Injecting the BM25 Score as Text Improves BERT-Based Re-rankers

Arian Askari[1], Amin Abolghasemi[1], Gabriella Pasi[2], Wessel Kraaij[1], and Suzan Verberne[1]

[1] Leiden Institute of Advanced Computer Science, Leiden University
{a.askari,m.a.abolghasemi,w.kraaij,s.verberne}@liacs.leidenuniv.nl
[2] Department of Informatics, Systems and Communication, University of Milano-Bicocca gabriella.pasi@unimib.it

Abstract. In this paper we propose a novel approach for combining first-stage lexical retrieval models and Transformer-based re-rankers: we inject the relevance score of the lexical model as a token in the middle of the input of the cross-encoder re-ranker. It was shown in prior work that interpolation between the relevance score of lexical and BERT-based re-rankers may not consistently result in higher effectiveness. Our idea is motivated by the finding that BERT models can capture numeric information. We compare several representations of the BM25 score and inject them as text in the input of four different cross-encoders. We additionally analyze the effect for different query types, and investigate the effectiveness of our method for capturing exact matching relevance. Evaluation on the MSMARCO Passage collection and the TREC DL collections shows that the proposed method significantly improves over all cross-encoder re-rankers as well as the common interpolation methods. We show that the improvement is consistent for all query types. We also find an improvement in exact matching capabilities over both BM25 and the cross-encoders. Our findings indicate that cross-encoder re-rankers can efficiently be improved without additional computational burden and extra steps in the pipeline by explicitly adding the output of the first-stage ranker to the model input, and this effect is robust for different models and query types.

Keywords: Injecting BM25 · Two-stage retrieval · Transformer-based rankers · BM25 · Combining lexical and neural rankers

1 Introduction

The commonly used ranking pipeline consists of a first-stage retriever, e.g. BM25 [47], that efficiently retrieves a set of documents from the full document collection, followed by one or more re-rankers [59,40] that improve the initial ranking. Currently, the most effective re-rankers are BERT-based rankers with a cross-encoder architecture, concatenating the query and the candidate document in the input [40,2,25,44]. In this paper, we refer to these re-rankers as Cross-Encoder$_{\text{CAT}}$ (CE$_{\text{CAT}}$). In the common re-ranking set-up, BM25 [47] is

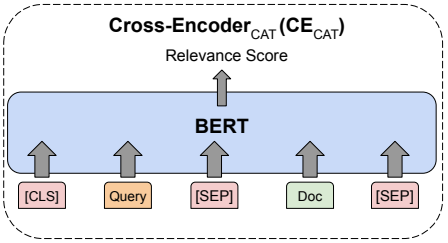

Fig. 1. Regular cross-encoder input

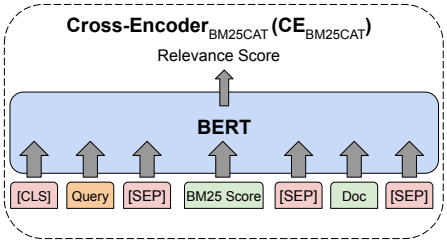

Fig. 2. Injection of BM25 in input

widely leveraged [7,27,20] for finding the top-k documents to be re-ranked; however, the relevance score produced by BM25 based on exact lexical matching is not explicitly taken into account in the second stage. Besides, although cross-encoder re-rankers substantially improve the retrieval effectiveness compared to BM25 alone [34], Rau et al. [43] show that BM25 is a more effective *exact lexical matcher* than CE_{CAT} rankers; in their exact-matching experiment they only use the words from the passage that also appear in the query as the input of the CE_{CAT}. This suggests that CE_{CAT} re-rankers can be further improved by a better exact word matching, as the presence of query words in the document is one of the strongest signals for relevance in ranking [48,50]. Moreover, obtaining improvement in effectiveness by interpolating the scores (score fusion [58]) of BM25 and CE_{CAT} is challenging: a linear combination of the two scores has shown to decrease effectiveness on the MSMARCO Passage collection compared to only using the CE_{CAT} re-ranker in the second stage retrieval [34].

To tackle this problem, in this work, we propose a method to enhance CE_{CAT} re-rankers by directly injecting the BM25 score as a string to the input of the Transformer. Figure 2 show our method for the injection of BM25 in the input of the CE re-ranker. We refer to our method as $\text{CE}_{\text{BM25CAT}}$. Our idea is inspired by the finding by Wallace et al. [54] that BERT models can capture numeracy. In this regard, we address the following research questions:

Décris dans la partie 2

RQ1: What is the effectiveness of BM25 score injection in addition to the query and document text in the input of CE re-rankers?

To answer this question we setup two experiments on three datasets: MSMARCO, TREC DL'19 and '20. First, since the BM25 score has no defined range, we investigate the effect of different representations of the BM25 score by applying various normalization methods. We also analyze the effect of converting the normalized scores of BM25 to integers. Second, we evaluate the best representation of BM25 – based on our empricial study – on four cross-encoders: BERT-base, BERT-large [53], DistillBERT [49], and MiniLM [56], comparing $\text{CE}_{\text{BM25CAT}}$ to CE_{CAT} across different Transformer models with a smaller and larger number of parameters. Next, we compare our proposed approach to common interpolation approaches:

RQ2: What is the effectiveness of $\text{CE}_{\text{BM25CAT}}$ compared to common approaches for combining the final relevance scores of CE_{CAT} and BM25?

To analyze $\text{CE}_{\text{BM25CAT}}$ and CE_{CAT} in terms of exact matching compared to BM25 we address the following question:

RQ3: How effective can $\text{CE}_{\text{BM25CAT}}$ capture exact matching relevance compared to BM25 and CE_{CAT}?
Furthermore, to provide an explanation on the improvement of $\text{CE}_{\text{BM25CAT}}$, we perform a qualitative analysis of a case where CE_{CAT} fails to identify the relevant document that is found using $\text{CE}_{\text{BM25CAT}}$ with the help of the BM25 score.[3]

Hummmm

To the best of our knowledge, there is no prior work on the effectiveness of cross-encoder re-rankers by injecting a retrieval model's score into their input. Our main contributions in this work are four-fold:

1. We provide a strategy for efficiently utilizing BM25 in cross-encoder re-rankers, which yields statistically significant improvements on all official metrics and is verified by thorough experiments and analysis.
2. We find that our method is more effective than the approaches which linearly interpolate the scores of BM25 and CE_{CAT}.
3. We analyze the exact matching effectiveness of CE_{CAT} and $\text{CE}_{\text{BM25CAT}}$ in comparison to BM25. We show that $\text{CE}_{\text{BM25CAT}}$ is a more powerful exact matcher than BM25 while CE_{CAT} is less effective than BM25.
4. We analyze the effectiveness of CE_{CAT} and $\text{CE}_{\text{BM25CAT}}$ on different query types. We show that $\text{CE}_{\text{BM25CAT}}$ consistently outperforms CE_{CAT} over all type of queries.

