Ad-hoc Document Re-ranking by Query Attention and Document-level Context

Sanchit Nevgi
snevgi@umass.edu
University of Massachusetts, Amherst

ABSTRACT

Transformer-based models have recently been applied to the ad-hoc document ranking task, where they have outperformed other neural models such as KNRM [17], PACRR [18] and other traditional IR methods. However, Transformer-based models have a severe limitation that prevent them from being applied as-is to full documents, resulting in using paragraph-level workarounds. Using paragraphs scores may not perform well for documents which have information spread across paragraphs. The recently proposed Longformer model [1] addresses the input size limitation. It handles long documents by using a combination of a local sliding window attention and an end-task motivated global attention mechanism. In this work, we apply the Longformer architecture in the document re-ranking task. Our hypothesis is that the local attention effectively captures the document representation, while the global attention on the query tokens aids in matching. Our code is made publicly available here [1]

1 INTRODUCTION

Ad-hoc ranking over a large data regime is a fundamental task in Information Retrieval. It reflects common real-world web search scenarios. Formally, in ad-hoc retrieval, given a Query Q represented by terms $\{q_1, q_2, ...q_{|Q|}\}$ and a document D given by $\{d_1, d_2, ...d_{|D|}\}$, the task is to compute $score(Q, D) = sim(Q, D)$ where sim represents the similarity between the query and the document, which is then used to rank the documents. In most applications of IR, the collection may contain millions of documents and so the system must be reasonably efficient enough to handle such cases.

The TREC Deep Learning Track challenge was designed to test this very purpose. The challenge consists of two tasks — (1) Document ranking and (2) Passage ranking task. In each task, participants can train models for full-ranking of the documents or use the provided top-1000 documents for re-ranking.

Nowadays, most practical IR systems have adopted a pipelined approach, where an initial set of candidates documents is obtained by statistical approaches such as BM25. The candidate set is then re-ranked to obtain the final rankings. To re-rank the candidate set, recent approaches have used neural methods such as Transformer models. However due to memory limitations of Transformer models, present techniques use paragraph-level workarounds to generate the document relevancy score.

However, the efficacy of these approaches poses several questions. The information that satisfies the user need could potentially be split across many paragraphs and the model would need document-level training signals to effectively capture relevancy.

In this project, we seek to incorporate document-level context in training neural models to obtain a better document ranking. To achieve this, we explore the approaches briefly listed below

(1) A recently proposed model known as Longformer [1] makes use of global and local attention to obtain a representation of long documents. It has outperformed previous neural models in question answering domain. However, to the best of our knowledge it's performance in IR application is as yet untested. In this work, we propose using this model to obtain a representation of the document and get a matching score with the query.
(2) Transformer models have pushed the state-of-the-art in abstractive summarization tasks. In this approach, we train a model to generate document summary ("conditioned" on the query), with the intuition that this summary would closely reflect the user query. Next, we can conventional matching techniques to obtain a score for the re-phrased document query pair.

The paper is organized as follows; §2 describes the recent applications of neural networks, particularly Transformers, to large-scale document retrieval and ranking. §3 describes the LongFormer architecture and how it applies to the document ranking problem. §4 and §5 describes the TREC DLT dataset and our baselines respectively. §7 outlines our method and experiments and §8 describes the results.

2 RELATED WORK

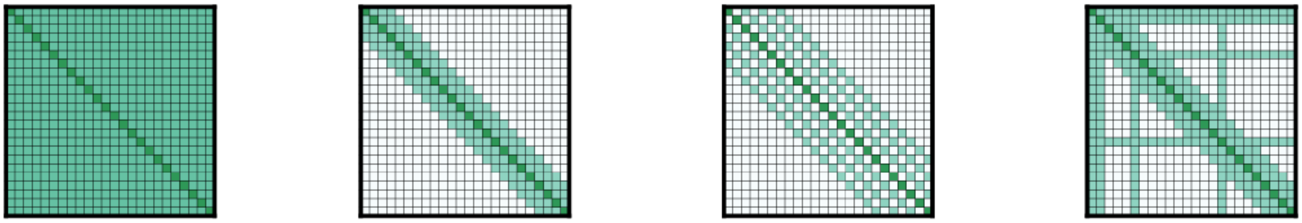
Neural models. With advancements in Deep Learning, neural models such as DRMM [6], (Co)-PACRR [7, 8], (Conv)-KNRM [16, 17] have shown improved performance over traditional IR approaches in the document ranking domain.

