An in-depth analysis of passage-level label transfer for contextual document ranking

Rudra, Koustav; Fernando, Zeon Trevor; Anand, Avishek

DOI
10.1007/s10791-023-09430-5

Publication date
2023

Document Version
Final published version

Published in
Information Retrieval Journal

Information Retrieval Journal (2023) 26:13
https://doi.org/10.1007/s10791-023-09430-5

An in-depth analysis of passage-level label transfer for contextual document ranking

Koustav Rudra[1] · **Zeon Trevor Fernando**[2] · **Avishek Anand**[3]

Received: 4 March 2021 / Accepted: 15 November 2023

Abstract
Pre-trained contextual language models such as BERT, GPT, and XLnet work quite well for document retrieval tasks. Such models are fine-tuned based on the query-document/query-passage level relevance labels to capture the ranking signals. However, the documents are longer than the passages and such document ranking models suffer from the token limitation (512) of BERT. Researchers proposed ranking strategies that either truncate the documents beyond the token limit or chunk the documents into units that can fit into the BERT. In the later case, the relevance labels are either directly transferred from the original query-document pair or learned through some external model. In this paper, we conduct a detailed study of the design decisions about splitting and label transfer on retrieval effectiveness and efficiency. We find that direct transfer of relevance labels from documents to passages introduces *label noise* that strongly affects retrieval effectiveness for large training datasets. We also find that query processing times are adversely affected by fine-grained splitting schemes. As a remedy, we propose a careful passage level labelling scheme using weak supervision that delivers improved performance (3–14% in terms of nDCG score) over most of the recently proposed models for ad-hoc retrieval while maintaining manageable computational complexity on four diverse document retrieval datasets.

Keywords Ad-hoc document retrieval · BERT · Transfer learning · Distant supervision · Label transfer

1 Introduction

In web search and information retrieval, document retrieval is a standard task whether the objective is to rank the documents with respect to a query such that the most relevant documents appear on top of the list. Neural ranking approaches (Pang et al., 2016; Hui et al.,

This paper is an extended work of our previously published paper "Distant Supervision in BERT-based Ad-hoc Document Retrieval". Koustav Rudra and Avishek Anand. In Proc. CIKM 2020. 2197-2200. In this paper, we extended this work over other datasets to understand the efficiency-efficacy trade-offs of different passage granularity. This work is supported in part by the Science and Engineering Research Board, Department of Science and Technology, Government of India, under Project SRG/2022/001548. Koustav Rudra is a recipient of the DST-INSPIRE Faculty Fellowship [DST/INSPIRE/04/2021/003055] in the year 2021 under Engineering Sciences.

Extended author information available on the last page of the article

Table 1 Two sample passages for the query **"Parkinson's disease"** taken from the document marked as relevant by human annotators

Passage 1:
Eventually **Parkinson's surges forward**, leaving advancing dysfunction and death in its wake. For me, that—never mind my career, or my new marriage or my dreams of having children—is the future. The millions of other Americans **afflicted with Parkinson's**, diabetes and the other diseases have their own stories of unrealized dreams; of watching their bodies fail them, and being unable to do anything to stop it.
Passage 2:
" The bill, which could free such funding from political intervention, faces its first vote this month. And when it does, a congressional "pro-life" force is expected to attempt to reduce the discussion to the anti-abortion rhetoric that was used to justify the presidential moratorium. "

However, the first passage is relevant to the query while the second one is not

2017; McDonald et al., 2018) show better performance over term-matching based strategies (Strohman et al., 2005). Recent, pretrained language model based methods (Dai & Callan, 2019; MacAvaney et al., 2019; Yilmaz et al., 2019; Rudra & Anand, 2020; Li et al., 2020) show significant improvement because they can understand the intent of a short query through contextual interaction with the documents. However, the major issue with these language models is the limitation of input tokens. Even if some recent models (e.g., XLnet) support large input lengths, the problem of gradient vanishing is there.

The most common design choice used in retrieval and other NLP tasks when dealing with long documents is truncating or considering a limited part of the document (MacAvaney et al., 2019). However, truncation leads to undesirable information loss. Consequently, recent approaches have tried dividing the document into passages (Dai & Callan, 2019), or sentences (Yilmaz et al., 2019). Specifically, Dai and Callan (2019) considered the relevance label of all the passages of a document the same as the document level relevance label. We show examples of relevant and non-relevant passages from the same document for the query '**Parkinson's disease**' in Table 1.

Another dimension that makes machine learning for ad-hoc retrieval a challenging task is due to *label sparsity* and *label noise*. Firstly, the training labels are sparse because not all relevant query-document pairs are labelled due to large document collection sizes and exposure bias effects due to the ranking of documents (Craswell et al., 2008). Secondly, gathering user assessments for documents (relevant or not) given under specified queries using implicit feedback (White et al., 2002; Kelly & Teevan, 2003) techniques adds label noise. These limitations have resulted in two types of datasets being available to the IR community—*mostly labelled* small dataset of queries like TREC Robust data or partially labelled large dataset of queries derived from implicit feedback (i.e. query logs) like TREC-DL dataset (Craswell et al., 2019). In this context, passage label assignments as in Dai and Callan (2019), derived from document-level assessments, are yet another source of label noise.

In this paper, we follow a simple premise. If we can only consider a subset of relevant passages for training, we can significantly reduce the noise in the label assignment to passages. Towards this, we build on the recent finding by Yilmaz et al. (2019) who show that document retrieval performance can be improved by using a model trained on retrieval tasks that do not exhibit input length limitation problems. Specifically, unlike Yilmaz et al. (2019) that use an external model only during inference, we use an external passage

ranking model for QA tasks (QA-Model) to label relevant passages before training. In our examples in Table 1, the QA-Model marks the second passage as irrelevant to the query as desired. Apart from potential noise reduction in passages, such a simple labelling scheme has implications for improving training and inference efficiency as well. In sum, we ask the following research questions and summarize our key findings on the effectiveness of passage-level label transfer for document retrieval:

- What impact do large training datasets have on different labelling strategies of contextual ranking models?
- What impact do models trained on different collections have on the ranking performance?
- What is the impact of transfer-based contextual ranking models on the efficiency of *training* and *inference*?

Note that, our distant supervision based architecture was proposed in Rudra and Anand (2020). In this paper, we have performed a detailed set of experiments to understand the robustness and efficiency of our proposed approach.

1. Earlier, we tested our approach only over a small fraction of TREC-DL training and development set. In the current work, we validate the performance of our proposed approach over the recent test set of TREC-DL. The performance over the entire training set (367K queries) and variation in performance over different training sizes are also checked. Side by side, we also validate the efficacy of QA-Model on three more datasets Robust04, Core17, and ClueWeb09.
2. In our previous version, the transfer model was trained on MSMARCO passage levels and applied over TREC-DL to judge the query-passage relevance. However, both the datasets come from the same distribution and we don't have such passage level counterparts for other standard document retrieval datasets such as Robust04, Core17, and ClueWeb09. In this paper, we apply the transfer model trained on MSMARCO over different document retrieval datasets and surprisingly it performs quite well for other datasets. It gives a signal that transfer knowledge works efficiently in document reranking.
3. There is a significant dependency between the document chunking procedure and document ranking. In this paper, we have shown the influence of document chunking on the performance of different transfer models over various types of datasets. We also highlight the necessity of efficient label transfer in maintaining the robustness of the ranker models.
4. In this work, we also analyze the efficiency of different BERT based models in terms of model training and inference time. Apart from that, we also explore the role of zero shot learning in document retrieval. We observe that recent contextual models may be directly applied over a new dataset for ranking and competitive performance may be achieved based on the selection of an appropriate dataset(Sect. 5.6).
5. We have uploaded our code to Github (https://github.com/krudra/QADocRank).

Key Takeaways—We conduct extensive experiments on four TREC datasets of different collection and label properties. We show that our distantly supervised retrieval model (Qa-DocRank) is highly sample efficient as compared to document level label transfer. Distantly supervised training of BERT-based models outperforms most of the existing

baseline models. We also find that cautious cross-domain knowledge transfer from a QA passage ranking model helps balance between **retrieval performance** and **computational complexity**.

2 Related work

Ad-hoc retrieval is a classical task in information retrieval where there is precedence of classical probabilistic models based on query likelihood (Lavrenko & Croft, 2017) and BM25 (Robertson & Zaragoza, April 2009) proving hard to beat. In recent times, neural models have played a significant role in ad-hoc document retrieval and reranking. Researchers not only focused on the performance issue but also on the efficiency issues of the models. In this section, we give a brief description of different categories of ranking models.

Neural Models: Neural models bring significant changes in modeling queries and documents. They help in getting the semantic representations (Huang et al., 2013; Shen et al., 2014a, b), positional information (Hui et al., 2017, 2018; McDonald et al., 2018), local query-document interactions (Guo et al., 2016; Pang et al., 2016; Xiong et al., 2017; Nie et al., 2018a, b) or a combination of both (Mitra et al., 2017). Broadly, there are representation and interaction based models that explore query and document representations and interactions between them. Representation models present queries and documents in low dimensional latent feature space by passing them through deep neural networks and then computing the similarity between the vectors. (Huang et al., 2013) passed queries and documents through simple feed forward networks to get the semantic representations and measure the similarity score based on those vectors. (Shen et al., 2014b) used CNN instead of feed forward network to capture the local context window. CNN is also used in many other representation based neural ranking models that rely on semantic representation of queries and documents (Hu et al., 2014; Shen et al., 2014a; Qiu & Huang, 2015). Another line of work focuses on the word sequences in the queries and documents and they represent them using sequence aware models such as RNN or LSTM (Hochreiter & Schmidhuber, 1997). LSTM is used to learn the vector representations of queries and documents and finally measure the similarities between those vectors using cosine similarities (Mueller & Thyagarajan, 2016; Palangi et al., 2016). Later on, (Wan et al., 2016) proposed Bi-LSTM based representation of queries and documents, and the final similarity is measured through a neural layer.