After a discussion of related work in Section 2, we describe the retrieval models employed in section 3 and the specifics of our experiments and methods in Section 4. The results are examined and the research questions are addressed in Section 5. Finally, the conclusion is described in Section 6.

2 Related work

Modifying the input of re-rankers. Boualili et al. [12,13] propose a method for highlighting exact matching signals by marking the start and the end of each occurrence of the query terms by adding markers to the input. In addition, they modify original passages and expand each passage with a set of generated queries using Doc2query [41] to overcome the vocabulary mismatch problem. This strategy is different from ours in two aspects: (1) the type of information added to the input: they add four tokens as markers for each occurrence of query terms, adding a burden to the limited input length of 512 tokens for query and document together, while we only add the BM25 score. (2) The need for data augmentation: they need to train a Doc2query model to provide the exact matching signal for improving the BERT re-ranker while our strategy does not need any extra overhead in terms of data augmentation. A few recent, but less related examples are Al-Hajj et al. [4], who experiment with the use of different

[3] In this work, we interchangeably use the words document and passage to refer to unit that should be retrieved.

supervised signals into the input of the cross-encoder to emphasize target words in context and Li et al. [30], who insert boundary markers into the input between contiguous words for Chinese named entity recognition.

Numerical information in Transformer models. Thawani et al. [52] provide an extensive overview of numeracy in NLP models up to 2021. Wallace et al. [54] analyze the ability of BERT models to work with numbers and come to the conclusion that the models capture numeracy and are able to do numerical reasoning; however the models appeared to struggle with interpreting floats. Moreover, Zhang et al. [63] show that BERT models capture a significant amount of information about numerical scale except for general common-sense reasoning. There are various studies that are inspired by the fact that Transformer models can correctly process numbers [11,38,26,15,22,21]. Gu et al. [23] incorporate text, categorical and numerical data as different modalities with Transformers using a combining module accross different classification tasks. They discover that adding tabular features increases the effectiveness while using only text is insufficient and results in the worst performance.

Methods for combining rankers. Linearly interpolating different rankers' scores has been studied extensively in the literature [34,58,10,9,8]. In this paper, we investigate multiple linear and non-linear interpolation ensemble methods to analyze the performance of them for combining BM25 and CE_{CAT} scores in comparison to $\text{CE}_{\text{BM25CAT}}$. For the sake of a fair analysis, we do not compare $\text{CE}_{\text{BM25CAT}}$ with a Learning-to-rank approach that is trained on 87 features by [65]. The use of ensemble methods brings additional overhead in terms of efficiency because it adds one more extra step to the re-ranking pipeline. It is noteworthy to mention that in this paper, we concentrate on analyzing the improvement by combining the first-stage retriever and a BERT-based re-ranker: BM25 and CE_{CAT} respectively. However, we are aware that combining scores of BM25 and Dense Retrievers that both are first-stage retrievers has also shown improvements [55,1,6] that are outside the scope of our study. In particular, CLEAR [20] proposes an approach to train the dense retrievers to encode semantics that BM25 fails to capture for first stage retrieval. However, in this study, our aim is to improve re-ranking in the second stage of two-stage retrieval setting.

Ok why not, il existe un truc plus performant mais plus couteux

3 Methods

3.1 First stage ranker: BM25

Lexical retrievers estimate the relevance of a document to a query based on word overlap [46]. Many lexical methods, including vector space models, Okapi BM25, and query likelihood, have been developed in previous decades. We use BM25 because of its popularity as first-stage ranker in current systems. Based on the statistics of the words that overlap between the query and the document, BM25 calculates a score for the pair:

$$s_{lex}(q,d) = BM25(q,d) = \sum_{t \in q \cap d} rsj_t . \frac{tf_{t,d}}{tf_{t,d} + k_1\{(1-b) + b\frac{|d|}{l}\}} \quad (1)$$

where t is a term, $tf_{t,d}$ is the frequency of t in document d, rsj_t is the Robertson-Spärck Jones weight [47] of t, and l is the average document length. k_1 and b are parameters [33,32].

3.2 CE_{CAT}: cross-encoder re-rankers without BM25 injection

Concatenating query and passage input sequences is the typical method for using cross-encoder (e.g., BERT) architectures with pre-trained Transformer models in a re-ranking setup [40,36,60,25]. This basic design is referred to as CE_{CAT} and shown in Figure 1. The query $q_{1:m}$ and passage $p_{1:n}$ sequences are concatenated with the $[SEP]$ token, and the $[CLS]$ token representation computed by CE is scored with a single linear layer W_s in the CE_{CAT} ranking model:

C'est quoi ce token wsh

$$CE_{CAT}(q_{1:m}, p_{1:n}) = CE([CLS]\, q\, [SEP]\, p\, [SEP]) * W_s \quad (2)$$

We use CE_{CAT} as our baseline re-ranker architecture. We evaluate different cross-encoder models in our experiments and all of them follow the above design.

3.3 $\text{CE}_{\text{BM25CAT}}$: cross-encoder re-rankers with BM25 injection

To study the effectiveness of injecting the BM25 score into the input, we modify the input of the basic input format as follows and call it $\text{CE}_{\text{BM25CAT}}$:

$$CE_{BM25CAT}(q_{1:m}, p_{1:n}) = CE([CLS]\, q\, [SEP]\, BM25\, [SEP]\, p\, [SEP]) * W_s \quad (3)$$

where BM25 represent the relevance score produced by BM25 between query and passage.

