Understanding the Behaviors of BERT in Ranking

Yifan Qiao
Tsinghua University
qiaoyf15@mails.tsinghua.edu.cn

Chenyan Xiong
Microsoft Research
Chenyan.Xiong@microsoft.com

Zhenghao Liu
Tsinghua University
liu-zh16@mails.tsinghua.edu.cn

Zhiyuan Liu
Tsinghua University
liuzy@tsinghua.edu.cn

几种基于微调的bert的模型：
任务：MSMARCO 的 passage reranking
TREC 的 document ranking

ABSTRACT

This paper studies the performances and behaviors of BERT in ranking tasks. We explore several different ways to leverage the pre-trained BERT and fine-tune it on two ranking tasks: MS MARCO passage reranking and TREC Web Track ad hoc document ranking. Experimental results on MS MARCO demonstrate the strong effectiveness of BERT in question-answering focused passage ranking tasks, as well as the fact that BERT is a strong interaction-based seq2seq matching model. Experimental results on TREC show the gaps between the BERT pre-trained on surrounding contexts and the needs of ad hoc document ranking. Analyses illustrate how BERT allocates its attentions between query-document tokens in its Transformer layers, how it prefers semantic matches between paraphrase tokens, and how that differs with the soft match patterns learned by a click-trained neural ranker.

1 INTRODUCTION

In the past several years, neural information retrieval (Neu-IR) research has developed several effective ways to improve ranking accuracy. *Interaction-based* neural rankers soft match query-documents using their term interactions [3]; *Representation-based embeddings* capture relevance signals using distributed representations [7, 8]; large capacity networks learn relevance patterns using large scale ranking labels [1, 5, 7]. These techniques lead to promising performances on various ranking benchmarks [1, 3, 5, 7].

Recently, BERT, the pre-trained deep bidirectional Transformer, has shown strong performances on many language processing tasks [2]. BERT is a very deep language model that is pre-trained on the surrounding context signals in large corpora. Fine-tuning its pre-trained deep network works well on many downstream sequence to sequence (seq2seq) learning tasks. Different from seq2seq learning, previous Neu-IR research considers such surrounding-context-trained neural models not as effective in search as relevance modeling [7, 8]. However, on the MS MARCO passage ranking task, fine-tuning BERT and treating ranking as a classification problem outperforms existing Neu-IR models by large margins [4].

This paper studies the performances and properties of BERT in ad hoc ranking tasks. We explore several ways to use BERT in ranking, as representation-based and interaction-based neural rankers, as in combination with standard neural ranking layers. We study the behavior of these BERT-based rankers on two benchmarks: the MS MARCO passage ranking task, which ranks answer passages for questions, and TREC Web Track ad hoc task, which ranks ClueWeb documents for keyword queries.

Our experiments observed rather different performances of BERT-based rankers on the two benchmarks. On MS MARCO, fine-tuning BERT significantly outperforms previous state-of-the-art Neu-IR models, and the effectiveness mostly comes from its strong cross question-passage interactions. However, on TREC ad hoc ranking, BERT-based rankers, even further pre-trained on MS MARCO ranking labels, perform worse than feature-based learning to rank and a Neu-IR model pre-trained on user clicks in Bing log.

We further study the behavior of BERT through its learned attentions and term matches. We illustrate that BERT uses its deep Transformer architecture to propagate information more globally on the text sequences through its attention mechanism, compared to interaction-based neural rankers which operate more individually on term pairs. Further studies reveal that BERT focuses more on document terms that directly match the query. It is similar to the semantic matching behaviors of previous surrounding context-based seq2seq models, but different from the relevance matches neural rankers learned from user clicks.

2 BERT BASED RANKERS

This section describes the notable properties of BERT and how it is used in ranking.

2.1 Notable Properties of BERT

We refer readers to the BERT and Transformer papers for their details [2, 6]. Here we mainly discuss its notable properties that influence its usage in ranking.