In the representation based models, query and document feature vectors are learned independently and their interaction is deferred up to the last stage. Hence, most of the important matching signals get missed and it affects the performance of the document ranker. Hence, researchers proposed interaction based models over representation ones. (Guo et al., 2016) proposed a Deep Relevance Matching Model (DRMM) that first learns an interaction matrix between query and document using embeddings of query and document tokens. From this matrix, DRMM learns histogram based matching patterns to predict the relevance of query-document pairs. DRMM relies on hard assignment and it poses a problem for backpropagation. Hence, (Xiong et al., 2017) proposed a kernel pooling based soft matching approach to overcome this limitation. Several interaction based approaches such as Hierarchical Neural matching model (HiNT) (Fan et al., 2018a), aNMM (Yang et al., 2016), MatchPyramid (Pang et al., 2016), DeepRank (Pang et al., 2017), Position-Aware

Convolutional Recurrent Relevance (PACRR) (Hui et al., 2017) rely on interaction matrix and similarity measures like cosine similarity, dot product, etc.

A CNN is used in many interaction based models (Hui et al., 2017; Dai et al., 2018; Nie et al., 2018b; McDonald et al., 2018; Zhiwen & Grace, 2019). In general, such models use different size kernels (1D, 2D) in multiple layers of CNN and finally predict the query-document level relevance score using some MLP at the final layer. (Dai et al., 2018) extends the idea of KNRM (Xiong et al., 2017) in their Conv-KNRM model that uses CNN filters to compose n-grams from query and documents and the embeddings of such n-grams are used to learn the similarity between query-document pairs. PACRR-DRMM (McDonald et al., 2018) consider the modeling benefits of both PACRR (Hui et al., 2017) and DRMM (Guo et al., 2016) i.e., it learns document aware query token encoding in the place of histogram in DRMM.

Along with CNN, sequential neural models (RNN, GRU, LSTM) also play a key role in interaction based reranking approaches. Several approaches used LSTM based modeling of queries and documents (Fan et al., 2018b). (Wan et al., 2016) proposed Match-SRNN based on GRU to accumulate matching signals. In a similar line, (Pang et al., 2016) and DeepRank (Pang et al., 2017) fed the interaction matrix between the query and document to a GRU to learn the final feature vector. Some models such as DUET (Mitra et al., 2017) combined the benefit of both representation (distributed model) and interaction (local model) based networks to achieve better reranking performance.

Deep Contextualized Autoregressive Neural Model based Rankers: Recently introduced pre-trained language models such as ELMO (Peters et al., 2018), GPT-2 (Radford et al., 2019), SentenceBERT (Reimers & Gurevych, 2019), and BERT (Devlin et al., 2018) show promising improvement in different NLP tasks. Such models are trained on huge volumes of unlabelled data. Such contextual models, e.g., BERT, have proven to be superior in the document reranking task than the above neural models. The sentence classification task of BERT is extensively used in BERT based document retrieval techniques (Dai & Callan, 2019; Nogueira et al., 2019; Yang et al., 2019; Wu et al., 2019). Previous models addressed BERT's fixed input restriction either by sentence-wise labelling (Yilmaz et al., 2019) or passage-level labelling (Dai & Callan, 2019; Wu et al., 2020). Dai and Callan (2019). Dai and Callan (2019) split documents into passages, and obtain passage level relevance score by fine-tuning the BERT model (Doc-Labelled).

On the other hand,BERT-3S (Yilmaz et al., 2019) is a cross-domain knowledge transfer based modeling approach. It splits documents into passages and computes the score of each query-sentence pair using a secondary model trained on the MSMARCO passage and Microblogging retrieval dataset. Finally, it computes the relevance score of a query-document pair by interpolation between a sparse index based score (BM25/QL score) and the semantic score of the top three sentences learned via the transfer model. Doc-Labelled approach considers all passages of a relevant document as relevant and this introduces label noise. BERT-3S method does sentence-level knowledge transfer and takes a large inference time. This approach performs contextual modeling of documents and provides a useful upper bound on performance.

Subsequently, (MacAvaney et al., 2019) combined the power of BERT (contextual representations) and interaction based models such as Conv-KNRM (Dai et al., 2018), KNRM (Xiong et al., 2017) to improve the performance of ranking models. Recently, (Li et al., 2020) proposed an end-to-end PARADE method to overcome the limitation of independent inference of passages and predict a document's relevance by aggregating passage representations. (Hofstätter et al., 2020a, b) proposed local self-attention strategies to extract information from long text. This transformer-kernel based pooling

strategy becomes helpful to overcome the fixed length token limitation of BERT. Side by side, this kernel based strategy consumes less amount of parameters than BERT models.

So far, all the BERT-based approaches jointly modeled query-document sequences (cross-attention). This incurs huge computational costs, especially in inference time where we have to rerank around hundred to thousand documents per query. Such large transformer based models show better performance at the cost of orders of magnitude longer inference time (Hofstätter & Hanbury, 2019; MacAvaney et al., 2019). To overcome this restriction, researchers also proposed independent modeling of queries and documents. Dual encoder architecture is a strategy that encodes query and document independently of each other (Lee et al., 2019; Ahmad et al., 2019; Chang et al., 2020; Karpukhin et al., 2020; Khattab, 2020; Hofstätter et al., 2020b). This shows promising results both in terms of performance and inference cost. The BERT model of the document arm is only fine-tuned during the training phase but froze in the testing phase. Query-independent latent document representations (Luan et al., 2020) make the precise matching of terms and concepts difficult and therefore explicit term matching methods are also combined along with latent representations (Nalisnick et al., 2016; Mitra et al., 2016). Xiong et al. (2020) have recently established that the training data distribution may have a significant influence on the performance of dual encoder models under the full retrieval setting. Tilde (Zhuang & Zuccon, 2021a) and Tildev2 (Zhuang & Zuccon, 2021b) proposed a deep query and document likelihood based model instead of a query encoder to improve the ranking efficiency. The SpaDE (Choi et al., 2022) model improves the ranking efficiency by using simplified query representations and a dual document encoder containing term weighting and term expansion components. Other approaches also tried to improve the ranking efficiency by compressing document representations (Cohen et al., 2022) and removing unnecessary word representations (COLBERTER) (Hofstätter et al., 2022). Further, researchers also explored *hybrid models* where they interpolate between the scores of sparse and dense retrieval models. There exist several models in this line such as CLEAR (Gao et al., 2021b), COIL (Gao et al., 2021a), COILCR (Fan et al., 2023). (Anand et al., 2023) proposed a data augmentation based robust document retrieval framework. (Leonhardt et al., 2023b, 2022, 2023a) focused on the interpretability and efficiency of the document retrieval models.

Large Language Model based Rankers: Recent works explored large language models for document reranking task. They used pairwise ranking prompt (Qin et al., 2023) and listwise approach (Ma et al., 2023; Sun et al., 2023) for document reranking. These methods used both open source LLMs such as Flan-UL2 having 20B parameters and ChatGPT models with 175B parameters.

In this paper, we particularly focus on BERT based reranking approaches that jointly model query and document sequences using cross attention approach. Our objective is to explore the trade-off between effective transfer and efficient transfer rather than new architectural improvements as in (MacAvaney et al., 2019; Li et al., 2020; Hofstätter et al., 2020b; Choi et al., 2022; Fan et al., 2023; Leonhardt et al., 2023a). However, we believe our study could be extended to other kinds of reranking models such as dual encoder, hybrid models.

Weak supervision: Another line of work tried to train neural ranking architectures using large-scale weak or noisy labels (Dehghani et al., 2017, 2018; Zhang et al., 2020). The teacher-student paradigm (Hinton et al., 2015; Xiao et al., 2015) is also used to infer better labels from noisy labels. These labels are further used to supervise the network training (Sukhbaatar et al., 2014; Veit et al., 2017). Although similar in spirit to our approach, our work falls in the intersection of transfer learning and weak supervision in that we

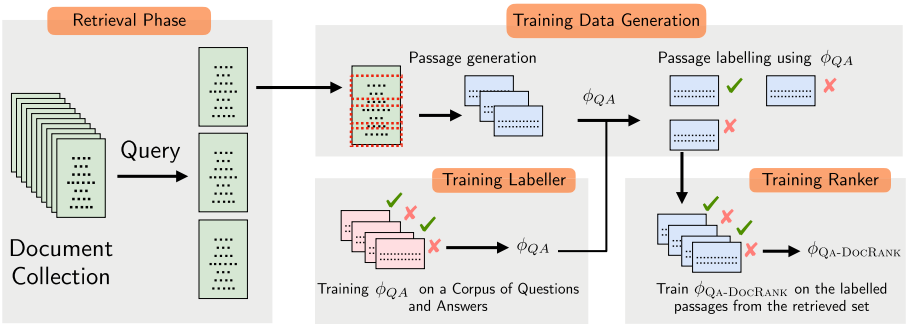


Fig. 1 Training $\phi_{\text{QA-DocRank}}$

judiciously select a subset of training instances from the original training instances (passage-query pairs) using a model trained on another task.