We study different representations of BM25 to find the optimal approach for injecting BM25 into the cross-encoders. The reasons are: (1) BM25 scores do not have an upper bound and should be normalized for having an interpretable score given a query and passage; (2) BERT-based models can process integers better than floating point numbers [54] so we analyze if converting the normalized score to an integer is more effective than injecting the floating point score. For normalizing BM25 scores, we compare three different normalization methods: Min-Max, Standardization (Z-score), and Sum:

$$Min\text{-}Max(s_{BM25}) = \frac{s_{BM25} - s_{min}}{s_{max} - s_{min}} \quad (4)$$

$$Standard(s_{BM25}) = \frac{s_{BM25} - \mu(S)}{\sigma(S)} \quad (5)$$

$$Sum(s_{BM25}) = \frac{s_{BM25}}{sum(S)} \quad (6)$$

Where s_{BM25} is the original score, and s_{max} and s_{min} are the maximum and minimum scores respectively, in the ranked list. $Sum(S)$, $\mu(S)$, and $\sigma(S)$ refer to sum, average and standard deviation over the scores of all passages retrieved

for a query. The anticipated effect of the Sum normalizer is that the sum of the scores of all passages in the ranked list will be 1; thus, if the top-n passages receive much higher scores than the rest, their normalized scores will have a larger difference with the rest of passages' scores in the ranked list; this distance could give a good signal to $\text{CE}_{\text{BM25CAT}}$. We experiment with Min-Max and Standardization in a local and a global setting. In the local setting, we get the minimum or maximum (for Min-Max) and mean and standard deviation (for Standard) from the ranked list of scores per query. In the global setting, we use $\{0, 50, 42, 6\}$ as {minimum, maximum, mean, standard deviation} as they have been empirically suggested in prior work to be used as default values across different queries to globally normalize BM25 scores [37]. In our data, the {minimum, maximum, mean, standard deviation} values are $\{0, 98, 7, 5\}$ across all queries. Because of the differences between the recommended defaults and the statistics of our collections, we explore other global values for Min-Max, using $25, 50, 75, 100$ as maximum and 0 as minimum. However, we got the best result using default values of [37]. To convert the float numbers to integers we multiply the normalized score to 100 and discard decimals. Finally, we store the number as a string.

3.4 Linear interpolation ensembles of BM25 and CE_{CAT}

We compare our approach to common ensemble methods [34,64] for interpolating BM25 and BERT re-rankers. We combine the scores linearly using the following methods: (1) Sum: compute sum over BM25 and CE_{CAT} scores, (2) Max: select maximum between BM25 and CE_{CAT} scores, and (3) Weighted-Sum:

$$s_i = \alpha \,.\, s_{BM25} + (1 - \alpha) \,.\, s_{CE_{CAT}} \tag{7}$$

Where s_i is the weighted sum produced by the interpolation, s_{BM25} is the normalized BM25 score, $s_{CE_{CAT}}$ is the CE_{CAT} score, and $\alpha \in [0..1]$ is a weight that indicates the relative importance. Since CE_{CAT} score $\in [0, 1]$, we also normalize BM25 score using Min-Max normalization. Furthermore, we train ensemble models that take s_{BM25} and $s_{CE_{CAT}}$ as features. We experiment with three different classifiers for this purpose: SVM with a linear kernel, SVM with an RBFkernel, Naive Bayes, and Multi Layer Perceptron (MLP) as a non-linear method and report the best classifier performance in Section 5.1.

Train avec quoi ??
en X et en y

4 Experimental design

Dataset and metrics. We conduct our experiments on the MSMARCO-passage collection [39] and the two TREC Deep Learning tracks (TREC-DL'19 and TREC-DL'20) [19,17]. The MSMARCO-passage dataset contains about 8.8 million passages (average length: 73.1 words) and about 1 million natural language queries (average length: 7.5 words) and has been extensively used to train deep language models for ranking because of the large number of queries. Following prior work on MSMARCO [28,34,35,68,67], we use the dev set ($\sim 7k$ queries) for

our empirical evaluation. MAP@1000 and $nDCG$@10 are calculated in addition to the official evaluation metric MRR@10. The passage corpus of MSMARCO is shared with TREC DL'19 and DL'20 collections with 43 and 54 queries respectively. We evaluate our experiments on these collections using $nDCG$@10 and MAP@1000, as is standard practice in TREC DL [17,19] to make our results comparable to previously published and upcoming research. We cap the query length at 30 tokens and the passage length at 200 tokens following prior work [25].

Pas certain des metrics ni de la méthode d'éval

Training configuration and model parameters. We use the Huggingface library [57], Cross-encoder package of Sentence-transformers library [45], and PyTorch [42] for the cross-encoder re-ranking training and inference. For injecting the BM25 score as text, we pass the BM25 score in string format into the BERT tokenizer in a similar way to passing query and document. Please note that the integer numbers are already included in the BERT tokenizer's vocabulary, allowing for appropriate tokenization. Following prior work [25] we use the Adam [29] optimizer with a learning rate of $7 * 10^{-6}$ for all cross-encoder layers, regardless of the number of layers trained. To train cross-encoder re-rankers for each TREC DL collection, we use the other TREC DL query set as the validation set and we select both TREC DL ('19 and '20) query sets as the validation set to train CEs for the MSMARCO Passage collection. We employ early stopping, based on the nDCG@10 value of the validation set. We use a training batch size of 32. For all cross-Encoder re-rankers, we use Cross-Entropy loss [66] . For the lexical retrieval with BM25 we employ the tuned parameters from the Anserini documentation [33,32]. [4]

Plein d'info pratiques

5 Results

5.1 Main results: addressing our research questions

Choice of BM25 score representation As introduced in Section 3.3, we compare different representations of the BM25 score in Table 1 for injection into $\text{CE}_{\text{BM25CAT}}$. We chose MiniLM [56] for this study as it has shown competitive results in comparison to BERT-based models while it is 3 times smaller and 6 times faster. [5] Our first interesting observation is that injecting the original float score rounded down to 2 decimal points (row b) of BM25 into the input seems to slightly improve the effectiveness of re-ranker. We assume this is due to the fact that the average query and passage length is relatively small in the MSMARCO Passage collection, which prevents from getting high numbers – with low interpretability for BERT – as BM25 score. Second, we find that the normalized BM25 score with Min-Max in the global normalization setting converted to integer (row f) is the most significant effective[6] representation for injecting BM25.