Large Capacity. BERT uses standard Transformer architecture—multi-head attentions between all term pairs in the text sequence—but makes it very deep. Its main version, `BERT-Large`, includes 24 Transformer layers, each with 1024 hidden dimensions and 16 attention heads. It in total has 340 million learned parameters, much bigger than typical Neu-IR models.

Pretraining. BERT learns from the surrounding context signals in Google News and Wikipedia corpora. It is learned using two tasks: the first predicts random missing words (15%) using the rest of the sentence (Mask-LM); the second predicts whether two sentences appear next to each other. In the second task, the two sentences are concatenated to one sequence; a special token "[SEP]" marks the sequence boundaries. Its deep network is very resource consuming in training: `BERT-Large` takes four days to train on 64 TPUs and easily takes months on typical GPUs clusters.

Fine Tuning. End-to-end training BERT is unfeasible in most academic groups due to resource constraints. It is suggested to use the pre-trained BERT as a fine-tuning method [2]. BERT provides a

"[CLS]" token at the start of the sequence, whose embeddings are treated as the representation of the text sequence(s), and suggests to add task-specific layers on the "[CLS]" embedding in fine-tuning.

2.2 Ranking with BERT

We experiment with four BERT based ranking models: `BERT (Rep)`, `BERT (Last-Int)`, `BERT (Mult-Int)`, and `BERT (Term-Trans)`. All four methods use the pre-trained BERT to obtain the representation of the query q, the document d, or the concatenation of the two qd. In the concatenation sequence qd, the query and document are concatenated to one sequence with boundary marked by a marker token ("[SEP]").

The rest of this section uses subscript i, j, or cls to denote the tokens in q, d, or qd, and superscript k to denote the layer of BERT's Transformer network: $k = 1$ is the first layer upon word embedding and $k = 24$ or "last" is the last layer. For example, $\vec{qd}^{k}_{cls}$ is the embedding of the "[CLS]" token, in the k-th layer of BERT on the concatenation sequence qd.

BERT (Rep) uses BERT to represent q and d:

拿q和d的最后一层的CLS处的表达，计算cos

$$\texttt{BERT (Rep)}(q, d) = \cos(\vec{q}^{\,\text{last}}_{cls}, \vec{d}^{\,\text{last}}_{cls}). \tag{1}$$

It first uses the last layers' "[CLS]" embeddings as the query and document *representations*, and then calculates the ranking score via their cosine similarity (cos). Thus it is a *representation-based* ranker.

BERT (Last-Int) applies BERT on the concatenated qd sequence:

QD的最后一层的CLS处的表达，通过一层全连接

$$\texttt{BERT (Last-Int)}(q, d) = w^{T} \vec{qd}^{\,\text{last}}_{cls}. \tag{2}$$

It uses the last layer's "[CLS]" as the matching features and combines them linearly with weight w. It is the recommended way to use BERT [2] and is first applied to MARCO passage ranking by Nogueira and Cho [4]. The ranking score from `BERT (Last-Int)` includes all term pair interactions between the query and document via its Transformer's cross-match attentions [6]. Thus it is an *interaction-based* ranker.

BERT (Mult-Int) is defined as:

拿QD的每一层的CLS处的表达，通过一层全连接

$$\texttt{BERT (Mult-Int)}(q, d) = \sum_{1 \le k \le 24} (w^{k}_{Mult})^{T} \vec{qd}^{k}_{cls}. \tag{3}$$

It extends `BERT (Last-Int)` by adding the matching features $\vec{qd}^{k}_{cls}$ from all BERT's layers, to study whether different layers of BERT provide different information.

BERT (Term-Trans) adds a neural ranking network upon BERT, to study the performance of their combinations:

$$s^{k}(q, d) = \text{Mean}_{i,j}(\cos(\text{relu}(P^{k} \vec{q}^{\,k}_{i}), \text{relu}(P^{k} \vec{d}^{\,k}_{j}))) \tag{4}$$

$$\texttt{BERT (Term-Trans)}(q, d) = \sum_{k} w^{k}_{trans} s^{k}(q, d). \tag{5}$$

It first constructs the translation matrix between query and document, using the cosine similarities between the projections of their contextual embeddings. Then it combines the translation matrices from all layers using mean-pooling and linear combination.

