Information Retrieval Journal (2022) 25:414–460
https://doi.org/10.1007/s10791-022-09414-x

Highlighting exact matching via marking strategies for ad hoc document ranking with pretrained contextualized language models

Lila Boualili[1] · Jose G. Moreno[1] · Mohand Boughanem[1]

Received: 21 May 2021 / Accepted: 27 June 2022 / Published online: 6 August 2022

Abstract
Pretrained language models (PLMs) exemplified by BERT have proven to be remarkably effective for ad hoc ranking. As opposed to pre-BERT models that required specialized neural components to capture different aspects of query-document relevance, PLMs are solely based on transformers where attention is the only mechanism used for extracting signals from term interactions. Thanks to the transformer's cross-match attention, BERT was found to be an effective soft matching model. However, exact matching is still an essential signal for assessing the relevance of a document to an information-seeking query aside from semantic matching. We assume that BERT might benefit from explicit exact match cues to better adapt to the relevance classification task. In this work, we explore strategies for integrating exact matching signals using marker tokens to highlight exact term-matches between the query and the document. We find that this simple marking approach significantly improves over the common vanilla baseline. We empirically demonstrate the effectiveness of our approach through exhaustive experiments on three standard ad hoc benchmarks. Results show that explicit exact match cues conveyed by marker tokens are beneficial for BERT and ELECTRA variant to achieve higher or at least comparable performance. Our findings support that traditional information retrieval cues such as exact matching are still valuable for large pretrained contextualized models such as BERT.

Keywords Deep learning · Pretrained language models · Ad hoc ranking · Exact term-matching

✉ Lila Boualili
lila.boualili@irit.fr

Jose G. Moreno
jose.moreno@irit.fr

Mohand Boughanem
mohand.boughanem@irit.fr

[1] IRIT, University of Toulouse III, Toulouse, France

1 Introduction

Pretrained Language Models (PLMs), such as BERT (Devlin et al., 2019), ELECTRA (Clark et al., 2020) and T5 (Raffel et al., 2020), have become the core components for building highly effective ranking models. The success of PLMs is largely owed to the heavy pre-training on language modeling objectives on the one hand, and learning deeply-contextualized representations of input sequences using the transformer architecture (Vaswani et al., 2017) on the other. Thanks to the fine-tuning strategy and the availability of large publicly-released training datasets, applying a PLM to document ranking is straightforward. Nogueira and Cho (2019) was the first to propose a simple application of BERT to text ranking using fine-tuning on the large public MS MARCO (Nguyen et al., 2016) dataset. In this work, BERT was deployed as a relevance classifier trained to estimate the probability each document is "relevant" w.r.t a given query.

Compared to the first wave of neural ranking models including DRMM (Guo et al., 2016), DUET (Mitra et al., 2017), and KNRM (Xiong et al., 2017), referred to as pre-BERT models, BERT and its variants do not appear to require any specialized neural architectural components to capture different aspects of relevance between a query and a document (Lin et al., 2020). The same architecture based on homogeneous transformer layers is employed regardless of the downstream task. Qiao et al. (2019) study the behaviour of BERT for ranking and revealed that it focuses more on document terms that directly match the query. Compared to pre-BERT models such as Conv-KNRM (Dai et al., 2018) that prefer terms related to the query in search, BERT's pretraining on surrounding contexts favors text sequence pairs that are closer in their semantic meaning (Qiao et al., 2019). Qiao et al. (2019) conclude that BERT can be considered as an interaction-based sequence-to-sequence soft matching model that owes its effectiveness to the transformer's cross-match attention. While soft semantic matching is, undeniably, a valuable signal for relevance that alleviates the vocabulary mismatch problem, a ranking model needs proper handling of exact matching cues as well (Guo et al., 2016; Mitra et al., 2017; Luan et al., 2020). Let us take the following query: "Causes of *left* ventricular hypertrophy" form the MS MARCO passage ranking task, as an example. Table 1 reports extracts from the top passages retrieved by BERT. We can see that all top ranked passages are related to "*right* ventricular hypertrophy" due to the soft matching between "left" and "right". This example is a reminder of the importance of exact matching for relevance ranking. Boualili et al. (2020) suggest that a PLM like BERT can benefit from explicit exact matching signals for passage ranking. The authors propose MarkedBERT, a model that uses marker tokens to convey exact matches between the query and document terms from the input sequence. Special tokens, i.e, $[e_i]$ and $[/e_i]$, were added to the textual input sequence of BERT to indicate the start and the end, respectively, of terms that match exactly with the i-th term of the query. For example:

