An Evaluation of Weakly-Supervised DeepCT in the TREC 2019 Deep Learning Track

Zhuyun Dai and Jamie Callan
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA 15213, USA
{zhuyund, callan}@cs.cmu.edu

ABSTRACT

This paper describes our participation in the TREC 2019 Deep Learning Track document ranking task. We developed a deep learning based document term weighting approach based on our previous work of DeepCT. It used the contextualized token embeddings generated by BERT to estimate a term's importance in passages, and combines passage term weights into document-level term weights. The weighted document is stored in an ordinary inverted index and searched using a multi-field BM25, which is efficient. We tested two ways of training DeepCT: a query-based method using sparse relevant query-document pairs, and a weakly-supervised method using document title-body pairs.

1. INTRODUCTION

This paper describes our submissions to TREC 2019 Deep Learning (DL) Track document ranking task [2]. In this challenge, we took a simple BM25 retrieval approach, but we estimated document term weights using a deep neural network rather than using standard term frequency. The term weight estimation is an extension of our previous work on the deep contextual term weighting and indexing framework DeepCT [4].

Standard term frequency based weighting schemes are widely used in Information Retrieval. They assume that a term is more important to a document if it is mentioned frequently in the document. However, this is not necessarily true –in some cases, term frequency distribution can be flat or noisy.

DeepCT is a term weighting and indexing framework aiming to generate indexing weights based on a term's meaning in a passage. It uses BERT [5] to extract contextual features of a term in a specific passage, and maps them to a context-aware term weight. The weights are stored in an ordinary inverted index, replacing the original term frequency (tf) field. At query time, the resulting index, DeepCT-Index, can be efficiently searched using bag-of-words retrieval models such as BM25. In our previous research [4], the new term weights improved standard bag-of-words retrieval models such as BM25 in two passage retrieval tasks as they better reflect essential concepts in the passages.

One contribution of this work is that we extended DeepCT to support long documents. Due to the input length limitation of BERT [5], documents with more than 512 tokens can not be directly input into DeepCT. In this work, we first estimated a term's local importance in each passage using DeepCT. Then we combined the passage-specific term weights into a document bag-of-words representation. Finally, we constructed a standard inverted index using the weighted document bag-of-words representation.

DeepCT-Index is built offline. At query time, documents are ranked by the standard BM25 algorithm. Multiple document fields, such as titles, URLs, and bodies, and combined by a simple ensemble approach that weighted-sums the retrieval scores on each field. This is a single stage, bag-of-words retrieval, which is efficient.

Another contribution of this work is a title-based weakly-supervised approach of training DeepCT. We developed a title-based training strategy that solely uses document title-body pairs. The title-based training method was designed for cold-start scenarios and low-resource domains. Our submissions compared the title-based training strategy with the query-based training strategy [4], which requires relevance labels.

Evaluation results show that the title-trained DeepCT is more accurate than the query-trained version at the top of the ranking, but is less effective at deeper positions. The gap can be bridged by combining the relevance-trained DeepCT with the original tf weighted documents.

2. METHOD

Our approach extended DeepCT from our previous work [4]. Given a document d, it first estimates passage-level term weights using DeepCT. Next, it combines the passage-level term weights into document-level term weights. The output is a document bag-of-words representation that can be stored in a standard inverted index and retrieved by common bag-of-words retrieval models like BM25 .

2.1 Passage Term Weighting using DeepCT

We first estimated a term's importance in a passage. Instead of using standard term frequency based weighting, we used DeepCT to identify essential terms that are semantically important.

Given a document d, we split it into a sequence of passages. Passages consist of natural sentences of up to about 300 words. We applied DeepCT on the passages to weight their terms. This section briefly describes DeepCT; [4] provides more details.

Given a passage p, DeepCT generates contextual token embeddings using BERT [7], and feeds these contextualized token embeddings into a linear layer. It maps a token's contextual embedding into a real-number weight:

$$\hat{y}_{t,p} = wT_{\text{BERT}}(t,p) + b, \quad (1)$$

Table 1: Submitted runs and evaluation results.

