1.25 million documents. We used the TREC Web Topic Distillation topics from the years 2002, 2003, and 2004 for our unsupervised approaches. In our supervised experiment, we used the TREC 2002 Web Topic Distillation topics as test data, and both TREC 2003 and 2004 Web Track topics as training data. Since we have some queries with only a title and others with title and description, we have chosen to uniformly use the title only for all queries, having a total of 775 queries. In average, each document has a much higher number of tokens than RoBERTa can handle, making GOV a reliable collection to test our hypothesis.

4.2. Experimental Setup

To store and index our collection of documents we used Apache Solr[3], which is a well known text search platform. Considering that the majority of the GOV documents are in the HTML format, we created a parser to eliminate all document's unwanted content, like tags and Javascript code.

To split a document into sentences we used an English parser from the Spacy[4] library. When choosing the candidate sentences, we set to 10 the total number of sentences for the approaches named FIRST and TERMF, and 20 for the approach named FIRST+TERMF. These values were tuned based on a trade-off between sentence pool size and performance. In Section 5, we further investigate the variation of performance, given the different number of sentences processed by a RoBERTa-based approach.

4.3. Baselines

We compare our RoBERTa models against two traditional baselines, both implemented within Solr.

BM25 is an unsupervised ranking function that scores a document based on the term frequency and the inverse document frequency, considering the document length as a normalization factor [Robertson et al., 1996]. We set BM25 parameters as default, with $k_1 = 1.2$ and $b = 0.75$.

[2] http://ir.dcs.gla.ac.uk/test_collections/govinfo.html

[3] https://lucene.apache.org/solr

[4] https://spacy.io

	2002 Topics		2003 Topics		2004 Topics	
Model	MAP@1K	P@10	MAP@1K	P@10	MAP@1K	P@10
BM25	0.1617	0.1980	0.0892	0.0680	0.2321	0.0693
BM25+Porter	0.1915	0.2460	0.0858	0.0720	0.2478	0.0707
BM25+Porter [Bennett et al., 2008]	0.1888	0.2420	-	-	-	-

Table 1: Results of the different baselines, considering the TREC Web Topics Distillation topics from 2002, 2003, and 2004.

BM25+Porter combines BM25 with a stemming algorithm that reduces inflected or derived words to their root form [Porter, 1980]. This baseline also removes stop words with a Solr predefined list.

To check the performance of our baselines, we validate them against the method implemented by Bennett et al. [2008]. These authors reported to have used BM25 tuned with the same parameter values, also having the text pre-processed with Porter's algorithm and a stop words list. As shown in Table 1, for all the topic's years considered, we can see a substantial improvement from pre-processing the documents with a stemming algorithm and a list of stop words. There is also a slight improvement from our implementation of BM25+Porter compared with the one made by Bennett et al. [2008] for the year 2002, which validates our reranking baseline. Given these results, we decided to use the top 1000 documents retrieved by the method BM25+Porter in our reranking methods, for having the best performance.

4.4. Model Training

As mentioned previously in Section 3, we use a publicly available RoBERTa-Base[5] model, already fine-tuned for sentence similarity. In order to further fine-tune RoBERTa for the GOV dataset, we need to adapt the relevance judgments from documents to sentences. To do so, for each document, we choose the most relevant sentence from the pool of candidate sentences given by FIRST+TERMF and use that sentence as an instance. With this approach, we assume that all instances taken from relevant documents are relevant (i.e., similar to the query title) and all instances taken from non-relevant documents are non-relevant.

Training is performed on a single GPU GeForce GTX 1080, using a triplet loss where the anchor input s_a is compared to a positive input s_p and a negative input s_n (i.e., a query is compared to a relevant and a non-relevant sentence) denoted as:

$$\mathcal{L} = \sum_{i \in b} \max(\|s_{a_i} - s_{p_i}\| - \|s_{a_i} - s_{n_i}\| + \epsilon, 0) \quad (9)$$

In the previous equation, b is the batch of training instances, $\| \cdot \|$ is the Euclidean distance metric, and ϵ is a margin. Thus, the model is tuned so that the distance between the query and a relevant sentence is lower than the distance between the query and a non-relevant sentence.