Table 1 Extracts from top ranked passages by Vanilla BERT for the query: "causes of left ventricular hypertrophy"

ID	Passage
47203	Causes of *Right* Ventricular Hypertrophy. There are four usual causes of *right* ventricular hypertrophy.
5197133	The last common cause of *right* ventricular hypertrophy is the ventricular septal defect.
7504775	The most common causes of *right* ventricle hypertrophy (RVH) are diseases that damage the lung.

Query: Causes of [e_2]left[/e_2] [e_3]ventricular[/e_3] [e_4]hypertrophy[/e_4]
Passage: [e_2]Left[/e_2] [e_3]ventricular[/e_3] [e_4]hypertrophy[/e_4] can occur...

Exact term-matching integration via marking has proven to induce significant gains on the MSMARCO passage ranking task over "Vanilla" BERT (monoBERT) (Nogueira and Cho, 2019). Analysis of the attention shows that marker tokens bring more focus on the exact matches allowing more relevant documents to be ranked higher. Table 2 shows extracts from top ranked passages returned by MarkedBERT for the query "Causes of *left* ventricular hypertrophy" where we can count more documents related to "*left* ventricular hypertrophy" without explicit bias, since the passage 47203 ranked first by BERT is still ranked high (second) by MarkedBERT.

In this work, we follow the same hypothesis stating that exact matching cues can enhance PLMs and extend the previously proposed marking-based approach to ad hoc document ranking. We introduce new simple marking strategies to identify which aspect of exact match marking is important for ad hoc document ranking, namely: Does the model require marking both query and document segments or is marking the document enough? Does the model require query-term identification in the marker or is using the same marker for all query terms enough? And which combination works better? We conduct extensive experiments to determine the contribution of exact match marking on the most used PLM, BERT, and the more recent and effective ELECTRA model on standard ad hoc benchmarks. We empirically demonstrate the effectiveness of explicit exact match marking across different experimental scenarios including in-domain, zero-shot transfer and multi-phase fine-tuning settings. Since our approach aims at injecting an established traditional IR cue to recent pretrained transformers, we study the effectiveness of our models with interpolating the traditional BM25 scores. We find that best match scores obtained by BM25 are still valuable since they contribute to the end-to-end effectiveness. Furthermore, the marking-based models require less intervention from BM25 scores to achieve better ranking performance than the vanilla baseline.

Our main contributions can be summarized as follows:

- We present, to our knowledge, the first work investigating the impact of exact match integration into BERT for long document ranking.
- We extend the idea of exact match marking by introducing a new simple and unique marker token for highlighting all the exact term-matches without distinction and explore two marking levels: document and pair marking.
- We conduct extensive experiments to evaluate the effectiveness of our proposed marking strategies on in-domain data using the MS MARCO document ranking benchmark, and

Table 2 Extracts from top ranked passages by MarkedBERT for the query: "causes of left ventricular hypertrophy"

ID	Passage
8332546	*Left* ventricular hypertrophy can occur when...show evidence of *left* ventricular hypertrophy at.
47203	Causes of *Right* Ventricular Hypertrophy. There are four usual causes of *right* ventricular hypertrophy.
6484576	*Left* ventricular hypertrophy is a thickening of the wall of the heart's main pumping chamber.

zero-shot generalizability to out-of-domain data using the standard TREC ad hoc Robust04 and GOV2 benchmarks.

- We investigate the impact of short key word queries vs. long natural descriptions and propose a hybrid pipeline taking advantage of both the retriever and ranker strengths.
- We study the contribution of exact match scores from a bag-of-words model to the out-of-domain effectiveness of our models.
- We study the contribution of multi-phase fine-tuning with additional in-domain fine-tuning to the out-of-domain performance.
- We evaluate the robustness of our approach by considering different PLMs BERT and ELECTRA.
- We compare our best configurations with diverse state-of-the-art approaches.
- We publish our source code as well as our ready-to-use checkpoints at: https://github.com/BOUALILILila/ExactMatchMarking

2 Background and related work

In this paper, we focus on ad hoc document retrieval (also referred to as document ranking) over corpora comprising either news articles or web pages. Following the standard formulation: Given a corpus of documents *C*, potentially large, the task of a ranking system is to produce a ranked list of *k* documents from the corpus in response to a user's information need expressed as query *q*.