Runs	Training Labels	BM25E Fields	MAP@1000	NDCG@1000	P@10
`dct_qp_bm25e`	Sparse Relevance Labels	title, URL, DeepCT body	0.2858	0.5477	0.6140
`dct_tp_bm25e`	Document Titles	title, URL, DeepCT body	0.2628	0.5240	0.6372
`dct_tp_bm25e2`	Document Titles	title, URL, *tf* body, DeepCT body	0.2852	0.5490	0.6349

$T_{\text{BERT}}(t,p)$ is token t's contextualized embedding in passage p; w and b are the linear combination weights and bias; and, $\hat{y}_{t,p}$ is the predicted weight for token t in the passage p. $\hat{y}_{t,p}$ are mostly in the range of 0-1. This is because our training labels are in 0-1, so the model learns to generate predictions also in that range.

We scale the predictions into a *tf*-like integer that can be used with existing retrieval models. This weight is called tf_{DeepCT} to convey that it is an alternate way of representing the importance of term t in document d using DeepCT:

$$tf_{\text{DeepCT}}(t,p_i) = round(N * \sqrt{\hat{y}_{t,p_i}}), \tag{2}$$

$\hat{y}_{t,p}$ is the predicted weight from Eq (1). N scales the weight into a integer range. Our TREC submissions used $N = 100$, which keeps two digit precision. Different from the original DeepCT [4], we add the square-root function for smoothing.

At the end of this step, for a document d, we generated a sequence of bag-of-words passage vectors.

2.2 Extended DeepCT for Document Retrieval

Due to the input length limitation of BERT [5], DeepCT only supports short passages [4]. This work extended it also to support long documents.

The extended DeepCT generates a document bag-of-words representation from the passage-specific ones, and performs retrieval at the document level. A term's importance is a weighted sum of its DeepCT importance in each passage: $tf_{\text{DeepCT}}(t,d) = \sum_{i=1}^{n} \frac{1}{i} \times tf_{\text{DeepCT}}(t,p_i)$. $\frac{1}{i}$ discounts passages based on the position, following findings in prior research that passages at the beginning of a document tend to attract more attention from readers and are more important for relevance estimation [8, 1].

We store the weighted documents into an inverted index, where the new term weights replace the standard term frequency fields in the inverted lists [3]. This new index is called DeepCT-Index. DeepCT-Index is expected to improve retrieval by identifying key terms in a document.

DeepCT-Index is retrieved using the out-of-the-box BM25 formula. During retrieval, the term frequency field in BM25 is replaced with the term weights stored in the DeepCT-Index. Multiple document fields, such as titles, URLs, and bodies are combined by a simple ensemble approach that weighted-sums the retrieval scores on each field (BM25E).

In terms of efficiency, DeepCT-Index does not introduce new words into documents, so the index does not become larger. Usually, DeepCT-Index reduces the index size as some terms' weight becomes 0 during the scaling in Eq (2), thus can be faster.

2.3 Training DeepCT

As described in [4], DeepCT is trained on a passage-level per-token regression task. Assume we have the ground truth term weight for a term t in a passage p, denoted as $y_{t,p}$. DeepCT minimizes the mean square error between the predicted weights $\hat{y}$ and the target weights y:

$$MSE = \sum_{p} \sum_{t \in p} (y_{t,p} - \hat{y}_{t,p})^2. \tag{3}$$

In this work, we explored two ways of generating ground truth labels $y_{t,p}$, as described below.

Query-based Training. This approach follows the training strategy in our previous work [4]. Given a training document d, its passages $P_d = \{p_1, ..., p_n\}$, and its relevant queries $Q_d = \{q_1, ..., q_b\}$, we generate the *query-based* training labels for each passage using the query term-recall [4]:

$$y_{t,p} = \frac{|Q_{d,t}|}{|Q_d|}, p \in \{p_1, ..., p_n\}. \tag{4}$$

In this work, query-based training uses the sparse relevance labels provided by the training set of the TREC 2019 DL Track document ranking task.

Title-based Training. Titles provide a short summary of what a document is about. A term is considered essential to the document if the title mentions it.