The training data was constructed by pairing a relevant sentence with a non-relevant one from a random document that is picked from the top-50 non-relevant documents retrieved by BM25. We also use data augmentation by repeating each relevant document a total of five times, pairing it with different negative sentences. We fine-tune the model for 2 epochs with batches of 8 training instances, with a 10% random split between training and development data, having a total of 25200 training instances. We use the Adam optimizer with a learning rate of 3e-5 and with 10% of training data for warm-up.

4.5. Evaluation

The evaluation is made with the Mean Average Precision (MAP), the Normalized Discounted Cumulative Gain (nDCG), P@10 and nDCG@10 metrics. The MAP formula is denoted as follows:

$$\mathrm{MAP}(Q) = \frac{\sum_{q=1}^{Q} \mathrm{AP}(q)}{|Q|} \quad (10)$$

In the previous equation, $|Q|$ is the total number of queries and $\mathrm{AP}(q)$ is the average precision for query q, which is calculated as follows:

$$\mathrm{AP}(q) = \frac{\sum_{k=1}^{n} \mathrm{P}(k) \times rel(k)}{\#\mathrm{RelevantDocuments}} \quad (11)$$

In the previous equation, $\mathrm{P}(k)$ corresponds to the precision at cutoff k documents and $rel(k)$ is 1 or 0 depending if the document is relevant or non-relevant, respectively.

In turn, the nDCG formula is denoted as follows:

$$\mathrm{nDCG}_p = \frac{\mathrm{DCG}_p}{\mathrm{IDCG}_p} \quad (12)$$

In the previous equation, p is a rank position and IDCG_p is the value of DCG_p sorted by relevance. DCG_p can be obtained by the following formula:

$$\mathrm{DCG}_p = \sum_{i=1}^{p} \frac{rel(i)}{\log_2(i+1)} \quad (13)$$

The reranking threshold was set to 30 for optimal performance. In Section 5, we validate this choice by studying how the variation of the number of documents that are reranked affects the overall performance of our method.

[5] https://github.com/UKPLab/sentence-transformers

	Single System				MAPFuse			
Model	MAP@1K	nDCG@1K	P@10	nDCG@10	MAP@1K	nDCG@1K	P@10	nDCG@10
BM25	0.1617	0.4129	0.1980	0.2440	-	-	-	-
BM25+Porter	0.1915	0.4648	0.2460	0.3049	-	-	-	-
BM25 [Bennett et al., 2008]	0.1888	-	0.2420	-	-	-	-	-
RoBERTa (Full Text)	0.1533	0.4322	0.2400	0.2713	**0.1911**	0.4646	0.2480	0.3074
1. RoBERTa$_{\text{Max}}$	0.1512	0.4286	0.2400	0.2616	0.1887	0.4624	0.2620	0.3124
1. RoBERTa$_{\text{Sum}}$	0.1556	0.4354	0.2540	0.2791	0.1857	0.4603	0.2600	0.3094
1. RoBERTa$_{\text{W.Mean}}$	0.1365	0.4182	0.2120	0.2397	0.1827	0.4559	0.2320	0.2884
2. RoBERTa$_{\text{Max}}$	0.1488	0.4333	0.2100	0.2515	0.1838	0.4613	0.2540	0.3062
2. RoBERTa$_{\text{Sum}}$	0.1592	0.4337	0.2120	0.2504	0.1815	0.4580	0.2340	0.2888
2. RoBERTa$_{\text{W.Mean}}$	0.1417	0.4181	0.2020	0.2256	0.1815	0.4574	0.2340	0.2867
3. RoBERTa$_{\text{Max}}$	0.1527	0.4299	0.2360	0.2578	0.1891	0.4632	0.2620	0.3120
3. RoBERTa$_{\text{Sum}}$	0.1655	0.4418	0.2200	0.2664	0.1898	0.4657	0.2500	0.3117
3. RoBERTa$_{\text{W.Mean}}$	0.1504	0.4285	0.2060	0.2383	0.1840	0.4584	0.2300	0.2872
3. RoBERTa$_{\text{Max}}$ (fine-tuned)	0.1516	0.4394	0.2240	0.2732	0.1884	**0.4663**	**0.2660**	**0.3234**

Table 2: Results of different models on the GOV dataset, considering the TREC 2002 Web Topic Distillation topics. The rows labeled with 1. 2. or 3. correspond to the FIRST, TERMF, and FIRST+TERMF approaches, respectively. Best results are in **bold**.