[4] The code is available on https://github.com/arian-askari/injecting_bm25_score_bert

[5] `https://huggingface.co/microsoft/MiniLM-L12-H384-uncased`

[6] Although the evaluation metrics are not in an interval scale, Craswell et al. [18] show that they are mostly reliable in practice on MSMARCO for statistical testing.

Table 1. Effectiveness results. Lines *b*-*n* refer to the $\text{MiniLM}_{\text{BMCAT}}$ re-ranker using different representations of the BM25 score as text. Significance is shown with † for the best result (row *f*) compared to $\text{MiniLM}_{\text{CAT}}$ (row *a*). Statistical significance was measured with a paired t-test ($p < 0.05$) with Bonferroni correction for multiple testing.

Normalization	Local/Global	Float/Integer	MSMARCO DEV		
			nDCG@10	MAP	MRR@10
(*a*) $\text{MiniLM}_{\text{CAT}}$ (without injecting BM25 score)			.419	.363	.360
(*b*) Original Score	—	—	.420	.364	.362
(*c*) Min-Max	Local	Float	.411	.359	.354
(*d*) Min-Max	Local	Integer	.414	.361	.355
(*e*) Min-Max	Global	Float	.422	.365	.363
(*f*) Min-Max	Global	Integer	**.424**†	**.368**†	**.367**†
(*g*) Standard	Local	Float	.407	.355	.352
(*h*) Standard	Local	Integer	.410	.358	.354
(*i*) Standard	Global	Float	.420	.363	.361
(*j*) Standard	Global	Integer	.421	.365	.363
(*k*) Sum	-	Float	.402	.349	.338
(*l*) Sum	-	Integer	.405	.350	.342

The global normalization setting gives better results for both Min-Max (rows e, f) and Standardization (rows i, j) than local normalization (rows c, d and g, h).[7] The reason is probably that in the global setting a candidate document obtains a high normalized score (close to 1 in the floating point representation) if its original score is close to default maximum (for Min-Max normalization) so the normalized score could be more interpretable across different queries. On the other hand, in the local setting, the passages ranked at position 1 always receive 1 as normalized score with Min-Max even if its original score is not high and it does not have a big difference with the last passage in the ranked list.

Moreover, converting the normalized float score to integers gives better results for both Min-Max (rows d, f) and Standardization (rows h, j) than the float representation (rows c, e and g, i). We find that Min-Max normalization is a better representation for injecting BM25 than Standardization, which could be due to the fact that in Min-Max the normalized score could not be negative, and, as a result, interpreting the injected score is easier for $\text{CE}_{\text{BM25CAT}}$. We find that the Sum normalizer (rows k and l) decreases effectiveness. Apparently, our expectation that Sum would help distinguish between the top-n passages and the remaining passages in the ranked list (see Section 5.1) is not true.

[7] The range of normalized integer scores using the best normalizer (row *f*) are from 0 to 196 as the maximum BM25 score in the collection is 98.

Table 2. Effectiveness results. Fine-tuned cross-encoders are used for re-ranking over BM25 first stage retrieval with a re-ranking depth of 1000. † indicates a statistically significant improvement of a cross-encoder with BM25 score injection as text into the input (Cross-encoder$_{\text{BM25CAT}}$) over the same cross-encoder without BM25 score injection (Cross-encoder$_{\text{CAT}}$). Statistical significance was measured with a paired t-test ($p < 0.05$) with Bonferroni correction for multiple testing.

Model	**TREC DL 20**		**TREC DL 19**		**MSMARCO DEV**		
	nDCG@10	MAP	nDCG@10	MAP	nDCG@10	MAP	MRR@10
BM25	.480	.286	.506	.377	.234	.195	.187
Re-rankers							
BERT-Base$_{\text{CAT}}$	.689	.447	.713	.441	.399	.346	.342
BERT-Base$_{\text{BM25CAT}}$	.705†	.475†	.723†	.453†	.422†	.367†	.364†
BERT-Large$_{\text{CAT}}$	.695	.464	.714	.467	.401	.344	.360
BERT-Large$_{\text{BM25CAT}}$	**.728**†	**.482**†	**.731**†	**.477**†	**.424**†	.367†	**.369**†
DistilBERT$_{\text{CAT}}$	.670	.442	.679	.440	.383	.310	.325
DistilBERT$_{\text{BM25CAT}}$	.682†	.456†	.699†	.451†	.390†	.323†	.339†
MiniLM$_{\text{CAT}}$	.681	.448	.704	.452	.419	.363	.360
MiniLM$_{\text{BM25CAT}}$	.710†	.473†	.711†	.463†	**.424**†	**.368**†	.367†

Impact of BM25 injection for various cross-encoders (RQ1) Table 2 shows that injecting the BM25 score – using the best normalizer which is Min-Max in the global normalization setting converted to integer – into all four cross-encoders improves their effectiveness in all of the metrics compared to using them without injecting BM25. This shows that injecting the BM25 score into the input as a small modification to the current re-ranking pipeline improves the re-ranking effectiveness. This is without any additional computational burden as we train CE$_{\text{CAT}}$ and CE$_{\text{BM25CAT}}$ in a completely equal setting in terms of number of epochs, batch size, etc. We receive the highest result by BERT-Large$_{\text{BM25CAT}}$ for cross-encoder with BM25 injection, which could be due to the higher number of parameters of the model. We find that the results of MiniLM are similar to those for BERT-Base on MSMARCO-DEV while the former is more efficient.

Comparing BM25 Injection with Ensemble Methods (RQ2) Table 3 shows that while injecting BM25 leads to improvement, regular ensemble methods and Naive Bayes classifier fail to do so; combining the scores of BM25 and BERT$_{\text{CAT}}$ in a linear and non-linear (MLP) interpolation ensemble setting even leads to lower effectiveness than using the cross-encoder as sole re-ranker. Therefore, our strategy is a better solution than linear interpolation. We only report results for Naive Bayes – having BM25 and BERT$_{\text{CAT}}$ score as features – as

Table 3. The effectiveness of injecting BM25 score into the input (Bert-Base$_{\text{BM25CAT}}$) compared to interpolation performance of BM25 and Bert-Base$_{\text{CAT}}$ using common ensemble methods.