Given a training document d, its passages $P_d = \{p_1, ..., p_n\}$, and its title $title_d$, the title-based weak-supervision approach generates passage level training data as the following:

$$y_{t,p} = 0 \text{ if } t \in title_d \text{ else } 0, p \in \{p_1, ..., p_n\} \tag{5}$$

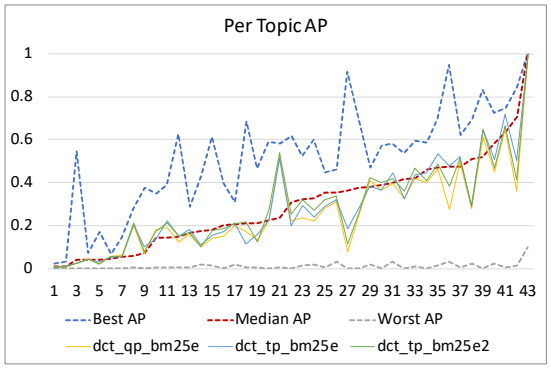
In this work, title-based training used randomly sampled documents from the TREC 2019 DL Track document ranking corpus. It does not require any labeled data, making it a good fit for low-resource domains and cold-start scenarios.

3. EXPERIMENTAL SETUP

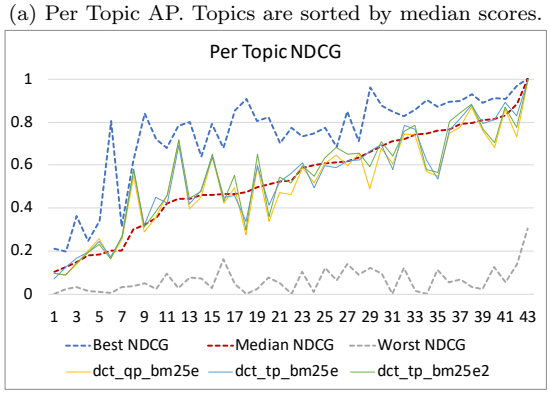
We trained all DeepCT models for 100K steps with a batch size of 16 and a learning rate of $2e{-}5$. The BERT component was initialized with the official pre-trained BERT (uncased, base model) [7]. The max input length of BERT was set to 512 tokens. The scaling coefficient N in Eq (2) were set to 100.

The trained DeepCT models were used to weight all document bodies in the MS-MARCO document ranking collection. The whole collection consists of 4 million documents. We used Lucene to build the DeepCT-Indexes. Separate indexes were built for other documents fields, including the title and the URL.

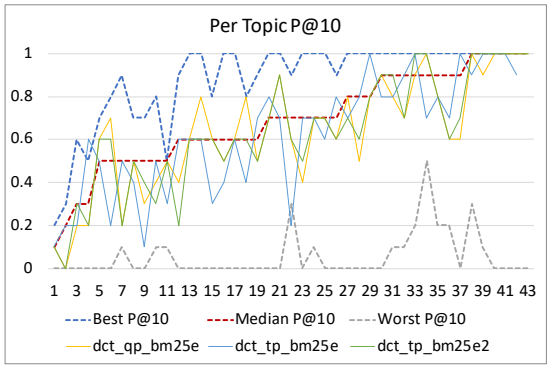
We used the BM25 implementation from the Anserini [9] toolkit to retrieve documents from the indexes. BM25 scores on different document fields are combined through a simple ensemble approach that weighted-sums the scores (BM25E). The parameters of BM25E were tuned on the evaluation query sets, including the k_1 and b parameters in BM25 and the field weights in the ensemble model.



(a) Per Topic AP. Topics are sorted by median scores.



(b) Per Topic NDCG. Topics are sorted by median scores.



(c) Per Topic P@10. Topics are sorted by median scores.

Figure 1: Per Topic Score Distribution. The per-topic AP/NDCG/P@10 scores achieved by our 3 runs, and the minimum, maximum, and median scores achieved across the 38 submitted runs.

4. SUBMITTED RUNS AND EVALUATION

We submitted three runs with different configurations. These runs differ in how DeepCT was trained, and which fields were used by the retrieval model BM25E.

- `dct_qp_bm25e`: This run used the query-based strategy to train DeepCT, relying on the sparse relevance labels provided by the TREC 2019 DL Track training set. Document bodies were indexed with the DeepCT term weights. At query time, documents were ranked by the ensemble of BM25 scores (BM25E) from the title, URL, and the DeepCT weighted body.

- `dct_tp_bm25e`: This run used the title-based strategy to train DeepCT, relying on the document title-body pairs from the TREC 2019 DL Track document collection. Document bodies were indexed with the DeepCT term weights. At query time, documents were ranked by the ensemble of BM25 scores (BM25E) from the title, URL, and the DeepCT weighted body.