4.6. Results

The ranking performance of our methods is shown in Tables 2, 3, and 4, corresponding to the 2002, 2003, and 2004 topic distillation years, respectively. We can see that when RoBERTa uses the document's full content, cut to the maximum number of word pieces that are allowed, it underperforms over some sentence-level versions for all the different topic distillation years. This confirms that for long-sized documents, RoBERTa benefits from a sentence-level representation approach. We can also conclude that almost all the MAPFuse results, given by the fusion between a RoBERTa reranking system and the BM25+Porter baseline ranking system, perform better than it's systems evaluated separately. In Table 3, we can observe that the best approach of the 2003 set, 3. RoBERTa$_{\text{Max}}$, already surpasses the results given by BM25+Porter, suggesting that, in some cases, our RoBERTa approach can outperform the baseline ranking function without the need of a fusion algorithm.

When analysing the different methods for choosing the candidate sentences, FIRST+TERMF yields the best results for the 2002 and 2003 sets, while FIRST is the most efficient approach for the 2004 set. This shows that the first sentences are essential to consider in a sentence-level approach, confirming that relevant information tends to be near the top of a document. On the other hand, the approach TERMF gave the worst results in all three experiments, reinforcing the importance of considering the document's first sentences, even if they have a low number of terms in common with the correspondent query.

As for the score aggregation methods, we can see that the most effective relevance score aggregation method is Sum for the 2002 set and Max for the 2003 and 2004 sets. This suggests that in particular topics it is more advantageous to predict a relevance score based equally on multiple sentences and in others to use the single most relevant sentence. Weighted Mean had always lower results than the other two methods, which is perhaps due to the fact that the first sentences in a document tend to be somewhat equally relevant. Also, BERT based models rely on contextual semantic information to predict it's embeddings, meaning that a high term frequency between query and sentence does not necessarily translate on a high similarity. In terms of the improvements of the nDCG@10 metric, the 2002 set had a maximum improvement of 2.5% over the initial baseline ranker BM25+Porter, while the 2003 set had an increase of 22.4% and the 2004 set improved 11.8%.

The best results regarding the RoBERTa's model fine-tuned on the GOV weak labeled dataset can be seen in the last row of Table 2. We can verify that this model achieved the best value of nDCG@10, improving 6.1% over BM25+Porter, while the other metrics are in line with the previous best RoBERTa method. We only reported the most effective approach, which was the FIRST+TERMF sentence pool together with Max aggregation. This was because, when building the dataset, we chose only the best sentence from the document's pool of sentences given by FIRST+TERMF.

5. Analysis

In this section, we research the impact of the following questions in our work:

- How does the number of sentences that is considered affect the performance of our ranking method?
- Can a different version of RoBERTa improve effectiveness without losing efficiency?
- How does the number of documents that are reranked by RoBERTa affect the performance of our ranking method?

	Single System				MAPFuse			
Model	MAP@1K	nDCG@1K	P@10	nDCG@10	MAP@1K	nDCG@1K	P@10	nDCG@10
BM25	0.0892	0.2897	0.0680	0.1161	-	-	-	-
BM25+Porter	0.0858	0.2935	0.0720	0.1123	-	-	-	-
RoBERTa (Full Text)	0.0889	0.3014	0.0800	0.1194	0.0945	0.3028	0.0880	0.1292
1. RoBERTa$_{\text{Max}}$	0.0998	0.3056	0.0840	0.1316	0.0866	0.2957	0.0800	0.1158
1. RoBERTa$_{\text{Sum}}$	0.0855	0.2968	0.0880	0.1242	0.0924	0.3018	**0.0900**	0.1286
1. RoBERTa$_{\text{W.Mean}}$	0.0854	0.2934	0.0700	0.1105	0.0930	0.2980	0.0820	0.1198
2. RoBERTa$_{\text{Max}}$	0.0708	0.2801	0.0640	0.0896	0.0754	0.2842	0.0840	0.1075
2. RoBERTa$_{\text{Sum}}$	0.0910	0.3008	0.0740	0.1205	0.0818	0.2916	0.0760	0.1124
2. RoBERTa$_{\text{W.Mean}}$	0.0827	0.2931	0.0700	0.1092	0.0846	0.2935	0.0860	0.1201
3. RoBERTa$_{\text{Max}}$	**0.1092**	**0.3115**	**0.0900**	**0.1374**	0.1019	0.3073	0.0840	0.1300
3. RoBERTa$_{\text{Sum}}$	0.0975	0.3021	0.0700	0.1151	0.0880	0.2963	0.0760	0.1142
3. RoBERTa$_{\text{W.Mean}}$	0.0936	0.3056	0.0720	0.1261	0.0918	0.3012	0.0800	0.1247