All four BERT based rankers are fine-tuned from the pre-trained BERT-Large model released by Google. The fine-tuning uses classification loss, i.e., to classify whether a query-document pair is relevant or not, following the prior research [4]. We experimented with pairwise ranking loss but did not observe any difference.

微调时，使用pairwise ranking loss和分类loss没有差别。

3 EXPERIMENTAL METHODOLOGIES

Datasets. Our experiments are conducted on `MS MARCO` passage reranking task and TREC Web Track ad hoc tasks with `ClueWeb` documents.

`MS MARCO` includes question-alike queries sampled from Bing search log and the task is to rank candidate passages based on whether the passage contains the answer for the question[1]. It includes 1,010,916 training queries and a million expert annotated answer passage relevance labels. We follow the official train/develop split, and use the given "Train Triples Small" to fine-tune BERT.

`ClueWeb` includes documents from ClueWeb09-B and queries from TREC Web Track ad hoc retrieval task 2009-2012. In total, 200 queries with relevance judgements are provided by TREC. Our experiments follow the same set up in prior research and use the processed data shared by their authors [1]: the same 10-fold cross validation, same data pre-processing, and same top 100 candidate documents from Galago SDM to re-rank.

We found that the TREC labels alone are not sufficient to fine-tune BERT nor train other neural rankers to outperform SDM. Thus we decided to first pre-train all neural methods on MS MARCO and then fine-tune them on ClueWeb.

Evaluation Metrics. MS MARCO uses MRR@10 as the official evaluation. Results on the Develop set re-rank top 100 from BM25 in our implementation. Results on Evaluations set are obtained from the organizers and re-rank top 1000 from their BM25 implementation. ClueWeb results are evaluated by NDCG@20 and ERR@20, the official evaluation metrics of TREC Web Track.

dev是在自己实现的BM25的top100上rerank。Eval是在官方的BM25的top1000上rerank

Statistical significance is tested by permutation tests with $p < 0.05$, except on MS MARCO Eval where per query scores are not returned by the leader board.

Compared Methods. The BERT based rankers are compared with the following baselines:

- `Base` is the base retrieval model that provides candidate documents to re-rank. It is BM25 on MS MARCO and Galago-SDM on ClueWeb.
- `LeToR` is the feature-based learning to rank. It is RankSVM on MS MARCO and Coordinate Ascent on ClueWeb.
- `K-NRM` is the kernel-based interaction-based neural ranker [7].
- `Conv-KNRM` is the n-gram version of `K-NRM`.

`K-NRM` and `Conv-KNRM` results on ClueWeb are obtained by our implementations and pre-trained on MS MARCO. We also include `Conv-KNRM (Bing)` which is the same Conv-KNRM model but pre-trained on Bing clicks by prior research [1]. The rest baselines reuse the existing results from prior research. Keeping experimental setups consistent makes all results directly comparable.

Implementation Details. All BERT rankers are trained using Adam optimizer and learning rate 3e-6, except `Term-Trans` which trains the projection layer with learning rate 0.002. On one typical GPU, the batch size is 1 at most; fine-tuning takes on average one day to converge. Convergence is determined by the loss on a small sample of validation data (MS MARCO) or the validation fold (ClueWeb). In comparison, K-NRM and Conv-KNRM take about 12 hours to converge on MS MARCO and one hour on ClueWeb. On MS MARCO all rankers take about 5% training triples to converge.

[1]http://msmarco.org

Table 1: Ranking performances. Relative performances in percentages are compared to LeToR, the feature-based learning to rank. Statistically significant improvements are marked by † (over Base), ‡ (over LeToR), § (over K-NRM), and ¶ (over Conv-KNRM). Neural methods on ClueWeb are pre-trained on MS MARCO, except Conv-KNRM (Bing) which is trained on user clicks.