Model	Ensemble	MSMARCO DEV		
		nDCG@10	MAP	MRR@10
BM25	—	.234	.195	.187
BERT-Base$_{\text{CAT}}$	—	.399	.346	.342
BM25 and BERT-Base$_{\text{CAT}}$	Sum	.270	.225	.218
BM25 and BERT-Base$_{\text{CAT}}$	Max	.237	.197	.190
BM25 and BERT-Base$_{\text{CAT}}$	Weighted-Sum (tuned)	.353	.295	.290
BM25 and BERT-Base$_{\text{CAT}}$	Naive Bayes	.314	.260	.254
BERT-Base$_{\text{BM25CAT}}$	BM25 Score Injection	**.422**	**.367**	**.364**

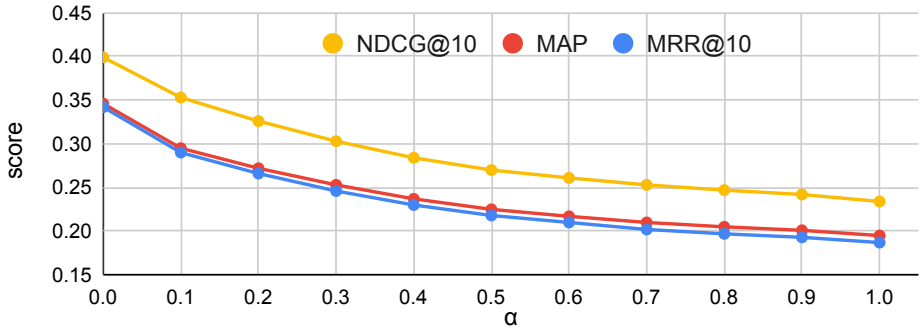

Fig. 3. Effectiveness on MSMARCO DEV with varying the interpolation weight of BM25 and BERT-Base$_{\text{CAT}}$ scores. $\alpha = 0$ means only BERT$_{\text{CAT}}$ scores are used.

it had the highest effectiveness of the four estimators. Still, the effectiveness is much lower than BERT$_{\text{BM25CAT}}$ and also lower than a simple Weighted-Sum. Weighted-Sum (tuned) in Table 3 is tuned on the validation set, for which $\alpha = 0.1$ was found to be optimal. We analyze the effect of different α values in a weighted linear interpolation (Weighted-Sum) to draw a more complete picture on the impact of combining scores on the DEV set. Figure 3 shows that by increasing the weight of BM25, the effectiveness decreases. The figure also shows that the tuned alpha which was found on the validation set in Table 3 is not the most optimal possible alpha value for the DEV set. The highest effectiveness for $\alpha = 0.0$ in Figure 3 confirms we should not combine the scores by current interpolation methods and only using scores of Bert-Base$_{\text{CAT}}$ is better, at least for the MS-MARCO passage collection.

Exact Matching Relevance Results (RQ3) To conduct exact matching analysis, we replace the passage words that do not appear in the query with

the $[MASK]$ token, leaving the model only with a skeleton of the original passage and force it to rely on the exact word matches between query and passage [43]. We do not train models on this input but use our models that were fine-tuned on the original data. Table 4 shows that BERT-Base$_{\text{BM25CAT}}$ performs better than both BM25 and BERT-Base$_{\text{CAT}}$ in the exact matching setting on all metrics. Moreover, we found that the percentage of relevant passages ranked in top-10 that are common between BM25 and BERT$_{\text{BM25CAT}}$ is 40%, which is higher than the percentage of relevant passages between BM25 and BERT$_{\text{CAT}}$ (37%). Therefore, the higher effectiveness of BERT$_{\text{BM25CAT}}$ in exact matching setting could be at least partly because it mimics BM25 more than BERT$_{\text{CAT}}$. In comparison, this percentage is 57 between BERT$_{\text{BM25CAT}}$ and BERT$_{\text{CAT}}$.

Table 4. Comparing exact matching effectiveness of BERT-Base$_{\text{BM25CAT}}$ and BERT-Base$_{\text{CAT}}$ by keeping only the query words in each passage for re-ranking. The increase and decrease of effectiveness compared to BM25 is indicated with ↑ and ↓.

Model	Input	MSMARCO DEV		
		nDCG@10	MAP	MRR@10
BM25	Full text	.234	.195	.187
BERT-Base$_{\text{CAT}}$	Only query words	.218 (↓1.6)	.186 (↓0.9)	.180 (↓0.7)
BERT-Base$_{\text{BM25CAT}}$	Only query words	**.243** (↑.9)	**.209** (↑1.4)	**.202** (↑1.5)

5.2 Analysis of the results

Query types. In order to analyze the effectiveness of BERT-base$_{\text{CAT}}$ and BERT-base$_{\text{BM25CAT}}$ across different types of questions, we classify questions based on the lexical answer type. We use the rule-based answer type classifier[8] inspired by [31] to extract answer types. We classify MSMARCO queries into 6 answer types: abbreviation, location, description, human, numerical and entity. 4105 queries have a valid answer type and at least one relevant passage in the top-1000. We perform our analysis in two different settings: normal (full-text) and exact-matching (keeping only query words and replacing non-query words with $[MASK]$). The average MRR@10 per query type is shown in Table 5. The table shows that BERT$_{\text{BM25CAT}}$ is more effective than BERT$_{\text{CAT}}$ consistently on all types of queries.

Qualitative analysis. We show a qualitative analysis of one particular case in Figure 4 to analyze more in-depth what the effect of BM25 injection is and why it works. In the top row, while BERT$_{\text{CAT}}$ mistakenly ranked the relevant passage at position 104, BM25 ranked that passage at position 3 and BERT$_{\text{BM25CAT}}$ – apparently helped by BM25 – ranked that relevant passage at position 1. In the

[8] `https://github.com/superscriptjs/qtypes`

Table 5. MRR@10 on MSMARCO-DEV per query type for comparing BERT-Base$_{\text{BM25CAT}}$ and BERT-Base$_{\text{CAT}}$ on different query types in full-text and exact-matching (only keeping query words) settings.