2.1 Exact matching in pre-BERT models

Deep Learning approaches have steadily grown in popularity since their introduction in IR over a decade ago. Even though Learning to Rank had reached its zenith early in the 2010s (Liu, 2009; Li, 2011), its use of discrete hand-crafted features, numbering in the hundreds or even more was a major limitation. The promise of Deep Learning models was precisely to obviate the need of such costly manual-engineered features by relying on neural networks and continuous vector representations. Soon, numerous neural ranking models emerged, such as DRMM (Guo et al., 2016), DUET (Mitra et al., 2017), KNRM (Xiong et al., 2017) and Conv-KNRM (Dai et al., 2018). We do not have sufficient space to thoroughly review early neural ranking models and therefore refer the readers to existing overviews (Mitra et al., 2018; Onal et al., 2018). Aside from the models that were specifically designed for document ranking, models from the NLP community built for semantic similarity share some architectural similarities and there has been cross-fertilization between NLP and IR Lin et al. (2020). This interaction lead IR researchers to realise that *relevance matching* and *semantic matching* (e.g: sentence similarity) are different tasks (Guo et al., 2016). While the former requires proper handling of the exact matching signals, the later requires accurately capturing semantics. Thus, neural ranking models required new architecture designs to handle both semantic and exact matching signals. In Mitra et al. (2017), authors proposed a duet architecture composed of two deep neural networks, a *local model* that captures exact matching signals and a *distributed model* for semantic matching. Despite the reported successes of these neural models, there has recently been some skepticism about whether these successes, in the absence of large amounts of data, are not just inflated by comparison to weak baselines. The study conducted over a 100 papers by Yang et al. (2019) on the

Robust04 dataset showed that most models failed against strong non-neural baselines (RM3 (Lavrenko and Croft, 2001)).

2.2 PLMs for multi-stage reranking

Recently, the inception of the transformer architecture (Vaswani et al., 2017) instigated a new wave of approaches (Nogueira and Cho, 2019; MacAvaney et al., 2019; Akkalyoncu Yilmaz et al., 2019) that, at last, were able to significantly outperform well-tuned traditional IR baselines such as RM3 (Lavrenko and Croft, 2001). Nogueira and Cho (Nogueira and Cho, 2019) describe the first successful application of BERT (Devlin et al., 2019) —known as monoBERT— to passage reranking where the ranking task is modeled as a binary classification problem over individual candidate passages. This work marks the beginning of the "BERT revolution". The results of the TREC Deep Learning Track 2019 (Craswell et al., 2020) demonstrated clearly the effectiveness of BERT-based models and revealed a significant distinction with the pre-BERT models. Regardless of its effectiveness, BERT has a key limitation for document ranking: it cannot handle long input sequences that are longer than 512 tokens. In order to address this challenge, (Yang et al., 2019) apply inference on sentences individually, and then use interpolation of the original document score —obtained by a traditional ranker— and the weighted top *n* sentence scores to rerank the documents. Following the same strategy, Birch (Akkalyoncu Yilmaz et al., 2019) reports state-of-the-art effectiveness on the TREC newswire test collections Robust04, Core17 and Core18 using fine-tuned monoBERT on exclusively out of domain passage-level datasets (TREC Microblog, MS MARCO and TREC CAR). Their experiments demonstrate that relevance models can be transferred across different domains, which solves the problem of the lack of passage-level relevance annotations in the target domain. Similarly, (Dai and Callan, 2019) use passage-level evidence to fine-tune BERT by considering all passages from a relevant document as relevant. For inference, the document is split into overlapping passages and each passage is scored individually. Document scores based either on the score of the first passage, the best passages or the sum of all passage scores have been investigated, simple best passage score was found to be the best approach (BERT-MaxP). This was the first work to highlight BERT's capacity to exploit linguistically rich descriptions as opposed to previous keyword search techniques. (MacAvaney et al., 2019) propose a new approach (CEDR) that incorporates the BERT's classification token [CLS] that encodes the representation of the full input into existing pre-BERT neural IR models. The authors show that this joint approach outperforms a vanilla BERT ranker. Instead of aggregating the *scores* of individual passages as in Birch and BERT-MaxP, Parade (Li et al., 2020) aggregates the passage *representations*. This yields an end-to-end differential model like CEDR but without the use of pre-BERT models. In order to obtain the document representation, several aggregation methods were investigated and using a small stack of transformer encoders was found to be the best method. Arguing that exact matching is a valuable cue for ranking, (Boualili et al., 2020) propose a new adaptation of monoBERT, entitled MarkedBERT, that uses a marking technique to highlight exact match signals in the input sequence. The authors demonstrate the effectiveness of MarkedBERT on the MS MARCO passage ranking task and confirmed that marker tokens bring focus on exact matching terms through attention analysis. Beyond BERT, (Nogueira et al., 2020) report new state-of-the-art effectiveness on Robust04 using a novel adaptation of the pretrained sequence-to-sequence model T5 (Raffel et al., 2020) to the document ranking task. This new generation-based approach proved to be more effective than BERT in the data-poor regime with limited training data.