3 Document ranking with passage level label transfer

Typical approaches to the ad-hoc retrieval problem follow a telescoping setup Matveeva et al. (2006) and consist of two main stages. First, in a retrieval phase, a small number (for example, a thousand) of possibly relevant documents to a given query are retrieved from a large corpus of documents by a standard retrieval model such as BM25 or QLM Lavrenko and Bruce (2001). In the second stage, each of these retrieved documents is scored and re-ranked by a more computationally intensive method. Our focus is on the ranking problem in the second stage using contextual ranking models based on BERT introduced in Devlin et al. (2018).

3.1 Limitation of BERT

Nogueira and Cho (2019) were the first to show the effectiveness of contextual representations using BERT for the passage reranking task for QA on the MS MARCO dataset. They proposed QA-MODEL for the passage reranking task. However, the maximum token size for BERT is 512. This fits quite well for sentence pair tasks or question-passage tasks. Documents are longer than sentences and passages and it is difficult to fit them into the BERT model. This poses a real challenge to the query-document ranking task. In the following sections, we present our approach to handling long documents in BERT. The overall training process is outlined in Fig. 1.

3.2 Passage generation and labeling

Passage Generation. We follow the basic framework of Dai and Callan (2019) in dealing with long documents in that documents are chunked into passages of fixed size. Specifically, we follow the approach proposed by Fan et al. (2018b) for passage generation. That is, we first prepend the title to the document text. We then split each document into

passages of length 100 (white-space tokens). If the last sentence of a passage crosses the word boundary of 100, we also take the remaining part of that sentence into the current passage.

Passage Labeling. Unlike Dai and Callan (2019) that indiscriminately transfers document relevance labels to all its passages, in this paper we follow an alternate labelling scheme to selectively label passages of a relevant document. Our idea is to use an external model that is trained on a different (yet related) task of finding relevant passages given a query as a labeler for our generated passages. Towards this, we choose the model proposed by Nogueira and Cho (2019) for passage reranking task and refer to this model as QA-MODEL:

$$\phi_{QA} : (\mathbf{q}, \mathbf{p}) \rightarrow \{\texttt{relevant}, \neg\texttt{relevant}\}. \tag{1}$$

This model makes use of BERT architecture. In this part, we introduce the BERT architecture first and then show the working procedure of QA-MODEL. (Nogueira & Cho, 2019) used BERT's sentence-pair model to get the relevance score of each query-passage pair. BERT is trained on an unsupervised language modeling (LM) task on English Wikipedia and Book corpus datasets. Two different LM tasks (*Masked LM* and *Next Sentence Prediction*) are chosen to optimize the BERT model. In *Masked LM*, some words are randomly chosen and they are replaced either with [MASK] token or a random word. The goal of the *Masked LM* task is to predict the masked word correctly. Given the two sentences, the objective of the *Next Sentence Prediction* is to decide whether two sentences in a paragraph appear next to each other. BERT learns to represent sentences in the process of learning the above mentioned two tasks over a large text corpus. Thats why pre-trained BERT contains lots of parameters, e.g., $BERT_base$ contains around 110 M parameters. Pre-trained BERT can be fine-tuned for several other NLP tasks. (Nogueira & Cho, 2019) used pre-trained BERT model to get the relevance of a query-passage pair. In this process, all the parameters of BERT are also fine-tuned in an end-to-end manner for the query-passage relevance detection task. BERT can be viewed as a combination of multilayer Transformer network (Vaswani et al., 2017).

Technically, this is realized by forming an input to BERT of the form [[CLS], **q**, [SEP], **p**, [SEP]] and padding each sequence in a mini-batch to the maximum length (typically 512 tokens) in the batch. The final hidden state corresponding to the [CLS] token in the model is fed to a single layer neural network whose output represents the probability that the passage is relevant to the query **q**. Figure 2 depicts the framework.

We use the trained QA-MODEL (Eq. 1) to obtain relevance labels for query-passage pairs derived from the initial retrieved set of documents. Specifically, we label a passage $\mathbf{p} \in \mathbf{d}$ of a **relevant document** as relevant if $\phi_{QA}(\mathbf{q}, \mathbf{p}) = \texttt{relevant}$.

3.3 Proposed document ranking model

We now detail our training and inference procedure on the newly labelled query-passage pairs.

Training. After obtaining the passage labels, we now finetune another BERT model on these query-passage pairs following the approach proposed by Nogueira and Cho (2019). That, using the pre-trained BERT base uncased model (Devlin et al., 2018), we fine-tune the model end to end using the passage level relevance signal. We randomly sample the same number of non-relevant query-passage pairs as relevant ones because the number of non-relevant pairs is way larger than the relevant ones. In some sense, our setup resembles

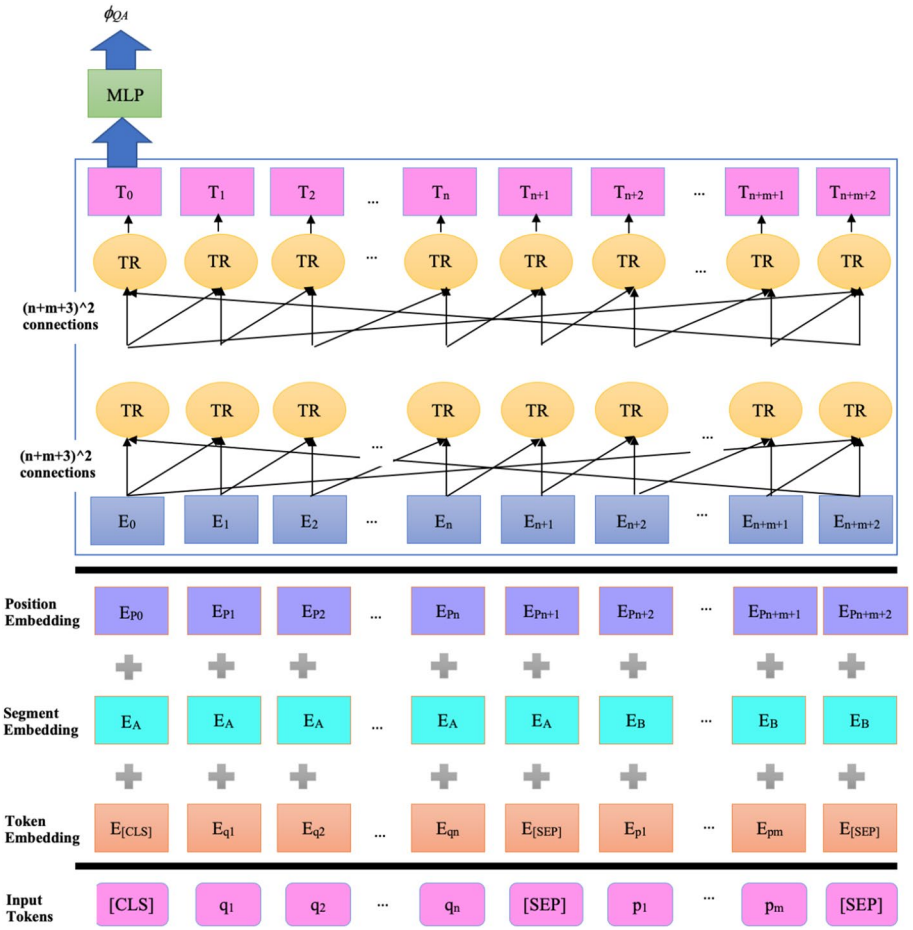


Fig. 2 QA-MODEL(ϕ_{QA}) model architecture. The BERT model takes a query $q = q_1, q_2, \ldots, q_n$ and passage $p = p_1, p_2, \ldots, p_m$ of length n and m respectively. This is passed through several transformer (TR) layers and finally, the representation of the *CLS* token is passed through a feed forward layer to predict the relevance score of the query-passage pair

a teacher-student training paradigm where our QA-MODEL works as a *teacher* to determine relevant passages in a document to assist the *student*, here QA-DOCRANK, in the training process.

Inference. Finally, the trained QA-DOCRANK model is applied over the test set to predict the relevance labels of query-passage pairs and these scores are *aggregated* to get the final score of the corresponding query-document pair. *Note that, this* QA-MODEL *based passage-level judgements are only applied to training and validation sets. For the test set, documents are ranked based on different aggregation methods applied over passage level scores.* Towards this, we adopt four different aggregation strategies as proposed by Dai and Callan (2019). Apart from that, we use two position aware passage score aggregation strategies. The aggregation functions are given below:

1. **FirstP:** Score of the first passage

2. **MaxP:** Score of the best passage
3. **SumP:** Sum of all passage scores
4. **AvgP:** Average of all passage scores
5. **DecaySumP:** Instead of giving equal weight to all the passages when summing up the scores, the passage weights are multiplied by the inverse of their position in the document.

$$\psi(\mathbf{q}, \mathbf{d}) = \sum_{\substack{i=1 \\ \mathbf{p_i} \in \mathbf{d}}}^{m} \left(\phi_{\text{Qa-DocRank}}(\mathbf{q}, \mathbf{p_i}) * \frac{1}{i} \right) \tag{2}$$

6. **DecayAvgP:** Similar to DecaySumP, but the total score is normalized by the number of passages.

$$\psi(\mathbf{q}, \mathbf{d}) = \frac{\sum_{\substack{i=1 \\ \mathbf{p_i} \in \mathbf{d}}}^{m} (\phi_{\text{QA-DocRank}}(\mathbf{q}, \mathbf{p_i}) * \frac{1}{i})}{m} \tag{3}$$

$$\psi(\mathbf{q}, \mathbf{d}) = \underset{\mathbf{p} \in \mathbf{d}}{\texttt{Aggr}}(\phi_{\text{QA-DocRank}}(\mathbf{q}, \mathbf{p})) \tag{4}$$

In our experiments we refer to the ranking based on scoring after aggregation (Eq. 4) as QA-DocRank.