Model	Input	ABBR	LOC	DESC	HUM	NUM	ENTY
# queries		9	493	1887	455	933	328
BERT-BaseCAT	Full text	.574	.477	.397	.435	.361	.399
BERT-BaseBM25CAT	Full text	**.592**	**.503**	**.428**	**.457**	**.405**	**.411**
BM25	Only query words	.184	.256	.215	.238	.200	**.221**
BERT-BaseCAT	Only query words	.404	.204	.224	.240	.177	.200
BERT-BaseBM25CAT	Only query words	**.438**	**.278**	**.245**	**.258**	**.215**	.216

Fig. 4. Example query and two passages in the input of BERT$_{\text{BM25CAT}}$. The color of each word indicates the word-level attribution value according to Integrated Gradient (IG) [51], where red is positive, blue is negative, and white is neutral. We use the brightness of different colors to indicate the values of these gradients.

Query [SEP] BM25 [SEP] Passage	Label	Model: Rank
[CLS] what is the shingles jab ? [SEP] 22 [SEP] the shingles vaccine . the vaccine , called zostavax , is given as a single injection under the skin (subcutaneously) . it can be given at any time in the year . unlike with the flu jab	R	BM25: 3 BERT$_{\text{BM25CAT}}$: 1 BERT$_{\text{CAT}}$: 104
[CLS] what is the shingles jab ? [SEP] 11 [SEP] shingle is a corruption of german schindle (schindel) meaning a roofing slate . shingles historically were called tiles and shingle was a term applied to wood shingles , as is still mostly the case outside the us [SEP]	N	BM25: 146 BERT$_{\text{BM25CAT}}$: 69 BERT$_{\text{CAT}}$: 1

bottom row, BERT$_{\text{CAT}}$ mistakenly ranked the irrelevant passage at position 1 and informed by the low BM25 score, BERT$_{\text{BM25CAT}}$ ranked it much lower, at 69. In order to interpret the importance of the injected BM25 score in the input of CE$_{\text{BM25CAT}}$ and show its contributions to the matching score in comparison to other words in the query and passage, we use Integrated Gradient (IG) [51] which has been proven to be a stable and reliable interpretation method in many different applications including Information Retrieval [62,16,61]. [9] On both rows of Figure 4, we see that the BM25 score ('22' in the top row and '11' in the bottom row) is a highly attributed term in comparison to other terms. This shows that injecting the BM25 score assists BERT$_{\text{BM25CAT}}$ to identify relevant or non-relevant passages better than BERT$_{\text{CAT}}$.

As a more general analysis, we randomly sampled 100 queries from MSMARCO-DEV. For each query, we took the top-1000 passages retrieved by BM25, we fed all pairs of query and their corresponding retrieved passages ($100k$ pairs) into BERT$_{\text{BM25CAT}}$, and computed the attribution scores over the input at the word-

[9] We refer readers to [51] for a detailed explanation.

level. We ranked tokens based on their importance using the absolute value of their attribution score and found the mode of the rank of the BM25 token over all samples is 3. This shows that $\text{BERT}_{\text{BM25CAT}}$ highly attributes the BM25 token for ranking.

6 Conclusion and future work

In this paper we have proposed an efficient and effective way of combining BM25 and cross-encoder re-rankers: injecting the BM25 score as text in the input of the cross-encoder. We find that the resulting model, $\text{CE}_{\text{BM25CAT}}$, achieves a statistically significant improvement for all evaluated cross-encoders. Additionally, we find that our injection approach is much more effective than linearly interpolating the initial ranker and re-ranker scores. In addition, we show that $\text{CE}_{\text{BM25CAT}}$ performs significantly better in an exact matching setting than both BM25 and CE_{CAT} individually. This suggests that injecting the BM25 score into the input could modify the current paradigm for training cross-encoder re-rankers.

While it is crystal clear that our focus is not on chasing the state-of-the-art, we believe that as future work, our method could be applied into any cross-encoder in the current multi-stage ranking pipelines which are state-of-the-art for the MSMARCO Passage benchmark [24]. Moreover, previous studies show that combining BM25 and BERT re-rankers on *Robust04* [5] leads to improvement [3]. It is interesting to study the effect of injecting BM25 for this task because documents often have to be truncated to fit the maximum model input length [14]; injecting the BM25 score might give information to the cross-encoder re-ranker about the lexical relevance of the whole text of the document. Another interesting direction is to study how Dense Retrievers can benefit from injecting lexical ranker scores. Moreover, injecting scores of several lexical rankers and adding more traditional Learning-to-Rank features could be also interesting.

ACKNOWLEDGMENTS

This work was supported by the EU Horizon 2020 ITN/ETN on Domain Specific Systems for Information Extraction and Retrieval (H2020-EU.1.3.1., ID: 860721).

References

1. Abolghasemi, A., Askari, A., Verberne, S.: On the interpolation of contextualized term-based ranking with bm25 for query-by-example retrieval. In: Proceedings of the 2022 ACM SIGIR International Conference on Theory of Information Retrieval. p. 161–170. ICTIR '22, Association for Computing Machinery, New York, NY, USA (2022). https://doi.org/10.1145/3539813.3545133, https://doi.org/10.1145/3539813.3545133