Table 3: Results of different models on the GOV dataset, considering the TREC 2003 Web Topic Distillation topics. The rows labeled with 1. 2. or 3. correspond to the FIRST, TERMF, and FIRST+TERMF approaches, respectively. Best results are in **bold**.

	Single System				MAPFuse			
Model	MAP@1K	nDCG@1K	P@10	nDCG@10	MAP@1K	nDCG@1K	P@10	nDCG@10
BM25	0.2321	0.3943	0.0693	0.2746	-	-	-	-
BM25+Porter	0.2478	0.4057	0.0707	0.2914	-	-	-	-
RoBERTa (Full Text)	0.2079	0.3741	0.0702	0.2532	0.2498	0.4091	0.0782	0.3035
1. RoBERTa$_{\text{Max}}$	0.2375	0.4011	0.0716	0.2837	**0.2736**	**0.4286**	**0.0800**	**0.3257**
1. RoBERTa$_{\text{Sum}}$	0.2134	0.3785	0.0707	0.2592	0.2662	0.4226	0.0769	0.3162
1. RoBERTa$_{\text{W.Mean}}$	0.2037	0.3705	0.0618	0.2429	0.2640	0.4196	0.0724	0.3059
2. RoBERTa$_{\text{Max}}$	0.1898	0.3637	0.0627	0.2329	0.2591	0.4152	0.0738	0.3026
2. RoBERTa$_{\text{Sum}}$	0.1914	0.3568	0.0551	0.2217	0.2567	0.4123	0.0680	0.2931
2. RoBERTa$_{\text{W.Mean}}$	0.2047	0.3699	0.0591	0.2419	0.2574	0.4129	0.0724	0.2993
3. RoBERTa$_{\text{Max}}$	0.2271	0.3953	0.0764	0.2827	0.2590	0.4173	0.0787	0.3145
3. RoBERTa$_{\text{Sum}}$	0.1765	0.3475	0.0609	0.2179	0.2455	0.4044	0.0724	0.2892
3. RoBERTa$_{\text{W.Mean}}$	0.1669	0.3410	0.0631	0.2131	0.2505	0.4090	0.0724	0.2967

Table 4: Results of different models on the GOV dataset, considering the TREC 2004 Web Topic Distillation topics. The rows labeled with 1. 2. or 3. correspond to the FIRST, TERMF, and FIRST+TERMF approaches, respectively. Best results are in **bold**.

5.1. Number of Considered Sentences

One important hyper-parameter is the number of sentences considered when choosing the document's candidate sentences. In this section, we analyse how the variation of sentences processed by RoBERTa influences the reranking effectiveness of our approaches.

Figure 2 shows the results in terms of nDCG@10, with the number of sentences varying from 8 to 64. It would be expected for our ranking approaches to increase their performance with a larger amount of document data preserved. However, we can see that for RoBERTa$_{\text{Sum}}$ and RoBERTa$_{\text{W.Mean}}$, the more sentences are considered, the less effective the model becomes. On the other hand, for RoBERTa$_{\text{Max}}$ whose score aggregation method only considers the single most relevant sentence, we havestable results with the increase in the number of sentences. These results validate our hypothesis that it is beneficial, not only in terms of computational costs but also performance-wise, to select a pool of the document's most relevant sentences to be processed by RoBERTa.

5.2. Effectiveness and Efficiency of RoBERTa

In this section, we analyse the effectiveness and efficiency of our model compared with RoBERTa-Large, a more robust version of RoBERTa, as well as against a DistilRoBERTa model trained on the MSMARCO dataset. Although approaches based on Transformer models such as RoBERTa have achieved state-of-the-art results, they are computationally expensive. In the task of document ranking, we need to consider that the ranking system will be applied in real time and thus it is necessary to have an efficient architecture.