	MS MARCO Passage Ranking				ClueWeb09-B Ad hoc Ranking			
Method	**MRR@10 (Dev)**		**MRR@10 (Eval)**		**NDCG@20**		**ERR@20**	
Base BM25	0.1762	−9.45%	0.1649	+13.44%	0.2496§	−6.89%	0.1387	−14.25%
LeToR	0.1946	–	0.1905	–	0.2681	–	0.1617	–
K-NRM	0.2100†‡	+7.92%	0.1982	+4.04%	0.1590	−40.68%	0.1160	−28.26%
Conv-KNRM	0.2474†‡§	+27.15%	0.2472	+29.76%	0.2118§	−20.98%	0.1443§	−10.78%
Conv-KNRM (Bing)	n.a.	n.a.	n.a.	n.a.	0.2872†‡§¶	+7.12%	0.1814†‡§¶	+12.18%
BERT (Rep)	0.0432	−77.79%	0.0153	−91.97%	0.1479	−44.82%	0.1066	−34.05%
BERT (Last-Int)	0.3367†‡§¶	+73.03%	0.3590	+88.45%	0.2407§¶	−10.22%	0.1649†§¶	+2.00%
BERT (Mult-Int)	0.3060†‡§¶	+57.26%	0.3287	+72.55%	0.2407§¶	−10.23%	0.1676†§¶	+3.64%
BERT (Term-Trans)	0.3310†§¶	+70.10%	0.3561	+86.93%	0.2339§¶	−12.76%	0.1663†§¶	+2.81%

在预训练的BERT的基础上使用MSMARCO微调

使用Bing的点击数据预训练ConvKNRM

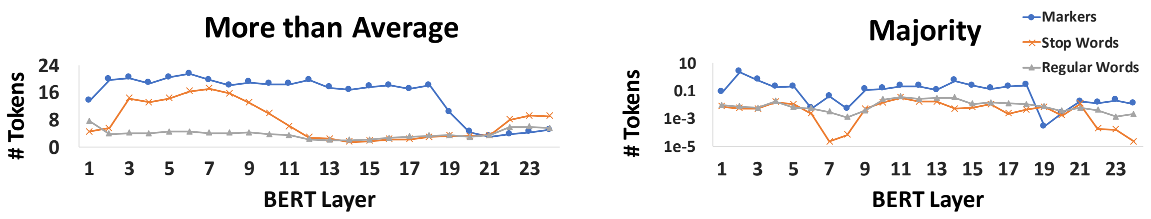

Figure 1: The attentions to Markers, Stopwords, and Regular Words in BERT (Last-Int). X-axes mark layer levels from shallow (1) to deep (24). Y-axes are the number of tokens sending More than Average or Majority attentions to each group.

4 EVALUATIONS AND ANALYSES

This section evaluates the performances of BERT-based rankers and studies their behaviors.

4.1 Overall Performances

Table 1 lists the evaluation results on MS MARCO (left) and ClueWeb (right). BERT-based rankers are very effective on MS MARCO: All interaction-based BERT rankers improved Conv-KNRM, a previous state-of-the-art, by 30%-50%. The advantage of BERT in MS MARCO lies in the cross query-document attentions from the Transformers: BERT (Rep) applies BERT on the query and document individually and discard these cross sequence interactions, and its performance is close to random. BERT is an *interaction-based* matching model and is not suggested to be used as a representation model.

BERT的优势在于transformer带来的cross attention。（因为实验结果表明分别获得表达，效果并不好）

The more complex architectures in Multi-Int and Term-Trans perform worse than the simplest BERT (Last-Int), even with a lot of MARCO labels to fine-tune. It is hard to modify the pre-trained BERT dramatically in fine-tuning. End-to-end training may make modifying pre-trained BERT more effective, but that would require more future research in how to make BERT trainable in accessible computing environments.

BERT-based rankers behave rather differently on ClueWeb. Although pre-trained on large corpora and then on MARCO ranking labels, none of BERT models significantly outperforms LeToR on ClueWeb. In comparison, Conv-KNRM (Bing), the same Conv-KNRM model but pre-trained on Bing user clicks [1], performs the best on ClueWeb, and much better than Conv-KNRM pretrained on MARCO labels. These results demonstrate that MARCO passage ranking is closer to seq2seq task because of its question-answering focus, and BERT's surrounding context based pre-training excels in this setting. In comparison, TREC ad hoc tasks require different signals other than surrounding context: pre-training on user clicks is more effective than on surrounding context based signals.