2. Abolghasemi, A., Verberne, S., Azzopardi, L.: Improving bert-based query-by-document retrieval with multi-task optimization. In: European Conference on Information Retrieval. pp. 3–12. Springer (2022)
3. Akkalyoncu Yilmaz, Z., Wang, S., Yang, W., Zhang, H., Lin, J.: Applying BERT to document retrieval with birch. 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): System Demonstrations. pp. 19–24. Association for Computational Linguistics, Hong Kong, China (Nov 2019). https://doi.org/10.18653/v1/D19-3004, https://aclanthology.org/D19-3004
4. Al-Hajj, M., Jarrar, M.: Arabglossbert: Fine-tuning bert on context-gloss pairs for wsd. arXiv preprint arXiv:2205.09685 (2022)
5. Allan, J.: Overview of the trec 2004 robust retrieval track. In: Proceedings of TREC. vol. 13 (2004)
6. Althammer, S., Askari, A., Verberne, S., Hanbury, A.: DoSSIER@ COLIEE 2021: Leveraging dense retrieval and summarization-based re-ranking for case law retrieval. arXiv preprint arXiv:2108.03937 (2021)
7. Anand, M., Zhang, J., Ding, S., Xin, J., Lin, J.: Serverless bm25 search and bert reranking. In: DESIRES. pp. 3–9 (2021)
8. Askari, A., Verberne, S.: Combining lexical and neural retrieval with longformer-based summarization for effective case law retrieva. In: Proceedings of the second international conference on design of experimental search & information REtrieval systems. pp. 162–170. CEUR (2021)
9. Askari, A., Verberne, S., Pasi, G.: Expert finding in legal community question answering. In: Hagen, M., Verberne, S., Macdonald, C., Seifert, C., Balog, K., Nørvåg, K., Setty, V. (eds.) Advances in Information Retrieval. pp. 22–30. Springer International Publishing, Cham (2022)
10. Bartell, B.T., Cottrell, G.W., Belew, R.K.: Automatic combination of multiple ranked retrieval systems. In: SIGIR'94. pp. 173–181. Springer (1994)
11. Berg-Kirkpatrick, T., Spokoyny, D.: An empirical investigation of contextualized number prediction. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 4754–4764 (2020)
12. Boualili, L., Moreno, J.G., Boughanem, M.: Markedbert: Integrating traditional ir cues in pre-trained language models for passage retrieval. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 1977–1980 (2020)
13. Boualili, L., Moreno, J.G., Boughanem, M.: Highlighting exact matching via marking strategies for ad hoc document ranking with pretrained contextualized language models. Information Retrieval Journal pp. 1–47 (2022)
14. Boytsov, L., Lin, T., Gao, F., Zhao, Y., Huang, J., Nyberg, E.: Understanding performance of long-document ranking models through comprehensive evaluation and leaderboarding. arXiv preprint arXiv:2207.01262 (2022)
15. Chen, C.C., Huang, H.H., Chen, H.H.: Numclaim: Investor's fine-grained claim detection. In: Proceedings of the 29th ACM International Conference on Information & Knowledge Management. pp. 1973–1976 (2020)
16. Chen, L., Lan, Y., Pang, L., Guo, J., Cheng, X.: Toward the understanding of deep text matching models for information retrieval. arXiv preprint arXiv:2108.07081 (2021)
17. Craswell, N., Mitra, B., Yilmaz, E., Campos, D.: Overview of the trec 2020 deep learning track. arXiv preprint arXiv:2102.07662 (2021)

18. Craswell, N., Mitra, B., Yilmaz, E., Campos, D., Lin, J.: Ms marco: Benchmarking ranking models in the large-data regime. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 1566–1576 (2021)
19. Craswell, N., Mitra, B., Yilmaz, E., Campos, D., Voorhees, E.M.: Overview of the trec 2019 deep learning track. arXiv preprint arXiv:2003.07820 (2020)
20. Gao, L., Dai, Z., Chen, T., Fan, Z., Durme, B.V., Callan, J.: Complement lexical retrieval model with semantic residual embeddings. In: European Conference on Information Retrieval. pp. 146–160. Springer (2021)
21. Geva, M., Gupta, A., Berant, J.: Injecting numerical reasoning skills into language models. arXiv preprint arXiv:2004.04487 (2020)
22. Gretkowski, A., Wiśniewski, D., Ławrynowicz, A.: Should we afford affordances? injecting conceptnet knowledge into bert-based models to improve commonsense reasoning ability. In: Corcho, O., Hollink, L., Kutz, O., Troquard, N., Ekaputra, F.J. (eds.) Knowledge Engineering and Knowledge Management. pp. 97–104. Springer International Publishing, Cham (2022)
23. Gu, K., Budhkar, A.: A package for learning on tabular and text data with transformers. In: Proceedings of the Third Workshop on Multimodal Artificial Intelligence. pp. 69–73. Association for Computational Linguistics, Mexico City, Mexico (Jun 2021). https://doi.org/10.18653/v1/2021.maiworkshop-1.10, https://www.aclweb.org/anthology/2021.maiworkshop-1.10
24. Han, S., Wang, X., Bendersky, M., Najork, M.: Learning-to-rank with bert in tf-ranking. arXiv preprint arXiv:2004.08476 (2020)
25. Hofstätter, S., Althammer, S., Schröder, M., Sertkan, M., Hanbury, A.: Improving efficient neural ranking models with cross-architecture knowledge distillation. arXiv preprint arXiv:2010.02666 (2020)
26. Johnson, D., Mak, D., Barker, D., Loessberg-Zahl, L.: Probing for multilingual numerical understanding in transformer-based language models. arXiv preprint arXiv:2010.06666 (2020)
27. Kamphuis, C., Vries, A.P.d., Boytsov, L., Lin, J.: Which bm25 do you mean? a large-scale reproducibility study of scoring variants. In: European Conference on Information Retrieval. pp. 28–34. Springer (2020)
28. Khattab, O., Zaharia, M.: Colbert: Efficient and effective passage search via contextualized late interaction over bert. In: Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval. pp. 39–48 (2020)
29. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
30. Li, L., Dai, Y., Tang, D., Feng, Z., Zhou, C., Qiu, X., Xu, Z., Shi, S.: Markbert: Marking word boundaries improves chinese bert. arXiv preprint arXiv:2203.06378 (2022)
31. Li, X., Roth, D.: Learning question classifiers. In: COLING 2002: The 19th International Conference on Computational Linguistics (2002)
32. Lin, J., Ma, X., Lin, S.C., Yang, J.H., Pradeep, R., Nogueira, R.: Pyserini: A Python toolkit for reproducible information retrieval research with sparse and dense representations. In: Proceedings of the 44th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2021). pp. 2356–2362 (2021)
33. Lin, J., Ma, X., Lin, S.C., Yang, J.H., Pradeep, R., Nogueira, R.: Pyserini: Bm25 baseline for ms marco document retrieval (August 2021), https://github.com/castorini/pyserini/blob/master/docs/experiments-msmarco-doc.md