In Table 5, we have the sizes of the considered models, as well as the corresponding inference time over a document. The inference time, in seconds, includes the encoding of the query and each sentence within a document, plus the computation of the cosine similarity between the representations. RoBERTa-Large takes approximately 119% more computational time than RoBERTa-Base, while the distilled version of RoBERTa-Base is 36% more time efficient.

Table 6 shows the results achieved by the best methods for the three different models. Regarding the results of RoBERTa-Large, we can see

Model	# Layers	# Layer Size	# Parameters	Inference Time (s / doc)
RoBERTa-Large	24	1024	355M	0.10
RoBERTa-Base	12	768	125M	0.05
DistilRoBERTa-Base	6	768	82M	0.03

Table 5: Different versions of RoBERTa, compared in terms of size and computational time. Inference time over a document is estimated considering the use of the first 10 sentences.

	Single System				MAPFuse			
Model	MAP@1K	nDCG@1K	P@10	nDCG@10	MAP@1K	nDCG@1K	P@10	nDCG@10
1. RoBERTa-Large$_{\text{Max}}$	0.1550	0.4327	0.2500	0.2723	0.1871	0.4646	0.2660	0.3197
1. RoBERTa-Base$_{\text{Max}}$	0.1512	0.4286	0.2400	0.2616	0.1800	0.4565	0.2660	0.3130
1. DistilRoBERTa-Base$_{\text{Max}}$	0.1546	0.4368	0.2400	0.2808	0.1784	0.4617	0.2580	0.3187
1. RoBERTa-Large$_{\text{Sum}}$	0.1563	0.4355	0.2520	0.2815	0.1844	0.4718	0.2620	0.3308
1. RoBERTa-Base$_{\text{Sum}}$	0.1556	0.4354	0.2540	0.2791	0.1879	**0.4736**	0.2660	**0.3322**
1. DistilRoBERTa-Base$_{\text{Sum}}$	0.1602	0.4421	0.2320	0.2829	0.1870	0.4619	0.2640	0.3181
3. RoBERTa-Large$_{\text{Max}}$	0.1554	0.4322	0.2600	0.2742	0.1738	0.4647	0.2660	0.3202
3. RoBERTa-Base$_{\text{Max}}$	0.1527	0.4299	0.2360	0.2578	0.1803	0.4610	0.2560	0.3086
3. DistilRoBERTa-Base$_{\text{Max}}$	0.1615	0.4512	0.2420	0.2997	0.1927	0.4710	0.2600	0.3317
3. RoBERTa-Large$_{\text{Sum}}$	0.1582	0.4375	0.2320	0.2704	0.1798	0.4671	0.2580	0.3198
3. RoBERTa-Base$_{\text{Sum}}$	0.1655	0.4418	0.2200	0.2664	**0.1938**	0.4702	0.2620	0.3243
3. DistilRoBERTa-Base$_{\text{Sum}}$	0.1733	0.4474	0.2320	0.2826	0.1924	0.4670	**0.2680**	0.3276

Table 6: Comparison between the results of different RoBERTa versions. The rows labeled with 1. or 3. correspond to the FIRST and FIRST+TERMF approaches, respectively. Best results are in **bold**.

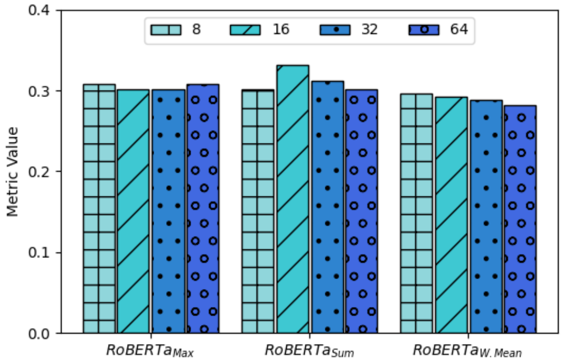

Figure 2: Results using 1. RoBERTa for different numbers of candidate sentences. nDCG@10 is reported.

that with no system fusion there is an increase of performance for all it's methods. This fact suggests that when we only consider the scores of RoBERTa, the model with the biggest size tends to perform better. For the MAPFuse results, RoBERTa-Large has better results than RoBERTa-Base when using the Max aggregation method, but performs worse regarding the Sum aggregation method. Overall, we can conclude that for a bigger RoBERTa version, a loss in efficiency does not necessarily translate in a significant improvement of effectiveness.