2.3 PLMs for sparse and dense retrieval

The commonly adopted monoBERT approach takes as input the concatenated query document text through BERT and use BERT's [CLS] output token to produce a relevance score. The PLM rerankers compute full cross-attention between contextualized token representations, and thus referred to as cross-encoders. However, their cross-attention operations are too expensive for full collection retrieval. To overcome this challenge, a line of work resorted to augmenting lexical retrieval with PLMs. (Nogueira et al., 2019) propose Doc-T5Query, a document expansion technique for reducing the vocabulary gap between queries and documents. The idea is to train a sequence-to-sequence (T5 (Raffel et al., 2020)) model that, given a text from a corpus, produces queries for which that document might be relevant. Dai and Callan (2019) propose a different framework, DeepCT, for estimating a term's context-specific term importance based on contextual embeddings from BERT. These term importance weights are then mapped into integers so that they can be directly interpreted as term frequencies, replacing term frequencies in a standard bag-of-words inverted index.

Another line of research proposes bi-encorders as an alternative trading off the higher effectiveness of cross-encoders for improved efficiency by encoding the query and document separately. Single-vector systems encode each query and each document into a single dense vector and relevance is modeled as a simple measure of vector similarity (Reimers and Gurevych, 2019; Karpukhin et al., 2020; Xiong et al., 2021). MacAvaney et al. (2020) proposed PreTTR, a hybrid model between bi and cross-encoders by eliminating cross attention on some layers of a cross-encoder model. Luan et al. (2020) raises the limited capacity of single-vector representation to support retrieval of long documents and propose Me-BERT that encodes documents into a set of vectors. Similarly, poly-encoder (Humeau et al., 2020) encodes queries into a set of vectors. Following the same paradigm, ColBERT (Khattab and Zaharia, 2020) represents both queries and documents with token-level vectors and estimates relevance using a late interaction mechanism capturing rich interactions between the two sets of vectors. However, encoding documents with all tokens impose an order-of-magnitude larger index complexity than all previous models.

For an exhaustive review of all research lines using BERT-like models we refer readers to this recent survey (Lin et al., 2020).

2.4 Understanding BERT's success

In the light of the improvements brought by BERT to a wide range of IR tasks, many researchers investigate the reasons behind such substantial improvements. Padigela et al. (2019) empirically study a set of hypotheses that show that BM25 is more biased towards high query term frequency which hurts its performance while BERT retrieves passages with more novel words. However, they found that BERT fails at capturing the query context for long queries. Dai and Callan (2019) demonstrate that unlike traditional IR models, BERT takes advantage of stop words and punctuation thanks to its capacity to model language structure. Qiao et al. (2019) show that BERT is an interaction-based model (Guo et al., 2016), its advantage lies in the cross query-document attentions. Discarding these cross sequence interactions lead to a performance close to random. They also find that BERT assigns extreme matching scores to query-document pairs and most pairs get either one or zero ranking scores, showing it is well tuned by pre-training on large corpora. Câmara and Hauff (2020) analyze BERT using diagnostic datasets built from retrieval heuristics (Rennings et al., 2019). Their experiments show that BERT does not fulfil most retrieval heuristics created by IR experts and argue that these axioms are not suitable to understand BERT performance. MacAvaney et al. (2020) introduce ABNIRML a new framework for analysing the behavior of neural IR models. The authors found that neural ranking models have fundamentally different characteristics from prior ranking models such as high sensitivity to word order and increasing relevance scores when non-relevant content is added to the document.