为什么不是因为数据集只有200个query，微调的不够？？？

4.2 Learned Attentions

This experiment illustrates the learned attention in BERT, which is the main component of its Transformer architecture.

Our studies focus on MS MARCO and BERT (Last-Int), the best performing combination in our experiments, and randomly sampled 100 queries from MS MARCO Dev. We group the terms in the candidate passages into three groups: Markers ("[CLS]" and "[SEP]"), Stopwords, and Regular Words. The attentions allocated to each group is shown in Figure 1.

The markers received most attention. Removing these markers decreases the MRR by 15%: BERT uses them to distinguish the two text sequences. Surprisingly, the stopwords received as much attention as non-stop words, but removing them has no effect in MRR performances. BERT learned these stopwords not useful and dumps redundant attention weights on them.

As the network goes deeper, less tokens received the majority of other tokens attention: the attention spreads more on the whole sequence and the embeddings are contextualized. However, this does not necessarily lead to more global matching decisions, as studied in the next experiment.

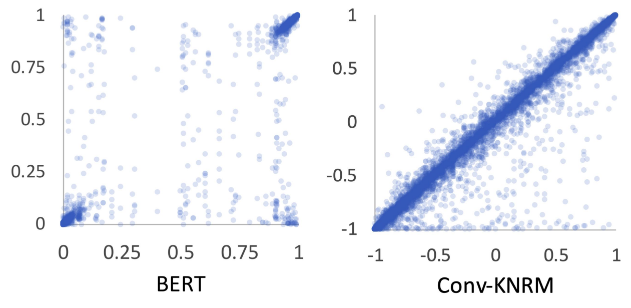

Figure 2: Influences of removing regular terms in BERT (Last-Int) and Conv-KNRM on MS MARCO. Each point corresponds to one query-passage pair with a random regular term removed from the passage. X-axes mark the original ranking scores and Y-axes are the scores after term removal.

4.3 Learned Term Matches

This experiment studies the learned matching patterns in BERT (Last-Int) and compares it to Conv-KNRM. The same MS MARCO Dev sample from last experiment is used.

We first study the influence of a term by comparing the ranking score of a document with and without the term. For each query-passage pair, we randomly remove a non-stop word, calculate the ranking score using BERT (Last-Int) or Conv-KNRM, and plot it w.r.t the original ranking score in Figure 2.

Figure 2 illustrates two interesting behaviors of BERT. First, it assigns more extreme ranking scores: most pairs receive either close to 1 or 0 ranking scores in BERT, while the ranking scores in Conv-KNRM are more uniformly distributed. Second, there are a few terms in each document that determine the majority of BERT's ranking scores; removing them significantly changes the ranking score—drop from 1 to near 0, while removing the majority of terms does not matter much in BERT—most points are grouped in the corners. It indicates that BERT is well-trained from the large scale pre-training. In comparison, terms contribute more evenly in Conv-KNRM; removing single term often varies the ranking scores of Conv-KNRM by some degree, shown by the wider band near the diagonal in Figure 2, but not as dramatically as in BERT.

We manually examined those most influential terms in BERT (Last-Int) and Conv-KNRM. Some examples of those terms are listed in Table 2. The exact match terms play an important role in BERT (Last-Int); we found many of the influential terms in BERT are those appear in the question or close paraphrases. Conv-KNRM, on the other hand, prefers terms that are more loosely related to the query in search [1]. For example, on MS MARCO, it focuses more on the terms that are the role of milk in macchiato ("visible mark"), the show and the role Sinbad played ("Cosby" and "Coach Walter"), and the task related to Personal Meeting ID ("schedule").

These observations suggest that, BERT's pre-training on surrounding contexts favors text sequence pairs that are closer in their semantic meanings. It is consistent with previous observations in Neu-IR research, that such surrounding context trained models are not as effective in TREC-style ad hoc document ranking for keyword queries [1, 7, 8].