As for DistilRoBERTa-Base, we wanted to investigate if a distilled version of RoBERTa-Base, fine-tuned on the MSMARCO dataset could bring performance improvements. With this experiment, we are trying to see if there are significant differences when RoBERTa is trained on a similar ranking task instead of a semantic task. We can see in Table 6 that DistilRoBERTa has promising results given a single system, beating the other two models. However, it does not achieve considerable improvements over the best methods. Overall, DistilRoBERTa gives a sense that it can be a possible option to explore the model's training phase on a passage-level dataset, given the advantage of decreasing the computational costs.

5.3. Number of Reranked Documents

The majority of neural ranking methods apply their model to the top-n documents retrieved by an initial ranking function. In our case, RoBERTa reranks the top documents retrieved by the baseline named BM25+Porter, which is described in Section 4. In this section, we investigate the impact in performance of varying the number of documents that are reranked by our method.

Figure 3 shows the performance of RoBERTa$_{\text{Sum}}$ with the FIRST method, with the number of documents reranked by RoBERTa varying from 10 to 100. We can see that nDCG@10 has its highest value at 20 reranked documents, and then slowly decreases with the increase of documents. P@10 reaches its maximum value at 20, 30, and 50 documents and then falls when considering 100 documents. We can conclude that from 50 reranked documents the model is not capable of increasing its performance. Considering these values and the trade-off between effectiveness and computational cost, we decided to fix the reranking threshold to 30 documents in our experiments.

6. Conclusions

We proposed a sentence-level approach based on RoBERTa for the task of document ranking,

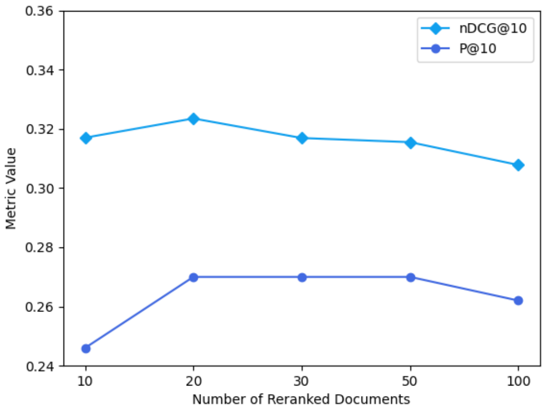

Figure 3: Results using 1. $\text{RoBERTa}_{\text{Sum}}$ for different numbers of reranked documents. nDCG@10 and P@10 are reported.

analysing its performance on the TREC ad-hoc collection named GOV. First, we pick a pool of candidate sentences to be processed by RoBERTa so as to generate relevance scores, and then aggregate the scores a final document-level relevance score. We studied different ways of choosing the best candidate sentences, as well as different aggregation methods. Additionally, we investigated the importance of creating an environment where the model is fine-tuned on the same domain and task, by converting the document-level labels into weak sentence-based signals. Experimental results show that our approach beats the baseline ranking function BM25 and has better results than using a document-level model architecture.

For future work, we believe it would be interesting to test this method with different test collections, that have already been used with recent state-of-the-art models, so that we have a more clear comparison between methods. Since our method is fully based on a sentence-level approach, from the training phase to the inference phase, we can consider experimenting with different Tranformer-based methods for producing sentence representations [Yang et al., 2020], or we can consider making comparisons against similar methods based on processing larger text passages. Also, we only tested a select few relevance score aggregation methods. More advanced functions can be implemented to further improve the results, including the use of rank aggregation methods such as MAPFuse to combine the scores from the candidate sentences [Borges et al., 2021]. As for the model training phase, more elaborate strategies to choose the representative sentences for each document is also a promising path for future work.

References

Jimmy Lin, Rodrigo Nogueira, and A. Yates. Pretrained transformers for text ranking: BERT and beyond. *arXiv:2010.06467*, 2020.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv:1810.04805v2*, 2019.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv:1907.11692*, 2019.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv:1301.3781*, 2013.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *arXiv:1802.05365*, 2018.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv:1706.03762v5*, 2017.

Zeynep Akkalyoncu Yilmaz, Shengjin Wang, Wei Yang, Haotian Zhang, and Jimmy Lin. Applying BERT to document retrieval with birch. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 19–24. Association for Computational Linguistics, 2019.