4 Experimental evaluation

In this section, we experimentally evaluate the efficiency of our proposed approach QA-DocRank. We begin by describing our baselines, experimental setup, and evaluation procedure in this section.

4.1 Baselines and competitors

The first-stage retrieval model, the query likelihood model, is also considered as a ranking baseline (Lavrenko & Bruce, 2001). Our competitors are the following Non-contextual and contextual rankers.

Non Contextual Neural Models. We then compare against non-contextual neural neural ranking models PacrrDrmm (McDonald et al., 2018) that combines the modelling of Pacrr (Hui et al., 2017) and aggregation of Drmm (Guo et al., 2016). We also tried other non-contextual neural models like MatchPyramid (Pang et al., 2016), Drmm, and Pacrr but PacrrDrmm consistently outperforms them. Hence, we use PacrrDrmm as a representative non-contextual neural model and skip other results.

Contextual Ranking Models. We consider following contextual ranking models.

1. **Doc-Labelled** (Dai & Callan, 2019): Baseline from Dai and Callan (2019) where relevance labels are transferred from the document level to passage level.
2. **BERT-CLS** (Devlin et al., 2018): The BERT model is fine-tuned with document level supervision. This is truncation based approach MacAvaney et al. (2019) where content beyond 512 tokens are dropped.

3. **BERT-3 S** (Yilmaz et al., 2019): Cross-domain knowledge transfer based approach. A model trained on MSMARCO and TREC microblog data is used to obtain query-sentence scores in a document. Finally, it aggregates document level score and top-k sentence scores (evaluated by a transfer model to compute the final document relevance score with respect to a query.

We exclude other recently proposed approaches like CEDR (MacAvaney et al., 2019) that focus on architectural engineering using BERT. Such methods are complementary to our study and can of course benefit from our analysis.

Training details: We follow the consistent and standard experimental design to train and validate the models. We use a fixed number of iterations using pairwise max-margin loss to train pairwise neural models. The best model is chosen based on the MAP score computed over the validation set. On the other hand, we train BERT BASED RANKERS by putting a classification layer on top of the BERT model and optimizing it using binary cross-entropy loss. Finally, we aggregate passage scores to compute the document level score. For fair comparisons, we choose the hyper-parameters commonly used in the earlier works, i.e., the sequence length of 512, the learning rate of $1e-5$, and the batch size of 16. The learning rate is chosen based on the performance of the validation set. We tried it over $1e-5$, $2e-5$, and $3e-5$ but did not observe any significant variations. We chose a learning rate of $1e-5$. The results are dependent on the version of the Pytorch and transformer models. In this paper, we used Pytorch and transformer versions 1.7.1 and 4.10.2 respectively.

Metrics: We measure the effectiveness of the ranking baselines using three standard metrics—MAP, P@20, nDCG@20 (Järvelin & Kekäläinen, 2002).

Avoiding Data Leakage. Note that the *teacher*, i.e., the QA-MODEL, is trained on the MSMARCO passage dataset and the queries are the same (highly overlapping) as TREC-DL dataset. Hence, we do not apply QA-MODEL over any of the test sets to avoid data leakage. In particular, this will lead to potential data leak issues for TREC-DL dataset. However, QA-DOCRANK does not have this issue because training, validation, and test query sets for TREC-DL are disjoint.

4.2 Datasets

We consider following four diverse TREC datasets with varying degrees of label properties.

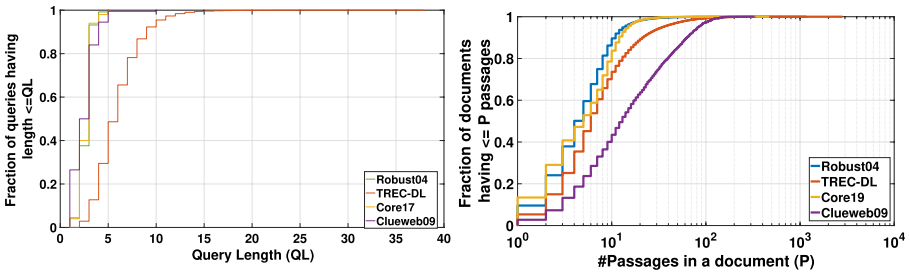
1. **Robust04**: We have 249 queries with their description and narratives. Along with queries, we also have a 528K document collection. We retrieve the top 1000 documents for each query using QLM (Strohman et al., 2005).
2. **TREC-DL**: The TREC-DL document ranking dataset is divided into training, development, and test set. The training set contains around 367K queries and the test set contains 200 queries. We randomly select 2000 queries to build the training set. For each of these queries, the top 100 documents are retrieved using QLM.[1]
3. **Core17**: The CORE17 contains 50 queries with sub-topics and descriptions. Queries are accompanied by a 1.8 M document collection. We retrieve the top 1000 documents for each query using QLM.

[1] https://microsoft.github.io/msmarco/TREC-Deep-Learning-2019.html.

Table 2 Statistics about the datasets

	Average query length	Mean passages
Robust04	2.65	5.87
TREC-DL	5.95	10.87
Core17	2.64	6.26
ClueWeb09	2.47	23.96

Average query length and number of passages in documents



(a) **NEW: Query Length** (b) **Document Passages**

Fig. 3 CDF of query length and passages of a document. *x − axis* is in log-scale

4. **ClueWeb09**: We consider the ClueWeb09 dataset shared by Dai and Callan (2019). The dataset contains 200 queries distributed uniformly in five folds and the top 100 documents for each query are retrieved using QLM.

The dataset details are available at https://github.com/krudra/IRJ_distant_supervision_adhoc_ranking. Standard ad-hoc document retrieval datasets reveal quite a different trend than the TREC-DL that is curated from query logs. The queries in TREC-DL are well specified. The documents in TREC-DL and ClueWeb09 are longer. On the other hand, documents in news corpus such as Robust04 and Core17 are relatively short. Table 2 provides specifications about different datasets. Figure 3 shows the distribution of query length and the number of passages in a document.

We conduct our experiments on a Nvidia 32GB V100 machine using PyTorch version 1.5.0 and evaluate baselines and our proposed models on four datasets. We have used BERT from the transformer repository (2.10.0) of Huggingface.[2] We have used a deterministic version of BERT and taken a fixed seed 123 to remove the external influencing factors and make the result consistent across models. For Robust04, Core17, and ClueWeb09 we conduct 5 fold cross-validation to minimize overfitting due to the limited number of queries in the collection. Topics are randomly split into 5 folds and the model parameters are tuned on 4-of-5 folds. The retrieval performance is evaluated on the final fold in each case using the optimal parameters. This process is repeated five times, once for each fold. For TREC-DL, we have 200 queries for the test set. *For* ClueWeb09, *we directly take the folds*

[2] https://huggingface.co/transformers/.

Table 3 Retrieval performance of baselines, and Qa-DocRank method

Method	Robust04			TREC-DL		
	MAP	nDCG20	P@20	MAP	nDCG20	P@20
QL Model	0.240	0.403*	0.347	0.237	0.487*	0.495
PacrrDrmm	0.263	0.445*	0.374	0.241	0.517*	0.508
Doc-Labelled	0.249	0.423*	0.363	0.258	0.557*	0.568
BERT-CLS	0.276	**0.474**	**0.414**	0.246	0.568*	0.579
BERT-3 S	**0.289**	**0.476**	**0.409**	**0.267**	**0.595**	0.586
Qa-DocRank	**0.294**	**0.471**	**0.406**	**0.269**	**0.603**	**0.602**
Method	Core17			ClueWeb09		
	MAP	nDCG20	P@20	MAP	nDCG20	P@20
QL Model	0.203	0.395*	0.474	0.165	0.277*	0.331
PacrrDrmm	0.215	0.418	0.497	0.169	0.285*	0.336
Doc-Labelled	0.239	0.445	0.514	0.177	0.309*	0.355
BERT-CLS	0.242	0.449*	0.549	0.183	0.313*	0.354
BERT-3 S	**0.258**	**0.476**	**0.571**	0.184	0.314*	0.366
Qa-DocRank	0.239	0.458	0.539	**0.193**	**0.341**	**0.383**

We report the result of the best aggregation method for Qa-DocRank and Doc-Labelled. For Doc-Labelled MaxP and DecaySumP show the best result for (Robust04, Core17, TREC-DL), and ClueWeb09 respectively. For Qa-DocRank MaxP and DecaySumP show the best result for (Robust04, TREC-DL), and (Core17, ClueWeb09) respectively. $*$ implies Qa-DocRank is statistically significantly better at 95% significance level, than the corresponding baseline method

Best values are marked in bold

from prior study (Dai & Callan, 2019). TREC-DL *is also evaluated over a standard test set. The folds for* Robust04 *and* Core17 *will be shared for reproducibility.*

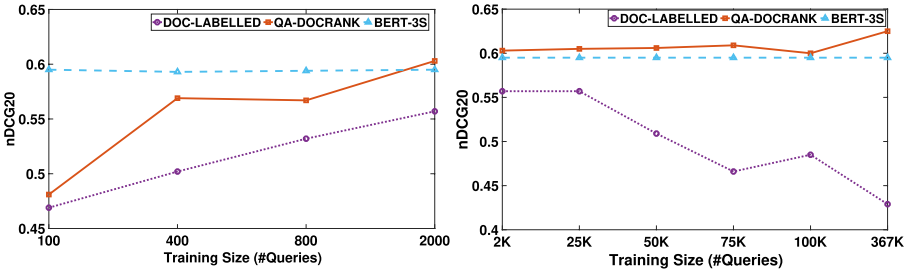
4.3 Results

We elaborate on the performance of Qa-DocRank in this section.