Our work falls in the category of cross-encoders, and this paper represents, to the best of our knowledge, the first paper detailing with a general approach of highlighting exact matching signals to enhance contextualized pretrained language models such as BERT and reporting an exhaustive set of experiments using long-document ranking benchmarks. Although, using a marking technique to emphasize exact term matches in the query-document pair was first proposed in our own previous work entitled MarkedBERT (Boualili et al., 2020) that represents our initial study. This last was limited to one marking technique on a passage ranking task with a weak training regime, raising the question of its full potential (Lin et al., 2020). Aside from MarkedBERT, marking techniques were mentioned in the descriptions of our TREC-COVID challenge (Voorhees et al., 2021) submissions. This present work is a generalisation of the approach followed in MarkedBERT where we present a complete description of our ideas, comprehensive evaluation on in and out of domain TREC ad hoc benchmarks with a better training regime, making it directly comparable to state-of-the-art models.

3 Augmenting pretrained contextualized language models with exact match signals

In this section, we first describe the general architecture we adopt in this work and then present the marking strategies we propose to explicitly highlight exact term matches in the query-document pairs before feeding them to the BERT model. We consider the traditional formulation of exact matching where two terms t_1 and t_2 match exactly if their stems are identical. We use the Porter algorithm for stemming and stop words are not considered during marking. By adding explicit indications of exact matching signals in the textual inputs, the models can benefit from this traditional hint and adapt better to the ad hoc task.

3.1 Model architecture

We adopt the model configuration described by Nogueira and Cho (2019) referred to as monoBERT or vanilla BERT. In this configuration, BERT is applied as a binary relevance classifier for text ranking. The architecture of the model is shown in Fig. 1. Using the same notation as Devlin et al. (2019), the query q is fed as *Segment A* and the candidate document d as *Segment B*. The special token [CLS] is prepended to the input sequence, and the special delimiter token [SEP] is placed at the beginning and end of the document segment to build the input sequence S as follows:

$$S = [[CLS], Q, [SEP], D, [SEP]] \tag{1}$$

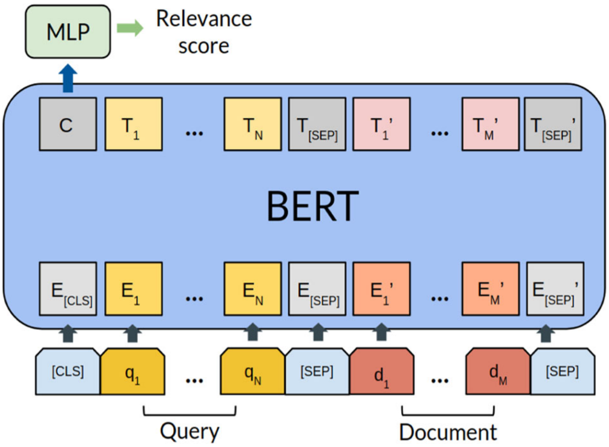


Fig. 1 BERT sentence pair classification architecture Devlin et al. (2019) used in vanilla BERT | monoBERT (Nogueira and Cho, 2019)

where Q and D represent the sequences of tokens obtained after applying the WordPiece tokenizer to the query q and document d texts, respectively.

Once the sequence S is passed through BERT, the final vector representation C of the standard classification token [CLS], that captures the interaction between the query and the document, is used as input to a single layer neural network that estimates a score $R(d, q)$ quantifying how relevant the candidate document d is to the query q. That is:

$$R(d,q) = P(Relevant = 1|q,d) \tag{2}$$

The details of the fine-tuning and inference process are given in Sect. 4.