34. Lin, J., Nogueira, R., Yates, A.: Pretrained transformers for text ranking: Bert and beyond. Synthesis Lectures on Human Language Technologies **14**(4), 1–325 (2021)
35. MacAvaney, S., Nardini, F.M., Perego, R., Tonellotto, N., Goharian, N., Frieder, O.: Expansion via prediction of importance with contextualization. In: Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval. pp. 1573–1576 (2020)
36. MacAvaney, S., Yates, A., Cohan, A., Goharian, N.: Cedr: Contextualized embeddings for document ranking. In: Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval. pp. 1101–1104 (2019)
37. Michael, N., Diego, C., Joshua, P., LP, B.: Learning to rank (May 2022), https://solr.apache.org/guide/solr/latest/query-guide/learning-to-rank.html#feature-engineering
38. Muffo, M., Cocco, A., Bertino, E.: Evaluating transformer language models on arithmetic operations using number decomposition. In: Proceedings of the Thirteenth Language Resources and Evaluation Conference. pp. 291–297. European Language Resources Association, Marseille, France (Jun 2022), https://aclanthology.org/2022.lrec-1.30
39. Nguyen, T., Rosenberg, M., Song, X., Gao, J., Tiwary, S., Majumder, R., Deng, L.: Ms marco: A human generated machine reading comprehension dataset. In: CoCo@ NIPs (2016)
40. Nogueira, R., Cho, K.: Passage re-ranking with bert. arXiv preprint arXiv:1901.04085 (2019)
41. Nogueira, R., Yang, W., Lin, J., Cho, K.: Document expansion by query prediction. arXiv preprint arXiv:1904.08375 (2019)
42. Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A.: Automatic differentiation in pytorch (2017)
43. Rau, D., Kamps, J.: How different are pre-trained transformers for text ranking? In: European Conference on Information Retrieval. pp. 207–214. Springer (2022)
44. Rau, D., Kamps, J.: The role of complex nlp in transformers for text ranking. In: Proceedings of the 2022 ACM SIGIR International Conference on Theory of Information Retrieval. pp. 153–160 (2022)
45. Reimers, N., Gurevych, I.: Sentence-bert: Sentence embeddings using siamese bert-networks. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics (11 2019), https://arxiv.org/abs/1908.10084
46. Robertson, S., Zaragoza, H., et al.: The probabilistic relevance framework: Bm25 and beyond. Foundations and Trends® in Information Retrieval **3**(4), 333–389 (2009)
47. Robertson, S.E., Walker, S.: Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In: SIGIR'94. pp. 232–241. Springer (1994)
48. Salton, G., McGill, M.J.: Introduction to modern information retrieval. mcgraw-hill (1983)
49. Sanh, V., Debut, L., Chaumond, J., Wolf, T.: Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108 (2019)
50. SARACEVIC, T.: A review of an a framework for the thinking on the notion in information science. Journal of the American Society for Information Science **26**
51. Sundararajan, M., Taly, A., Yan, Q.: Axiomatic attribution for deep networks. In: International conference on machine learning. pp. 3319–3328. PMLR (2017)

52. Thawani, A., Pujara, J., Szekely, P.A., Ilievski, F.: Representing numbers in nlp: a survey and a vision. arXiv preprint arXiv:2103.13136 (2021)
53. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in neural information processing systems. pp. 5998–6008 (2017)
54. Wallace, E., Wang, Y., Li, S., Singh, S., Gardner, M.: Do nlp models know numbers? probing numeracy in embeddings. arXiv preprint arXiv:1909.07940 (2019)
55. Wang, S., Zhuang, S., Zuccon, G.: Bert-based dense retrievers require interpolation with bm25 for effective passage retrieval. In: Proceedings of the 2021 ACM SIGIR International Conference on Theory of Information Retrieval. p. 317–324. ICTIR '21, Association for Computing Machinery, New York, NY, USA (2021). https://doi.org/10.1145/3471158.3472233, https://doi.org/10.1145/3471158.3472233
56. Wang, W., Wei, F., Dong, L., Bao, H., Yang, N., Zhou, M.: Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. Advances in Neural Information Processing Systems **33**, 5776–5788 (2020)
57. Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., et al.: Huggingface's transformers: State-of-the-art natural language processing. arXiv preprint arXiv:1910.03771 (2019)
58. Wu, S.: Applying statistical principles to data fusion in information retrieval. Expert Systems with Applications **36**(2), 2997–3006 (2009)
59. Yan, M., Li, C., Wu, C., Xia, J., Wang, W.: Idst at trec 2019 deep learning track: Deep cascade ranking with generation-based document expansion and pre-trained language modeling. In: TREC (2019)
60. Yilmaz, Z.A., Yang, W., Zhang, H., Lin, J.: 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. 3490–3496 (2019)
61. Zhan, J., Mao, J., Liu, Y., Guo, J., Zhang, M., Ma, S.: Interpreting dense retrieval as mixture of topics. arXiv preprint arXiv:2111.13957 (2021)
62. Zhan, J., Mao, J., Liu, Y., Zhang, M., Ma, S.: An analysis of bert in document ranking. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 1941–1944 (2020)
63. Zhang, X., Ramachandran, D., Tenney, I., Elazar, Y., Roth, D.: Do language embeddings capture scales? arXiv preprint arXiv:2010.05345 (2020)
64. Zhang, X., Yates, A., Lin, J.: Comparing score aggregation approaches for document retrieval with pretrained transformers. In: Hiemstra, D., Moens, M.F., Mothe, J., Perego, R., Potthast, M., Sebastiani, F. (eds.) Advances in Information Retrieval. pp. 150–163. Springer International Publishing, Cham (2021)
65. Zhang, Y., Hu, C., Liu, Y., Fang, H., Lin, J.: Learning to rank in the age of muppets: Effectiveness–efficiency tradeoffs in multi-stage ranking. In: Proceedings of the Second Workshop on Simple and Efficient Natural Language Processing. pp. 64–73 (2021)
66. Zhang, Z., Sabuncu, M.: Generalized cross entropy loss for training deep neural networks with noisy labels. Advances in neural information processing systems **31** (2018)
67. Zhuang, S., Li, H., Zuccon, G.: Deep query likelihood model for information retrieval. In: European Conference on Information Retrieval. pp. 463–470. Springer (2021)

68. Zhuang, S., Zuccon, G.: 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. pp. 1483–1492 (2021)