How effective is the passage level transfer for document retrieval?

We start with comparing the ranking performance of Qa-DocRank against other baselines in Table 3.

First, in line with previous works, we observe that the contextual rankers outperform other non-contextual rankers convincingly for most of the datasets. **Among the contextual models, BERT-3S and our approach outperform Doc-Labelled**. The improvements of Qa-DocRank over Doc-Labelled (nDCG20) are *statistically significant* with p-scores 0.002, 0.014 for Robust04 and TREC-DL as per paired t-test ($\alpha = 0.05$) with Bonferroni correction (Gallagher, 2019). BERT-3S obtains statistically significant improvement over Doc-Labelled for Robust04, TREC-DL, and Core17. However, the improvements of BERT-3S over Qa-DocRank are *not statistically significant for all the datasets.* As we show later though the ranking performance obtained by BERT-3S is competitive with our approach, their inference phase is computationally heavy (sometimes infeasible for large web collections) when evaluating long documents (Sect. 5).



(a) Small Training sets (b) Larger Training Sets

Fig. 4 Effect of training dataset size on ranking performance. **a** Small training set, **b** larger training sets

Threats to Validity. We note that there are differences between the ranking performance of baselines as measured by us and in the original paper of the authors and that can be attributed to experimental design choices for better comparison. The difference in Doc-Labelled is mainly due to the differences in the passage chunking setup. The folds are different in the case of Robust04. The implementation setup is different than the original paper. We use the full token length (512) of BERT instead of 256 token length and implement the code in Pytorch using a deterministic version of BERT.[3] The performance of the model depends on the version of the BERT model. Hence, we set a specific seed value to make the results reproducible. We train the entire setup using Pytorch version 1.7.1 and Transformer version 4.10.2. We attribute the performance difference of BERT-3S occurred for the following three design choices: (1) they selected BERT-LARGE as their fine-tuning model. However, we choose BERT-BASE to make a fair performance comparison among the three BERT based models. (2) they used MSMARCO+MICROBLOG based transfer model to learn the relevance of query-sentence pairs. We only select MSMARCO as the transfer model to keep consistency between Qa-DocRank and BERT-3S. (3) they used BM25+RM3 instead of QL to retrieve the initial document set. The objective of this paper is to compare the effectiveness-efficiency trade-off of BERT based reranking models and their variation based on the granularity of documents (passage/sentence), label transfer, etc. Hence, we try to keep external influencing factors almost the same across different models. However, we believe that considering BERT-LARGE and MSMARCO+MICROBLOG based transfer models will not change the trend significantly.

5 Analysis

We present our effectiveness-efficiency related findings based on the research questions formulated in Sect. 1.

[3] https://huggingface.co/docs/transformers/model_doc/bert.

Table 4 Retrieval performance (nDCG20 score) of QA-DocRank over different passage score aggregation set-ups

	FirstP	MaxP	SumP	AvgP	DecaySumP	DecayAvgP
Robust04	0.420	**0.471**	0.418	0.365	0.444	0.298
TREC-DL	0.581	**0.603**	0.520	0.525	0.568	0.449
Core17	0.391	0.408	0.437	0.365	**0.458**	0.273
ClueWeb09	0.313	0.308	0.313	0.285	**0.341**	0.218

Best values are marked in bold

5.1 How do larger training datasets impact the training of contextual ranking models?

We now measure the effectiveness of ranking models for different training data sizes—measured by several queries. We consider three contextual models and the best performing pairwise neural model (PACRRDRMM).

First, we look at the performance of rankers in a small data regime reported in Fig. 4a. We randomly select 100, 400, 800 queries from the training set of TREC-DL to train the models. We observe that for smaller datasets, QA-DocRank suffers due to its high selectivity that results in a small number of training instances (see nDCG@20 for 100 queries). However, the performance monotonically increases with the number of queries and is already equivalent to BERT-3S for 1000 queries. *Note that, the performance of* BERT-3S *is constant because the interpolation parameter α is learned on the validation set, and the training set has no impact on it.*

Next, we analyze the effect of using larger datasets on passage-level BERT models. Figure 4b presents the ranking performance with increasing training data from 2K queries to 100K queries from the TREC-DL dataset. All results are reported over the same test set of 2K queries. **We find that the performance of Doc-Labelled is significantly affected with increasing number of training queries.** We attribute this to the longer documents in TREC-DL. Specifically, the average document length of TREC-DL is almost twice that of Robust04 and Core17. *This means that longer relevant Web documents tend to contain more irrelevant passages and document to passage label transfer is susceptible to higher label noise.* This, along with results from the passage generation experiment, clearly establishes the negative impact of label noise introduced by document to passage label transfer. On the other hand, QA-DocRank is effectively able to filter out noise due to its judicious label selection strategy and is unaffected by increasing training size.

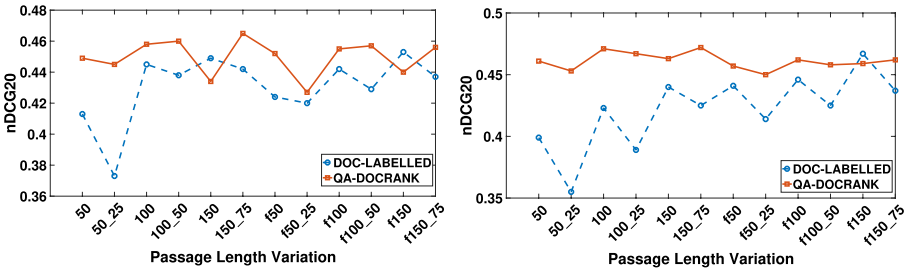
5.2 What is the role of passage aggregation strategy on overall performance?

It is evident from Table 3 that the performance of the models over different datasets is very sensitive to aggregation strategies. For Robust04 and TREC-DL, the maximum score of a passage turns out to be a good measure for the entire document. On the other hand, position decay weighted summation performs better than other aggregation strategies for Core17 and ClueWeb09. Table 4 shows the performance of QA-DocRank over different datasets for different aggregation strategies. The results suggest that the

Table 5 Retrieval performance (nDCG20 score) of QA-DOCRANK and DOC-LABELLED over two different passage generation set-ups

	QA-DOCRANK		DOC-LABELLED	
	*P*_100	*P*_150_75	*P*_100	*P*_150_75
Robust04	**0.471**	0.462	0.423	0.436
TREC-DL	**0.627**	0.555	0.429	0.434
Core17	**0.458**	**0.456**	0.445	0.437
ClueWeb09	**0.341**	0.316	0.309	0.289

Best values are marked in bold



(a) **Core17** (b) **Robust04**

Fig. 5 Effect of passage granularities on the performance of **a** CORE17, **b** ROBUST04

ranking strategy should dynamically determine the aggregation strategy and end-to-end setup shows promising results due to the ability of this data specific adaptation (Li et al., 2020).

5.3 How robust are the models to passage generation?

In the previous part, we observe that the performance of DOC-LABELLED drops with the increase in training query size. Here, our objective is to check the variation in the performance with the passage sizes. DOC-LABELLED splits documents into passages of length 150 words with an overlap of 75 words between consecutive passages and consider 30 passages (first,last, and random 28). However, we generate non-overlapping passages of length 100 following the approach proposed in Fan et al. (2018b) since we consider it a better way for passage splitting. To verify the superiority of our proposed approach, we also apply QA-DOCRANK and DOC-LABELLED to the passages generated using the approach mentioned in Dai and Callan (2019).[4]

Table 5 shows the nDCG scores of both methods under different passage generation setups over all four datasets. Note that, for DOC-LABELLED in TREC-DL we have used the entire training query set (367K) instead of 2000 queries; hence, the nDCG score in Table 5 under P_{100} column (0.429) is different from the value reported in Table 3 (0.541). QA-DOCRANK is robust to passage generation setup. However, DOC-LABELLED is very much

[4] We are grateful to the authors for sharing the data.