3.2 Exact match marking

We propose different marking strategies that only intervene at the textual input level to augment the input sequence S defined in Eq. (1). Instead of altering the model's architecture in order to integrate the desired traditional signal, we prefer letting the model learn how to use the given hints and avoid the risk of introducing a systematic bias towards exact term matching. A marking strategy is defined by two(2) parameters:

1. *Marker-Token type*: we introduce two types of marker tokens, namely: *Simple Markers* and *Precise Markers*.
2. *Marking level*: we investigate two levels for marking: *Document Marking* and *Pair Marking*.

Table 3 illustrates all the four(4) marking strategies that can be defined using the two marker-token types at the two different marking levels. Note that the Pre-Pair marking strategy corresponds to the strategy used in the MarkedBERT model (Boualili et al., 2020).

3.2.1 Marker-token type

We investigate two types of marker tokens to investigate whether distinguishing query terms is important or not to the model performance.

Table 3 Example of the proposed marking strategies applied to the query Q: "causes of left ventricular hypertrophy", and the document D: "Left ventricular hypertrophy can occur when some factor ..."

Marker	Level	Strategy	Marked input sequence
Simple	Document	Sim-Doc	Q: causes of left ventricular hypertrophy D: #Left# #ventricular# #hypertrophy# can occur.
	Pair	Sim-Pair	Q: causes of #left# #ventricular# #hypertrophy# D: #Left# #ventricular# #hypertrophy# can occur.
Precise	Document	Pre-Doc	Q: causes of left ventricular hypertrophy D: $[e_2]$Left$[/e_2]$ $[e_3]$ventricular$[/e_3]$ $[e_4]$hypertrophy$[/e_4]$ can occur.
	Pair	Pre-Pair	Q: causes of $[e_2]$left$[/e_2]$ $[e_3]$ventricular$[/e_3]$ $[e_4]$hypertrophy$[/e_4]$ D: $[e_2]$Left$[/e_2]$ $[e_3]$ventricular$[/e_3]$ $[e_4]$hypertrophy$[/e_4]$ can occur.

Simple Markers. Uses a simple unique marker (#) for all query terms without explicit distinction. Considering a query $Q = \{q_1, \ldots, q_{|Q|}\}$, whose terms q_n and q_m, with $1 < n < m < |Q|$, occur in the document and thus have to be marked, we obtain the new marked query segment $\tilde{Q}$ as follows:

$$\tilde{Q} = \{q_1, \ldots, \#q_n\#, \ldots, \#q_m\#, \ldots, q_{|Q|}\} \tag{3}$$

Precise Markers. Uses precise markers consisting of newly introduced tokens $[e_k]$ and $[/e_k]$, where $k = \{1, ..., |Q|\}$ identify query terms, that mark the start and the end of each matched term, respectively. This marking technique associates each unique query-term q_k with a unique pair of marker tokens $[e_k]$ and $[/e_k]$ that identifies it and its occurrences. If a term is repeated in the query, all occurrences of this query term will be highlighted using the same identifier i.e that of the first occurrence. For example, the query Q described in the previously paragraph with simple markers would be marked as follows:

$$\tilde{Q} = \{q_1, \ldots, [e_n]q_n[/e_n], \ldots, [e_m]q_m[/e_m], \ldots, q_{|Q|}\} \tag{4}$$

3.2.2 Marking level

In order to better understand whether it is relevant to mark both the query and the document segments or the document segment only, we investigate two marking levels: Document and Pair marking. In the former, the occurrences of query terms in the document are marked in the document segment while in the later, the exact matching terms are marked in both the document and query segments as shown in Table 3. We use the same notations defined in the model's architecture where Q refers to the query segment and D refers to the document segment that constitute the input sequence S.