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

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. MS MARCO: A human generated machine reading comprehension dataset. *arXiv:1611.09268*, 2016.

Wei Yang, Haotian Zhang, and Jimmy Lin. Simple applications of BERT for ad hoc document retrieval. *arXiv:1903.10972*, 2019.

James P. Callan. Passage-level evidence in document retrieval. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 302–310. Springer-Verlag, 1994.

Xiaoyong Liu and W. Bruce Croft. Passage retrieval based on language models. In *Proceedings of the ACM International Conference on Information and Knowledge Management*, pages 375–382. Association for Computing Machinery, 2002.

Zhijing Wu, Jiaxin Mao, Yiqun Liu, Min Zhang, and Shaoping Ma. Investigating passage-level relevance and its role in document-level relevance judgment. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 605–614. Association for Computing Machinery, 2019.

Zhijing Wu, Jiaxin Mao, Yiqun Liu, Jingtao Zhan, Yukun Zheng, Min Zhang, and Shaoping Ma. Leveraging passage-level cumulative gain for document ranking. In *Proceedings of The Web Conference*, pages 2421–2431. Association for Computing Machinery, 2020.

S. Robertson, S. Walker, M. Hancock-Beaulieu, Mike Gatford, and A. Payne. Okapi at TREC-4. In *The Text REtrieval Conference*, pages 73–96. Gaithersburg, MD: NIST, 1996.

Christopher J. C. Burges. From RankNet to LambdaRank to LambdaMART: An Overview. *Machine Learning*, 11(23-581), 2010.

Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the ACM International Conference on Information and Knowledge Management*, pages 2333–2338. Association for Computing Machinery, 2013.

Jiafeng Guo, Yixing Fan, Qingyao Ai, and W Bruce Croft. A deep relevance matching model for adhoc retrieval. In *Proceedings of the ACM International Conference on Information and Knowledge Management*, pages 55–64. Association for Computing Machinery, 2016.

Rodrigo Nogueira and Kyunghyun Cho. Passage re-ranking with BERT. *arXiv:1901.04085*, 2019.

Rodrigo Nogueira, Wei Yang, Kyunghyun Cho, and Jimmy Lin. Multi-stage document ranking with BERT. *arXiv:1910.14424*, 2019.

Luís Borges, Bruno Martins, and Jamie Callan. Assessing the benefits of model ensembles in neural re-ranking for passage retrieval. In *Proceedings of the European Conference on Information Retrieval*, 2021.

Sean MacAvaney, Andrew Yates, Arman Cohan, and Nazli Goharian. CEDR: Contextualized embeddings for document ranking. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1101–1104. Association for Computing Machinery, 2019.

Canjia Li, Andrew Yates, Sean MacAvaney, Ben He, and Yingfei Sun. PARADE: Passage representation aggregation for document reranking. *arXiv:2008.09093*, 2020.

Zhi Zheng, Kai Hui, Ben He, Xianpei Han, Le Sun, and Andrew Yates. BERT-QE: Contextualized query expansion for document re-ranking. In *Findings of the Association for Computational Linguistics*, pages 4718–4728. Association for Computational Linguistics, 2020.

Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 3982–3992. Association for Computational Linguistics, 2019.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 632–642. Association for Computational Linguistics, 2015.

Adina Williams, Nikita Nangia, and Samuel Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*, pages 1112–1122. Association for Computational Linguistics, 2018.

Daniel M. Cer, Mona T. Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. SemEval-2017 task 1: Semantic textual similarity - Multilingual and cross-lingual focused evaluation. *arXiv:1708.00055*, 2017.

David Lillis, Lusheng Zhang, Fergus Toolan, Rem W. Collier, David Leonard, and John Dunnion. Estimating probabilities for effective data fusion. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 347–354. Association for Computing Machinery, 2010.

Graham Bennett, Falk Scholer, and Alexandra Uitdenbogerd. A comparative study of probabilistic and language models for information retrieval. In *Australasian Database Conference*, pages 65–74. Australian Computer Society, 2008.

M. F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980. doi: 10.1108/eb046814. URL http://www.emeraldinsight.com/doi/abs/10.1108/eb046814.

Ziyi Yang, Yinfei Yang, Daniel Cer, Jax Law, and Eric Darve. Universal sentence representation learning with conditional masked language model. *arXiv:2012.14388*, 2020.