Transformer architectures. There is an increasing trend towards using Transformer architectures [15] such as BERT [5] to re-rank documents from a candidate set. MacAvaney et al. incorporated the contextualized word embeddings produced by BERT in existing neural IR architectures like the aforementioned PACRR and KNRM. Nogueira and Cho, Yang et al. proposes a method to obtain document relevance score by aggregating over individual passage scores of the document. The passage score is obtained by computing the similarity between query and passage representation. The aggregated score is usually computed by taking a max or average over the paragraph scores.

Using Document context. There has been limited work on using the entire document context in ranking documents, due to the memory constraints of Transformer models.The time complexity scales quadratically in proportion to the sequence length leading to using paragraph-level workarounds. This is a severe limitation as the information could be spread across paragraphs.

[1] https://github.com/sanchitnevgi/adhoc-ranking

Figure 1: The attention patterns of the LongFormer model. In our experiments, we globally attend to the query tokens. Image credit: "Longformer: The Long-Document Transformer"



(a) Full n^2 attention (b) Sliding window attention (c) Dilated sliding window (d) Global+sliding window

Li et al. introduce an end-to-end Transformer model that incorporates document-level context by evaluating BERT over a sliding window of words and concatenating the paragraph-level [CLS] tokens to obtain a document representation. An additional (smaller) Transformer model is applied to this representation, which supposedly incorporates the ordering and dependencies between passages.

Recently, Beltagy et al. have proposed the Longformer architectures that uses a local windowed attention combined with a task motivated global attention. This architecture has shown promise in long document classification and question answering tasks. Figure 1 shows the the attention pattern of the Longformer model. Note that the computation scales linearly with the input size instead of quadratically. In our approach, we globally attend to the query tokens.

Table 1: Document length statistics from a representative sample ($n = 1000$) of the dataset

Statistic	Query Length	Document Length
mean	31.09	8520.32
std	11.62	14472.29
min	10.00	8.00
25%	23.00	2114.25
50%	30.00	3963.50
75%	37.00	8204.50
max	148.00	125199.00

3 LONGFORMER ARCHITECTURE

Sliding Window: [1]. Kovaleva et al. show that local context contributes to the Transformers success. With this in mind, the LongFormer model applies a sliding window over the input. Formally, in a fixed window size w, the attention mechanism attends to $\frac{1}{2}w$ tokens on either side. Further, multiple stacked layers of this sliding window is applied to obtain the input representation. Figure 1, shows the various attention mechanism. In our case, the sliding window obtains a document representation. Further work is needed to justify if the obtained document representation is better compared to prior work.

Global Attention: For many tasks, the local context isn't sufficient to obtain a good model. For example, question answering tasks require the model to pay attention to the question to highlight the relevant spans in the provided context. The Longformer adds a symmetric global attention mechanism. Traditional attention mechanism [15] are computed as;

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{1}$$

The Longformer uses 2 sets of projection for local and global attention. Q_i, K_i, V_i for $i \in \{l, g\}$

4 DATASET

We use the data provided for the TREC DLT 2019 challenge. It comprises of a large human-generated set of training labels from the MS-MARCO dataset [2] with additional relevance judgement from NIST assessors.

The documents were judged on a four-point scale of Not Relevant (0), Relevant (1), Highly Relevant (2) and Perfect (3). The For binary relevance judgements, the levels 1–3 are considered to be *relevant* and level 0 is deemed *not relevant.*

Table 2 lists the TREC DLT dataset statistics. Note that in this work, we solely focus on the document retrieval dataset. Table 1 lists the document and query length statistics. This helps us select the input sequence length for our model as described in §7.

5 BASELINE

For our baselines, we implemented a fine-tuned BM25 model using the Anserini toolkit [19]. Anserini is an open-source toolkit built on Lucence [18] with the goal to enable reproducible baselines for academic as well as real-world Information Retrieval applications [20].

Using Anserini, we index the TREC 2019 corpus with the default settings, which uses the Porter stemmer. Next, we perform retrieval on the index using the fine-tuned BM25 model which is configured with $k1 = 3.44$ and $b = 0.87$. Additionally, we also use the RM3 re-ranker with 10 feedback documents and queries. The original query is given a weight of 0.5.