Document marking. It only augments the document segment D with marker tokens indicating the start and the end of each query-term occurrences in the document. Considering a query $Q = \{q_1, \ldots, q_{|Q|}\}$ and a document $D = \{d_1, \ldots, d_{|D|}\}$, if $\{d_i, d_j\}$ are occurrences of query term q_n and d_l is the only occurrence of q_m in D with $1 < n < m < |Q|$ and $1 < i < j < l < |D|$, the augmented query and document sequences $\tilde{Q}$ and $\tilde{D}$, respectively, are as follows when using the simple markers:

$$\tilde{Q} = \{q_1, \ldots, q_n, \ldots, q_m, \ldots, q_{|Q|}\}$$
$$\tilde{D} = \{d_1, \ldots, \#d_i\#, \ldots, \#d_j\#, \ldots, \#d_l\#, \ldots, d_{|D|}\}$$

and as follows when using the precise markers:

$$\tilde{Q} = \{q_1, \ldots, q_n, \ldots, q_m, \ldots, q_{|Q|}\}$$
$$\tilde{D} = \{d_1, \ldots, [e_n]d_i[/e_n], \ldots, [e_n]d_j[/e_n], \ldots, [e_m]d_l[/e_m], \ldots, d_{|D|}\}$$

Pair marking. It augments both the query and document sequences with marker tokens indicating the start and the end of each exact matched term between the query and the document. In our experiments, a query term with no occurrences in the document is not marked. Considering the same example as in the *Document marking* level, the augmented query and document sequences $\tilde{Q}$ and $\tilde{D}$, respectively, are as follows when using the simple markers:

$$\tilde{Q} = \{q_1, \ldots, \#q_n\#, \ldots, \#q_m\#, \ldots, q_{|Q|}\}$$
$$\tilde{D} = \{d_1, \ldots, \#d_i\#, \ldots, \#d_j\#, \ldots, \#d_l\#, \ldots, d_{|D|}\}$$

and as follows when using the precise markers:

$$\tilde{Q} = \{q_1, \ldots, [e_n]q_n[/e_n], \ldots, [e_m]q_m[/e_m], \ldots, q_{|Q|}\}$$
$$\tilde{D} = \{d_1, \ldots, [e_n]d_i[/e_n], \ldots, [e_n]d_j[/e_n], \ldots, [e_m]d_l[/e_m], \ldots, d_{|D|}\}$$

4 Experimental setup

This section describes the experimental setup used for studying the effectiveness of our models for document ranking. We present the detailed fine-tuning our models on the large-scale MS MARCO passage dataset and describe the MS MARCO document ranking benchmark used for in-domain evaluations, and the standard TREC Robust04 and GOV2 benchmarks used for studying the out-of-domain transfer capabilities of our models. We further describe the inference process and the diverse state-of-the-art baselines we use to comparatively evaluate our approach. We report results using the official metrics of each collection, namely: nDCG@10 and MAP@100 for MS MARCO document ranking collection in the context of TREC Deep Learning 2019 and 2020 tracks, and nDCG@20 and P@20 for Robust04 and GOV2, enabling thus direct comparisons with previous work.

4.1 Datasets

We conduct experiments on two standard ad hoc benchmarks: Robust04 and GOV2. In addition to these traditional benchmarks, we use the recent TREC Deep Learning (DL) Document Ranking benchmark from 2019 and 2020 tracks. *Robust04*[1] is a news wire collection comprising 500K documents (TREC Disks 4 and 5) and 249 judged topics. Each topic is composed of three fields: The "title" is a short keyword query, the "description" is a longer well-formed natural language sentence that describes the information need and the "narrative" is a paragraph that provides guidance for relevance assessment. Table 4 provides

[1] https://trec.nist.gov/data/robust/04.guidelines.html

Table 4 Example of Robust04 search topic: Topic 302

Title	Poliomyelitis and Post-Polio
Description	Is the disease of Poliomyelitis (polio) under control in the world?
Narrative	Relevant documents should contain data or outbreaks of the polio disease (large or small scale), medical protection against the disease, reports on what has been labeled as post-polio problems. Of interest would be location of the cases, how severe, as well as what is being done in the “post-polio” area.

Table 5 Benchmarks statistics

Benchmark	# Judged topics	# Documents	# Words per document
Robust04	249	0.5M	0.470K
GOV2	149	25M	0.835K
MS MARCO Document	43/45	3.2M	1.123K

The MS MARCO document dataset has 43 judged topics in DL 2019 and 45 judged topics in DL 2020

an example of a TREC Robust04 topic. *GOV2*[2] is a Web collection crawled from government Websites in early 2004 comprising 25M documents and only 149 topics in the same format as Robust04 topics with title, description and narrative. Documents in the GOV2 corpus are on average much longer than those in the Robust04 corpus; see Table 5. *MS MARCO Document Ranking dataset* is a benchmark for web search used in TREC DL 2019-2020 tracks (Craswell et al., 2020, 2021). The dataset contains more than 3M documents composed of three fields: title, URL and body. Dense NIST judgments are provided for 43 and 45 topics for DL 2019 and 2020, respectively.