随机去掉passage中的一个非停顿词，一般来说对BERT的影响较小，但对ConvKNRM而言，一般会导致效果变差。
而且去掉之后，对BERT有较大影响的那些词通常是精确匹配的词或释义的词。（个人理解：从侧面说明BERT有能力捕捉精确匹配信号）本文的结论是，BERT预训练的任务（NSE）帮助它可以捕捉具有相似语义的句子。

Table 2: Example of most influential terms in MS MARCO passages in BERT and Conv-KNRM.

Query: "What is a macchiato coffee drink"	
BERT (Last-Int)	macchiato, coffee
Conv-KNRM	visible mark
Query: "What shows was Sinbad on"	
BERT (Last-Int)	Sinbad
Conv-KNRM	Cosby, Coach Walter
Query: "What is a PMI id"	
BERT (Last-Int)	PMI
Conv-KNRM	schedule a meeting

5 CONCLUSIONS AND FUTURE DIRECTION

This paper studies the performances and behaviors of BERT in MS MARCO passage ranking and TREC Web Track ad hoc ranking tasks. Experiments show that BERT is an interaction-based seq2seq model that effectively matches text sequences. BERT based rankers perform well on MS MARCO passage ranking task which is focused on question-answering, but not as well on TREC ad hoc document ranking. These results demonstrate that MS MARCO, with its QA focus, is closer to the seq2seq matching tasks where BERT's surrounding context based pre-training fits well, while on TREC ad hoc document ranking tasks, user clicks are better pre-training signals than BERT's surrounding contexts.

对于MSMARCO这样更类似于QA的seq2seq匹配任务，BERT更好。。但对于TREC的ad-hoc retrieval任务，用用户点击数据进行预训练更好。

Our analyses show that BERT is a strong matching model with globally distributed attentions over the entire contexts. It also assigns extreme matching scores to query-document pairs; most pairs get either one or zero ranking scores, showing it is well tuned by pre-training on large corpora. At the same time, pre-trained on surrounding contexts, BERT prefers text pairs that are semantically close. This observation helps explain BERT's lack of effectiveness on TREC-style ad hoc ranking which is considered to prefer pre-training from user clicks than surrounding contexts.

Our results suggest the need of training deeper networks on user clicks signals. In the future, it will be interesting to study how a much deeper model—as big as BERT—behaves compared to both shallower neural rankers when trained on relevance-based labels.

REFERENCES

[1] Zhuyun Dai, Chenyan Xiong, Jamie Callan, and Zhiyuan Liu. 2018. Convolutional Neural Networks for Soft-Matching N-Grams in Ad-hoc Search. In *Proceedings of WSDM 2018*. ACM, 126–134.
[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint* (2018).
[3] Jiafeng Guo, Yixing Fan, Qingyao Ai, and W Bruce Croft. 2016. A deep relevance matching model for ad-hoc retrieval. In *Proceedings of CIKM 2016*. ACM, 55–64.
[4] Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage Re-ranking with BERT. *arXiv preprint arXiv:1901.04085* (2019).
[5] Liang Pang, Yanyan Lan, Jiafeng Guo, Jun Xu, Jingfang Xu, and Xueqi Cheng. 2017. Deeprank: A new deep architecture for relevance ranking in information retrieval. In *Proceedings of CIKM 2017*. ACM, 257–266.
[6] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of NeuIPS 2017*. 5998–6008.
[7] Chenyan Xiong, Zhuyun Dai, Jamie Callan, Zhiyuan Liu, and Russell Power. 2017. End-to-end neural ad-hoc ranking with kernel pooling. In *Proceedings of SIGIR 2017*. ACM, 55–64.
[8] Hamed Zamani and W Bruce Croft. 2017. Relevance-based word embedding. In *Proceedings of SIGIR 2017*. ACM, 505–514.

对于TREC类型的Ad-hoc retrieval（即关键词查询），基于上下文训练的模型效果并不好。。就像Conv_KNRM中有提到，也许基于上下文窗口训练的word embedding，并不适合ad-hoc retrieval任务。