The BM25 model with Relevance Feedback (RM3) outperforms the fine-tuned BM25 in MAP and nDCG metrics. Since these metrics

Table 2: Summary of statistics of TREC 2019 DLT challenge. The statistics are reproduced here verbatim [4]

	Document Retreival Dataset		**Passage Retreival Dataset**	
File Description	Number of records	File size	Number of records	File size
Collection	3,213,835	22 GB	8,841,823	2.9 GB
Train queries	367,013	15 MB	502,940	19.7 MB
Train qrels	384,597	7.6 MB	532,761	10.1 MB
Validation queries	5,193	216 KB	12,665	545 KB
Validation qrels	519,300	27 MB	59,273	1.1 MB
Test queries	200	12 KB	200	12 KB

Table 3: Summary of baseline results on the TREC DLT 2019 corpus using a fine-tuned BM25 model (with and without relevance feedback).

	MAP	NDCG@10	R@100	RR
BM25 (Fine-tuned)	0.3138	0.5140	0.3862	0.8872
+ RM3	0.3697	0.5485	0.4193	0.8074
BERT$_{passage}$ [3]	-	**0.6437**	-	0.8992
Longformer Base (ours)	**0.4353**	0.6418		**0.9011**

are the most relevant to us, we compare our approach to the this model going ahead.

The summary of our baseline results are provided in Table 3.

6 EVALUATION

We evaluate our model using the Mean Average Precision (MAP) and Normalized Discounted Cumulative Gain (nDCG) and Mean Reciprocal rank (RR) metrics. This is in line with the metrics used for the TREC DLT 2019 challenge [4]. These metrics were chosen to reflect the performance of real-world search engines, where a majority of queries return multiple relevant documents. Our model is evaluated on the official query test set, which consists of 200 queries and a ranked list of the top 100 documents and their relevancy scores.

7 EXPERIMENTAL SETUP

In modern applications of document information retrieval, the corpus size is very large (in the order of billions). To combat this, practitioners use a pipelined approach. An initial set of 100 documents is retrieved with traditional IR methods such as BM25. Subsequently, we use a neural model to re-rank the candidate set to get final rankings. The TREC dataset provides a candidate set, retrieved with a fine-tuned BM25 similar to the approach describes in §5.

As mentioned previously, we use the Anserini toolkit for our baselines. For our main approach, we make use of the `PyTorch`'s [2]auto-differentiation mechanics and the `transformers` [3] library to build and train our model. We evaluate our results using the `trec_eval` tool provided by Anserini.

Dataset Pre-processing: We generate a 10,000 training and 1,000 dev triples from the provided training and dev dataset, using the script [4] from the TREC DLT repository. Each triple consists of (q, d_{pos}, d_{neg}), where q is the query, d_{pos} is a positively judged document and d_{neg} is any other document, randomly picked from the initial candidate set. Using this we create a positive and a negative query-document pair resulting in 20,000 training examples.

In our approach, we use a `longformer-base` model with a linear layer on top of the classification token (`<s>`) for binary classification (relevant or non-relevant). The model was pre-trained on long documents using the MLM objective [5]. The input is a concatenated query-document pair, separated by a delimiter token and tokenized using the byte-level Byte-Pair-Encoding [12] as shown below.

```
<s> query </s> document </s>
```

In addition, we build a global attention mask that attends only to the query tokens.

Model configuration: We use a $seq\_len \in \{4000, 10000\}$ as it effectively captures 50% and 80% of the corpus without truncation as seen from Table 1. As expected, a longer sequence length performs better due to the longer context. Additionally, we use the GeLU non-linearity and 12 hidden layers and attention heads.

Training: We use the Adam optimizer [9] with a learning rate of $3e-5$. The pre-trained Longformer model is fine-tuned on an NVIDIA 2080-ti GPU for a maximum of 10000 steps. We also experiment with a learning rate schedule with a linear warmup for 200 steps. However, we found that it leads to a slower convergence and doesn't improve performance significantly. Due to the large input size, we use a batch size of 2 and accumulate gradients for 4 batches, leading to an effective batch size of 8. We also train on half precision to speed up training. The results of our experiments are shown in Table 5.