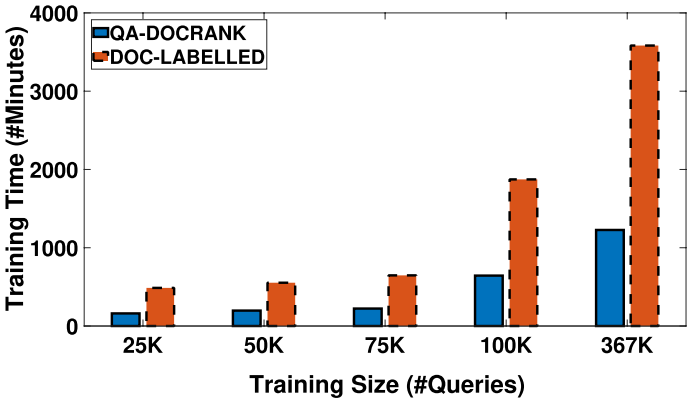


Fig. 6 Effect of training dataset size on training time for QA-DOCRANK and DOC-LABELLED

sensitive to the passages. We find that document to passage level label transfer introduces significant noise in the training phase.

Variation in performance based on passage size: Both DOC-LABELLED and QA-DOCRANK are dependent on the input size limit enforced by BERT models. In this experiment, we want to first evaluate if different granularities of partitioning documents into passages affect ranking performance. We study the ranking performance based on varying— *passage length*, *overlap between two consecutive passages*, and *number of passages*. This entails four scenarios—(1) X : documents are split into passages of length X and all the passages are considered, (2) X_Y : similar to case (1) but there is an overlap of Y words between two consecutive passages, (3) fX : passages are of length X. Following the approach in Dai and Callan (2019), the first, last, and randomly 28 other passages are chosen instead of all the passages, and (4) fX_Y : similar to case (3) with overlap of Y words. Figure 5 reports the performance under the above mentioned four scenarios for different X and Y. The major takeaway from this experiment is that DOC-LABELLED is sensitive to passage generation while QA-DOCRANK is robust. We observe that the performance of DOC-LABELLED improves considerably when a fixed number of passages is considered. This is the first evidence that the direct assignment of labels to all constituent passages is wasteful and leads to label noise. A fixed number of passages implicitly controls label noise.

5.4 How efficient is it to *train* transfer-based BERT retrieval models?

In this experiment, we measure the wall-clock times (in minutes) for training BERT-based models. Firstly, we note that BERT-3S does not involve any fine-tuning in the training phase and is not included in the experiment. On the other hand, QA-DOCRANK, and DOC-LABELLED require fine-tuning over the task specific dataset. For QA-DOCRANK, we also consider the time taken by QA-MODEL to find relevant training passages. However, the training/fine-tuning time of QA-MODEL is not considered to follow the same protocol as BERT-3S. As expected, we observe from Fig. 6 that the **training time of Doc-Labelled is** 2.5–3 **times higher than Qa-DocRank**. This is a direct impact of selective labelling of passages in QA-DOCRANK that results in far fewer training instances in comparison to DOC-LABELLED that indiscriminately transfers labels to all the passages. Specifically, the training size of QA-DOCRANK is around 7–8% of

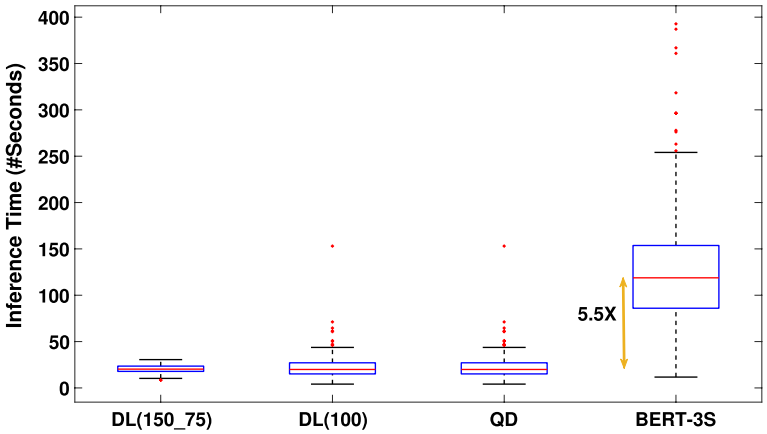


Fig. 7 Comparison of inference times for BERT based models (Doc-Labelled(*DL*(150_75)), Doc-Labelled(*DL*(100)), Qa-DocRank, BERT-3S). *DL*(150_75) consists of a passage length of 150 words with an overlap of 75 words between consecutive passages. Median inference time for BERT-3S is 5.5 times higher than Qa-DocRank and Doc-Labelled

Doc-Labelled approach. This further strengthens the hypothesis that selective passage transfer not only helps in retrieval effectiveness but has a direct impact on training efficiency by being sample efficient.

5.5 How efficient is *inference* for transfer-based BERT retrieval models?

Figure 7 reports the variation in wall-clock times for query processing or inference of Qa-DocRank, Doc-Labelled, and BERT-3S over the 200 test queries of TREC-DL. We only measure the inference time of the model and do not include the preprocessing times such as tokenization of queries and documents, batching, etc. For Doc-Labelled, we test both the variations of passage chunking. In *DL*(150_75), passages are created as proposed in Dai and Callan (2019) i.e., each passage is 150 words long with an overlap of 75 words between consecutive passages and 30 passages are selected. On the other hand, *DL*(100), passages are 100 words long and there is no overlap between passages. As expected, we observe that the **average query processing time for BERT-3 S is much larger than Qa-DocRank due to the sentence-level scoring**—around 5.5 times higher than Qa-DocRank and Doc-Labelled approach. There is no significant difference in mean and median between Doc-Labelled and Qa-DocRank. The average time taken by *DL*(150_75) and *DL*(100) is almost the same. However, the standard deviation of *DL*(100) is three times higher than *DL*(150_75) because the later version is restricted to 30 passages per document. BERT-3S also has a large standard deviation 4× higher than Qa-DocRank with some queries with long result documents taking 400 seconds to process. These results reflect on a yet to be resolved open question in terms of efficient inference of BERT-based models. *We did not consider any parallel optimization techniques in the score computation process for any of the methods. We believe that each method will get a similar kind of improvement in the running time (i.e., inferring scores of passages/sentences).*

Table 6 Cross-Domain Retrieval performance (nDCG20 score) of fine-tuned QA-DocRank with the aggregation strategy same as training dataset

	Test			
Train	Robust04	TREC-DL	Core17	ClueWeb09
Robust04	**0.471**	0.519(0.528)	0.435	0.245 (0.301)
TREC-DL	**0.459(0.471)**	**0.627**	0.410(0.443)	0.291 (0.312)
Core17	0.454(0.464)	0.533	**0.458**	0.314
ClueWeb09	0.443	0.599	0.424	**0.341**

Values in the bracket present the best score achieved through some aggregation strategy that is different from the training set

Best values are marked in bold

5.6 How well does a model fine-tuned on one document transfer to another collection?

In the last section, we check the efficacy of *transfer learning* from QA-Model to document ranking (QA-DocRank (passage-level), BERT-3S(sentence-level)). Interestingly, it performs quite well even without any fine-tuning in some cases (BERT-3S). Here, we verify the effectiveness of document-specific fine-tuned models. We fine-tune our QA-DocRank model over a dataset and test it over the other ones. It is interesting to note that in some cases cross-document testing gives almost similar ranking to the models trained on the same document. Table 6 shows the results on cross domain inference. In cross-domain setup, we don't have any clues about the test data set; hence, we have to rely on the aggregation strategy that works best for the training dataset. Table 6 reports both results i.e., results achieved on the test data based on the aggregation strategy of the training data and the best score achieved through another aggregation strategy(in brackets). Specifically QA-DocRank shows comparable performance to the in-domain testing but this depends on the training dataset. For example, Core17 achieves the best ranking for Robust04 and ClueWeb09 whereas ClueWeb09 performs well for TREC-DL. Robust04 and Core17 both are news corpus whereas ClueWeb09 and TREC-DL are web corpus; hence, these groups follow different information distribution. In general, a news corpus contains information mostly in the first couple of passages. TREC-DL contains a significantly larger number of queries than other datasets; hence, training and generalization of models are quite easy for this case. TREC-DL shows consistent performance over other datasets. However, the performance of other models is also not significantly worse than the best performer. It indicates that **careful selection of training documents and passages might help in the direct application of document specific fine-tuned models over new collection.**

6 Conclusion

In this paper, we illustrate the shortcomings of two transfer learning based modeling approaches Doc-Labelled and BERT-3S. The former suffers due to label noise that degenerates its performance beyond a certain point while the inference time restricts the utility of the later one. We have combined the positive aspects of both models and proposed an

approach that optimizes both **retrieval performance** and **computational complexity**. We also show the robustness of this model towards document splitting schemes and its applicability in cross-domain document ranking i.e., a model trained on one document set may be directly applied to another set.

Throughout this paper, we assume that passages are disjoint and treat them as separate entities during relevance prediction. In the future, it will be interesting to capture the interaction among different passages to check its impact on retrieval performance. Very few passages in a document are ultimately relevant to a given query. Hence, it will be interesting to find such a denoised version of the document before the retrieval and ranking task. This will also be helpful to bring interpretability into the framework. We also explore large language models to improve query rewriting and zero-shot model performance.

References

Ahmad, A., Constant, N., Yang, Y., & Cer, D. (2019). Reqa: An evaluation for end-to-end answer retrieval models. arXiv:1907.04780

Anand, A., Leonhardt, J., Singh, J., Rudra, K., & Anand, A. (2023). Data augmentation for sample efficient and robust document ranking.