- `dct_tp_bm25e2`: This run used the same title-trained DeepCT model as used in `dct_tp_bm25e`. At query time, documents were ranked by the ensemble of BM25 scores (BM25E) from the title, URL, the DeepCT weighted body, and the original term frequency (*tf*) weighted body.

Table 1 lists the official overall evaluation results of the submitted runs.

Query-based Training vs. Title-based Training As can be seen from Table 1, when the retrieval model is kept the same, the query-trained DeepCT (`dct_qp_bm25e`) performed better than the title-trained one (`dct_qp_bm25e`) at MAP@1000 and NDCG@1000, but was worse in terms of P@10.

It was to our surprise that the query-trained model had a lower P@10. This might be due to the differences between the training labels and the evaluation labels. The training used sparse relevance labels derived from user clicks [6]. The official TREC evaluation used manual, graded relevance judgments. The two types of labels might prefer different types of documents.

The title-trained DeepCT had a higher P@10. It confirmed our assumption that the titles provide a high-quality summary of the *aboutness* of the documents, and are helpful for determining term importance. It indicates that it is promising to train DeepCT solely based on the content and internal structures of documents, without using any manual relevance annotations or user data.

Ensemble of Multiple Document Representations. Comparing `dct_qp_bm25e` and `dct_qp_bm25e2`, we found that adding the original *tf*-weighted documents to the DeepCT weighted documents was beneficial. The precision at the top of the ranking (P@10) remained high, while the lower rankings were largely improved (MAP@1000 and NDCG@1000). *tf* and DeepCT provide two different representations of the same document text – one based on term frequency and one based on deep language modeling. Our results indicate that the two representations reflect different characteristics of the documents, and were both useful for document retrieval.

Per-Topic Evaluation. Figure 1 shows the per-topic evaluation results. It compares the per-topic scores achieved by our runs to the min, max and median scores achieved across the 38 submitted runs in the TREC 2019 Deep Learning track document ranking task. First, our three runs had consistent behavior, indicating that the different training strategies do not lead to very different models. When compared to runs submitted from other teams, our runs are around the median in terms of MAP and NDCG (@1000). In terms of P@10, our runs had more topics lower than the median. Our runs are simple single-stage, bag-of-words retrieval from the entire collection. Hence their precision is likely to be lower than most re-ranking methods, such as learning-to-rank approaches. In general, our runs seem to be of low-risk – they were rarely near the min scores.

5. CONCLUSION

In this paper, we present our methods in the TREC 2019 Deep Learning Track document ranking task. We took an efficient BM25 retrieval approach, but estimated the indexing term weights using the DeepCT framework based on our previous work [4]. Evaluation results suggest that the model can benefit from multiple document representations. The analysis also reveals a promising future direction of training DeepCT using the internal structures of documents without manual relevance annotations.

ACKNOWLEDGMENTS

This research was supported by NSF grant IIS-1815528. Any opinions, findings, and conclusions in this paper are the authors' and do not necessarily reflect those of the sponsor.

6. REFERENCES

[1] M. Catena, O. Frieder, C. I. Muntean, F. M. Nardini, R. Perego, and N. Tonellotto. Enhanced news retrieval: Passages lead the way! In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2019.

[2] N. Craswell, B. Mitra, E. Yilmaz, and D. Campos. Overview of the trec 2019 deep learning track. In *TREC (to appear)*, 2019.

[3] W. B. Croft, D. Metzler, and T. Strohman. *Search Engines - Information Retrieval in Practice*. Pearson Education, 2009.

[4] Z. Dai and J. Callan. Context-aware sentence/passage term importance estimation for first stage retrieval. *arXiv preprint arXiv:1910.10687*, 2019.

[5] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[6] T. Nguyen, M. Rosenberg, X. Song, J. Gao, S. Tiwary, R. Majumder, and L. Deng. Ms marco: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268*, 2016.

[7] I. Tenney, D. Das, and E. Pavlick. Bert rediscovers the classical nlp pipeline. *arXiv preprint arXiv:1905.05950*, 2019.

[8] Z. Wu, J. Mao, Y. Liu, M. Zhang, and S. Ma. Investigating passage-level relevance and its role in document-level relevance judgment. In *Proceedings of the 42nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2019.

[9] P. Yang, H. Fang, and J. Lin. Anserini: Enabling the use of lucene for information retrieval research. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2017.