[2] https://pytorch.org/

[3] https://github.com/huggingface/transformers

[4] `msmarco-doctriples.py`

8 RESULTS & ANALYSIS

In this work, we seek to improve the document re-ranking in a large data regime as put forth by the TREC DLT 2019 challenge. We do so by leveraging the LongFormer's local sliding window attention mechanism while globally attending to the query tokens to obtain a document relevancy score. We find that by globally attending to the query tokens the model is able to create a better representation of the document without having to use paragraph-level workarounds. Our approach using the Longformer-base model outperforms the baseline fine-tuned BM25 methods as measured by the nDCG, AP metrics and is comparable to the recent BERT models that have specialized architectures to utilize passage-level representations.

TEAM MEMBERS AND OVERLAP STATEMENT

This is an **individual** project, inspired from the **default** "Ad-hoc ranking" project specification. It *does not overlap* with any of the author's current or previous work.

REFERENCES

[1] Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The Long-Document Transformer. *ArXiv* abs/2004.05150 (2020).

[2] Daniel Fernando Campos, T. Nguyen, M. Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, L. Deng, and Bhaskar Mitra. 2016. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset. *ArXiv* abs/1611.09268 (2016).

[3] Xuanang Chen, Canjia Li, B. He, and Yingfei Sun. 2019. UCAS at TREC-2019 Deep Learning Track. In *TREC*.

[4] Nick Craswell, Bhaskar Mitra, E. Yilmaz, Daniel Fernando Campos, and E. Voorhees. 2020. Overview of the TREC 2019 deep learning track. *ArXiv* abs/2003.07820 (2020).

[5] J. Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT*.

[6] J. Guo, Y. Fan, Qingyao Ai, and W. Croft. 2016. A Deep Relevance Matching Model for Ad-hoc Retrieval. *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management* (2016).

[7] Kai Hui, Andrew Yates, K. Berberich, and G. Melo. 2017. PACRR: A Position-Aware Neural IR Model for Relevance Matching. In *EMNLP*.

[8] Kai Hui, Andrew Yates, K. Berberich, and G. Melo. 2018. Co-PACRR: A Context-Aware Neural IR Model for Ad-hoc Retrieval. *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining* (2018).

[9] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. *CoRR* abs/1412.6980 (2015).

[10] O. Kovaleva, Alexey Romanov, Anna Rogers, and Anna Rumshisky. 2019. Revealing the Dark Secrets of BERT. In *EMNLP/IJCNLP*.

[11] Canjia Li, A. Yates, S. MacAvaney, B. He, and Y. Sun. 2020. PARADE: Passage Representation Aggregation for Document Reranking. *ArXiv* abs/2008.09093 (2020).

[12] Y. Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, M. Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *ArXiv* abs/1907.11692 (2019).

[13] S. MacAvaney, Andrew Yates, Arman Cohan, and N. Goharian. 2019. CEDR: Contextualized Embeddings for Document Ranking. *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval* (2019).

[14] Rodrigo Nogueira and Kyunghyun Cho. 2020. Passage Re-ranking with BERT. *arXiv:1901.04085 [cs]* (April 2020). http://arxiv.org/abs/1901.04085 arXiv: 1901.04085.

[15] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, A. Gomez, L. Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. *ArXiv* abs/1706.03762 (2017).

[16] Chenyan Xiong, J. Callan, and Zhiyuan Liu. 2017. Convolutional Neural Networks for So-Matching N-Grams in Ad-hoc Search Zhuyun Dai.

[17] Chenyan Xiong, Zhuyun Dai, J. Callan, Zhiyuan Liu, and R. Power. 2017. End-to-End Neural Ad-hoc Ranking with Kernel Pooling. *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval* (2017).

[18] Peilin Yang, Hui Fang, and Jimmy Lin. 2017. Anserini: Enabling the Use of Lucene for Information Retrieval Research. *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval* (2017).

[19] Peilin Yang, Hui Fang, and Jimmy Lin. 2018. Anserini: Reproducible Ranking Baselines Using Lucene. *ACM J. Data Inf. Qual.* 10 (2018), 16:1–16:20.

[20] Peilin Yang, Hui Fang, and Jimmy Lin. 2018. Anserini: Reproducible Ranking Baselines Using Lucene. *J. Data and Information Quality* 10, 4, Article 16 (Oct. 2018), 20 pages. https://doi.org/10.1145/3239571

[21] Wei Yang, Haotian Zhang, and Jimmy Lin. 2019. Simple Applications of BERT for Ad Hoc Document Retrieval. *arXiv:1903.10972 [cs]* (March 2019). http://arxiv.org/abs/1903.10972 arXiv: 1903.10972.