Chang, W.-C., Yu, F. X., Chang, Y.-W., Yang, Y., & Kumar, S. (2020). Pre-training tasks for embedding-based large-scale retrieval. arXiv:2002.03932

Choi, E., Lee, S., Choi, M., Ko, H., Song, Y.-I., & Lee, J. (2022). Spade: Improving sparse representations using a dual document encoder for first-stage retrieval. In *Proceedings of the 31st ACM international conference on information and knowledge management, CIKM '22* (pp. 272-282).

Cohen, N., Portnoy, A., Fetahu, B., & Ingber, A. (2022). SDR: Efficient neural re-ranking using succinct document representation. In *Proceedings of the 60th annual meeting of the association for computational linguistics (Volume 1: Long Papers)*, (pp. 6624–6637). Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.acl-long.457, https://aclanthology.org/2022.acl-long.457

Craswell, N., Zoeter, O., Taylor, M., Ramsey, B. (2008). An experimental comparison of click position-bias models. In *Proceedings of the 2008 international conference on web search and data mining* (pp. 87–94).

Craswell, N., Mitra, B., Yilmaz, E., & Campos, D. (2019). TREC-2019-deep-learning. https://microsoft.github.io/TREC-2019-Deep-Learning/

Dai, Z., & Callan, J. (2019) Deeper text understanding for ir with contextual neural language modeling. In *ACM SIGIR'19* (pp. 985–988).

Dai, Z., Xiong, C., Callan, J., Liu, Z. (2018). Convolutional neural networks for soft-matching n-grams in ad-hoc search. In *Proceedings of the 11th ACM international conference on web search and data mining, WSDM '18* (pp. 126–134). ACM. ISBN 978-1-4503-5581-0. 10.1145/3159652.3159659, http://doi.acm.org/10.1145/3159652.3159659

Dehghani, M., Zamani, H., Severyn, A., Kamps, J., & Bruce Croft, W. (2017). Neural ranking models with weak supervision. In *SIGIR '17* (pp. 65–74). ACM. ISBN 978-1-4503-5022-8. 10.1145/3077136.3080832. http://doi.acm.org/10.1145/3077136.3080832

Dehghani, M., Mehrjou, A., Gouws, S., Kamps, J., & Schölkopf, B. (2018). Fidelity-weighted learning. In *ICLR '18*. https://openreview.net/forum?id=B1X0mzZCW

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of deep bidirectional transformers for language understanding. CoRR, abs/1810.04805, http://arxiv.org/abs/1810.04805

Fan, Y., Guo, J., Lan, Y., Xu, J., Zhai, C., Cheng, X. (2018a) Modeling diverse relevance patterns in ad-hoc retrieval. In *The 41st international ACM SIGIR conference on research and development in information retrieval, SIGIR '18* (pp. 375–384). ACM. ISBN 978-1-4503-5657-2. 10.1145/3209978.3209980, http://doi.acm.org/10.1145/3209978.3209980

Fan, Y., Guo, J., Lan, Y., Xu, Jun, Z., Chengxiang, & Cheng, X. (2018b). Modeling diverse relevance patterns in ad-hoc retrieval. In *ACM SIGIR'18* (pp. 375–384).

Fan, Z., Gao, L., Jha, R., & Callan, J. (2023). Coilcr: Efficient semantic matching in contextualized exact match retrieval. In J. Kamps, L. Goeuriot, F. Crestani, M. Maistro, H. Joho, B. Davis, C. Gurrin, U. Kruschwitz, & A. Caputo (Eds.), *Advances in information retrieval* (pp. 298–312). Cham: Springer.

Gallagher, L. (2019). Pairwise t-test on TREC run files. https://github.com/lgrz/pairwise-ttest/

Gao, L., Dai, Z., & Callan, J. (2021a). COIL: Revisit exact lexical match in information retrieval with contextualized inverted list. In *Proceedings of the 2021 conference of the north American chapter of the association for computational linguistics: Human language technologies* (pp. 3030–3042). Association for Computational Linguistics.

Gao, L., Dai, Z., Chen, T., Fan, Z., Van Durme, B., & Callan, J. (2021b). Complement lexical retrieval model with semantic residual embeddings. In Hiemstra, D., Moens, M.-F., Mothe, J., Perego, R., Potthast, M., & Sebastiani, F. (Eds.), *Advances in information retrieval* (pp. 146–160). Springer.

Guo, J., Fan, Y., Ai, Q., & Bruce Croft, W. (2016). A deep relevance matching model for ad-hoc retrieval. In *CIKM'16* (pp. 55–64). ACM. ISBN 978-1-4503-4073-1, https://doi.org/10.1145/2983323.2983769, http://doi.acm.org/10.1145/2983323.2983769

Hinton, G., Vinyals, O., Dean, J. (2015). Distilling the knowledge in a neural network. arXiv:1503.02531

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation, 9*(8), 1735–1780.

Hofstätter, S., Hanbury, A. (2019). Let's measure run time! extending the ir replicability infrastructure to include performance aspects. arXiv:1907.04614

Hofstätter, S., Khattab, O., Althammer, S., Sertkan, M., & Hanbury, A. (2022). Introducing neural bag of whole-words with colberter: Contextualized late interactions using enhanced reduction. In *Proceedings of the 31st ACM international conference on information and knowledge management, CIKM '22* (pp. 737–747).

Hofstätter, S., Zamani, H., Mitra, B., Craswell, N., & Hanbury, A. (2020a). Local self-attention over long text for efficient document retrieval. arXiv:2005.04908

Hofstätter, S., Zlabinger, M., & Hanbury, A. (2020b). Interpretable and time-budget-constrained contextualization for re-ranking. arXiv:2002.01854

Hu, B., Lu, Z., Li, H., & Chen, Q. (2014). Convolutional neural network architectures for matching natural language sentences. In *Proceedings of the 27th international conference on neural information processing systems-volume 2, NIPS'14* (pp. 2042–2050).

Huang, P.-S., He, X., Gao, J., Deng, L., Acero, A., & Heck, L. (2013). Learning deep structured semantic models for web search using clickthrough data. In *CIKM '13* (pp. 2333–2338). ACM. ISBN: 978-1-4503-2263-8, https://doi.org/10.1145/2505515.2505665, http://doi.acm.org/10.1145/2505515.2505665

Hui, K., Yates, A., Berberich, K., & de Melo, G. (2017). PACRR: A position-aware neural ir model for relevance matching. In *EMNLP '17* (pp. 1049–1058). https://www.aclweb.org/anthology/D17-1110

Hui, K., Yates, A., Berberich, K., & de Melo, G. (2018). Co-PACRR: A context-aware neural ir model for ad-hoc retrieval. In *WSDM '18* (pp. 279–287). ACM. ISBN:978-1-4503-5581-0, https://doi.org/10.1145/3159652.3159689, http://doi.acm.org/10.1145/3159652.3159689

Järvelin, K., & Kekäläinen, J. (2002). Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Management Information Systems, 20*(4), 422–446.

Karpukhin, V., Oğuz, B., Min, S., Wu, L., Edunov, S., Chen, D., & Yih, W.-t. (2020). Dense passage retrieval for open-domain question answering. arXiv:2004.04906

Kelly, D., & Teevan, J. (2003). Implicit feedback for inferring user preference: A bibliography. *Acm Sigir Forum* (Vol. 37, pp. 18–28). ACM.

Khattab, O. (2020). In Zaharia, M. (Eds.), *Efficient and effective passage search via contextualized late interaction over bert: Colbert*.

Lavrenko, V., & Bruce Croft, W. (2001). Relevance based language models. In *Proceedings of the 24th annual international ACM SIGIR conference on research and development in information retrieval, SIGIR '01*, (pp. 120–127). ACM. ISBN 1-58113-331-6, https://doi.org/10.1145/383952.383972, http://doi.acm.org/10.1145/383952.383972

Lavrenko, V. & Croft, W. B. (2017). Relevance-based language models. In *ACM SIGIR forum* (Vol. 51, pp. 260–267). ACM.

Lee, K., Chang, M.-W., & Toutanova, K. (2019). Latent retrieval for weakly supervised open domain question answering. arXiv:1906.00300

Leonhardt, J., Müller, H., Rudra, K., Khosla, M., Anand, A., & Anand, A. (2023). Efficient neural ranking using forward indexes and lightweight encoders. *ACM Transactions on Management Information Systems*. https://doi.org/10.1145/3631939

Leonhardt, J., Rudra, K., Khosla, M., Anand, A., & Anand, A. (2022). Efficient neural ranking using forward indexes. In *Proceedings of the ACM web conference 2022, WWW '22* (pp. 266–276).

Leonhardt, J., Rudra, K., & Anand, A. (2023). Extractive explanations for interpretable text ranking. *ACM Transactions on Management Information Systems*. https://doi.org/10.1145/3576924

Li, C., Yates, A., MacAvaney, S., He, B., & Sun, Y. (2020). Parade: Passage representation aggregation for document reranking. arXiv:2008.09093

Luan, Y., Eisenstein, J., Toutanova, K., & Collins, M. (2020). Sparse, dense, and attentional representations for text retrieval. arXiv:2005.00181

Ma, X., Zhang, X., Pradeep, R., & Lin, J. (2023). Zero-shot listwise document reranking with a large language model. arXiv:2305.02156

MacAvaney, S., Yates, A., Cohan, A., & Goharian, N. (2019). Contextualized word representations for document re-ranking. arXiv:1904.07094

Matveeva, I., Burges, C., Burkard, T., Laucius, A., & Wong, L. (2006). High accuracy retrieval with multiple nested ranker. In *SIGIR '06* (pp. 437–444). ACM. ISBN 1-59593-369-7, https://doi.org/10.1145/1148170.1148246, http://doi.acm.org/10.1145/1148170.1148246

McDonald, R., Brokos, G., & Androutsopoulos, I. (2018). Deep relevance ranking using enhanced document-query interactions. In *EMNLP '18* (pp. 1849–1860). ACL. http://aclweb.org/anthology/D18-1211

Mitra, B., Diaz, F., & Craswell, N. (2017). Learning to match using local and distributed representations of text for web search. In *WWW'17* (pp. 1291–1299). ISBN 978-1-4503-4913-0. https://doi.org/10.1145/3038912.3052579, https://doi.org/10.1145/3038912.3052579

Mitra, B., Nalisnick, E. T., Craswell, N., & Caruana, R. (2016). A dual embedding space model for document ranking. arXiv:1602.01137, http://arxiv.org/abs/1602.01137

Mueller, J., & Thyagarajan, A. (2016). Siamese recurrent architectures for learning sentence similarity. In *Proceedings of the thirtieth AAAI conference on artificial intelligence, AAAI'16* (pp. 2786–2792).

Nalisnick, E., Mitra, B., Craswell, N., & Caruana, R. (2016). Improving document ranking with dual word embeddings. In *WWW '16 companion* (pp. 83–84). ISBN 978-1-4503-4144-8, https://doi.org/10.1145/2872518.2889361, https://doi.org/10.1145/2872518.2889361

Nie, Y., Li, Y., & Nie, J.-Y. (2018a). Empirical study of multi-level convolution models for ir based on representations and interactions. In *ICTIR '18* (pp. 59–66). ACM. ISBN 978-1-4503-5656-5, https://doi.org/10.1145/3234944.3234954

Nie, Y., Sordoni, A., & Nie, J.-Y. (2018b). Multi-level abstraction convolutional model with weak supervision for information retrieval. In *SIGIR '18* (pp. 985–988). ACM. ISBN 978-1-4503-5657-2, https://doi.org/10.1145/3209978.3210123

Nogueira, R., & Cho, K. (2019). Passage re-ranking with BERT. CoRR: abs/1901.04085, http://arxiv.org/abs/1901.04085

Nogueira, R., Yang, W., Cho, K., & Lin, J. (2019). Multi-stage document ranking with bert.

Palangi, H., Deng, L., Shen, Y., Gao, J., He, X., Chen, J., Song, X., & Ward, R. (2016). Deep sentence embedding using long short-term memory networks: Analysis and application to information retrieval. *IEEE/ACM Transactions on Audio, Speech, and Language Processing, 24*(4), 694–707.

Pang, L., Lan, Y., Guo, J., Xu, J., & Cheng, X. (2016). A study of MatchPyramid models on ad-hoc retrieval. arXiv:1606.04648

Pang, L., Lan, Y., Guo, J., Xu, J., Xu, J., & Cheng, X. (2017). DeepRank: A new deep architecture for relevance ranking in information retrieval. In *Proceedings of the 2017 ACM on conference on information and knowledge management, CIKM '17* (pp. 257–266). ACM. ISBN 978-1-4503-4918-5, https://doi.org/10.1145/3132847.3132914

Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep contextualized word representations. In *Proceedings of the 2018 conference of the north american chapter of the association for computational linguistics: Human language technologies* (Vol. 1, Long Papers, pp. 2227–2237).

Qin, Z., Jagerman, R., Hui, K., Zhuang, H., Wu, J., Shen, J., Liu, T., Liu, J., Metzler, D., & Wang, X. (2023). Large language models are effective text rankers with pairwise ranking prompting. arXiv:2306.17563

Qiu, X., & Huang, X. (2015). Convolutional neural tensor network architecture for community-based question answering. In *Proceedings of the 24th international conference on artificial intelligence, IJCAI'15* (pp. 1305–1311).

Radford, A., Jeffrey, W., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. *OpenAI blog, 1*(8), 9.

Reimers, N., & Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. In *EMNLP/IJCNLP (1)* (pp. 3980–3990). Association for Computational Linguistics.

Robertson, S., & Zaragoza, H. (2009). The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends in Information Retrieval, 3*(4), 333–389.

Rudra, K., & Anand, A. (2020). Distant supervision in bert-based adhoc document retrieval. In *Proceedings of the 29th ACM international conference on information and knowledge management, CIKM '20* (pp. 2197–2200)

Shen, Y., He, X., Gao, J., Deng, L., & Mesnil, G. (2014a). A latent semantic model with convolutional-pooling structure for information retrieval. In *CIKM '14* (pp. 101–110). ACM. ISBN 978-1-4503-2598-1. https://doi.org/10.1145/2661829.2661935

Shen, Y., He, X., Gao, J., Deng, L., & Mesnil, G. (2014b). Learning semantic representations using convolutional neural networks for web search. In *WWW '14 companion* (pp. 373–374). ACM. ISBN: 978-1-4503-2745-9. http://doi.acm.org/10.1145/2567948.2577348

Strohman, T., Metzler, D., Turtle, H., & Croft, W. B. (2005). Indri: A language model-based search engine for complex queries. In *Proceedings of the international conference on intelligent analysis* (Vol. 2, pp. 2–6).

Sukhbaatar, S., Bruna, J., Paluri, M., Bourdev, L., & Fergus, R. (2014). Training convolutional networks with noisy labels. arXiv:1406.2080

Sun, W., Yan, L., Ma, X., Ren, P., Yin, D., & Ren, Z. (2023). Is chatgpt good at search? Investigating large language models as re-ranking agent. arXiv:2304.09542

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems* (Vol. 30).

Veit, A., Alldrin, N., Chechik, G., Krasin, I., Gupta, A., & Belongie, S. (2017). Learning from noisy large-scale datasets with minimal supervision. In *IEEE CVPR'17* (pp. 839–847).

Wan, S., Lan, Y., Guo, J., Xu, J., Pang, L., & Cheng, X. (2016). A deep architecture for semantic matching with multiple positional sentence representations. In *Proceedings of the thirtieth AAAI conference on artificial intelligence, AAAI'16* (pp. 2835–2841).

White, R. W, Jose, J. M, & Ruthven, I. (2002). Comparing explicit and implicit feedback techniques for web retrieval: Trec-10 interactive track report. In *Proceedings of the tenth text retrieval conference (TREC-10)* (pp. 534–538).

Wu, Z., Mao, J., Liu, Y., Zhang, M., & Ma, S. (2019). Investigating passage-level relevance and its role in document-level relevance judgment. In *SIGIR'19* (pp. 605–614).

Wu, Z., Mao, J., Liu, Y., Zhan, J., Zheng, Y., Zhang, M., & Ma, S. (2020). Leveraging passage-level cumulative gain for document ranking. In *Proceedings of the web conference 2020* (pp. 2421–2431).

Xiao, T., Xia, T., Yang, Y., Huang, C., & Wang, X. (2015). Learning from massive noisy labeled data for image classification. In *IEEE CVPR'15* (pp. 2691–2699).

Xiong, C., Dai, Z., Callan, J., Liu, Z., & Power, R. (2017). End-to-end neural ad-hoc ranking with kernel pooling. In *SIGIR '17* (pp. 55–64). ACM. ISBN 978-1-4503-5022-8, https://doi.org/10.1145/3077136.3080809

Xiong, L., Xiong, C., Li, Y., Tang, K.-F., Liu, J., Bennett, P., Ahmed, J., & Overwijk, A. (2020). Approximate nearest neighbor negative contrastive learning for dense text retrieval. arXiv:2007.00808

Yang, L., Ai, Q., Guo, J., & Bruce Croft, W. (2016). Anmm: Ranking short answer texts with attention-based neural matching model. In *Proceedings of the 25th ACM international on conference on information and knowledge management, CIKM '16* (pp. 287–296).

Yang, W., Zhang, H., & Lin, J. (2019). Simple applications of bert for ad hoc document retrieval. arXiv:1903.10972

Yilmaz, Z. A., Yang, W., Zhang, H., & Lin, J. (2019). Cross-domain modeling of sentence-level evidence for document retrieval. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)* (pp. 3481–3487).

Zhang, K., Xiong, C., Liu, Z., & Liu, Z. (2020). Selective weak supervision for neural information retrieval. In *Proceedings of the web conference 2020* (pp. 474–485).

Zhiwen, T., & Grace, H. Y. (2019). Deeptilebars: Visualizing term distribution for neural information retrieval. *Proceedings of the AAAI conference on artificial intelligence, 33*(01), 289–296.

Zhuang, S., & Zuccon, G. (2021a). Fast passage re-ranking with contextualized exact term matching and efficient passage expansion. arXiv preprint arXiv:2108.08513

Zhuang, S., & Zuccon, G. (2021b). Tilde: Term independent likelihood model for passage re-ranking. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval, SIGIR '21* (pp. 1483–1492).

Authors and Affiliations

Koustav Rudra[1] · Zeon Trevor Fernando[2] · Avishek Anand[3]

✉ Koustav Rudra
krudra@cai.iitkgp.ac.in

Zeon Trevor Fernando
zeon.trevor@gmail.com

Avishek Anand
Avishek.Anand@tudelft.nl

[1] Indian Institute of Technology Kharagpur, Kharagpur, India

[2] ImmobilienScout GmbH, Berlin, Germany

[3] Delft University of Technology, Delft, The Netherlands