Table 5 resumes some statistics on the evaluation benchmarks.

4.2 Baselines

We compare our models against diverse baselines including: Traditional non-neural approaches also known as Lexical Retrieval methods, sparse retrieval approaches, dense retrieval models (bi-encoders), and strong reranking models (cross-encoders).

4.2.1 Lexical retrieval baselines

- BM25, we use the Anserini (Yang et al., 2017) implementation with default parameters. For description queries, we set $k_1 = 0.9$ for Robust04 and $k_1 = 2.0$ for GOV2 and $b = 0.6$ for both datasets. This unsupervised model serves both as a baseline and as the first stage retriever in all our experiments.
- BM25+RM3, a query expansion model based on RM3 (Lavrenko and Croft, 2001) considered as a strong non-neural baseline. We use the Anserini (Yang et al., 2017) implementation with the default parameters. For description queries, we use 20 expansion terms following (Li et al., 2020).

[2] http://ir.dcs.gla.ac.uk/test_collections/gov2-summary.htm

4.2.2 Sparse retrieval baselines

- DeepCT (Dai and Callan, 2020), we report results on Robust04 and GOV2 obtained using the BOW+DeepCT-Query model (Dai and Callan, 2019), and use the re-weighted MS MARCO documents provided by the authors[3] using the HDCT model (Dai and Callan, 2020) in combination with Anserini's BM25 with default parameters for TREC DL 2019 and 2020 evaluations.
- DocT5Query (Nogueira et al., 2019), following the paper setup, we generate 40 expansion queries per document and use Anserini's BM25 with default parameters. Due to the large size of the GOV2 collection (see Table 5) and the high computational cost of DocT5query we do not report results on this collection.

4.2.3 Dense retrieval baselines

- DPR (Karpukhin et al., 2020), we use DPR as a retriever with the open source implementation from the Transformers library (Wolf et al., 2020) and the publicly released DPR checkpoints for Query[4] and Context[5] encoders.
- ANCE (Xiong et al., 2021), we use ANCE as a retriever and use the Sentence Transformers library (Reimers and Gurevych, 2019) with the publicly released checkpoint[6].
- ColBERT (Khattab and Zaharia, 2020), we use ColBERT as a dense retriever using the authors released code: after encoding the whole collection, we use the top-1000 documents retrieved using ANN with faiss (Johnson et al., 2017) and rerank them using ColBERT late-interaction operation. Considering the size of the GOV2 collection ($25M$ documents), and the important space footprint of ColBERT indexes[7], we could not produce results on GOV2.

4.2.4 Reranking baselines

- Vanilla baseline, the vanilla monoBERT model is our main baseline since it represents the core model we augment with explicit exact match cues in our proposed models. The vanilla baseline as well as our models share the same configuration and evaluation setup making it suitable for evaluating the impact of exact match marking.
- Birch (MS) and Birch (MS-MB) (Akkalyoncu Yilmaz et al., 2019), the notation in parentheses indicate the fine-tuning dataset(s): Ms for MS MARCO and MS-MB refers to the model fine-tuned first on MS MARCO and then further fine-tuned on Microblog (MB) data. We use the results reported by Li et al. (2020) that uses BM25 instead of BM25+RM3 as the first-stage retriever.

[3] http://boston.lti.cs.cmu.edu/appendices/TheWebConf2020-Zhuyun-Dai/rankings/

[4] https://huggingface.co/sentence-transformers/facebook-dpr-question_encoder-multiset-base

[5] https://huggingface.co/sentence-transformers/facebook-dpr-ctx_encoder-multiset-base

[6] https://huggingface.co/sentence-transformers/msmarco-roberta-base-ance-firstp

[7] With less than $4M$ documents, the size of the MS MARCO Document index was already as big as 200GB.

- BERT-MaxP (MS) (Dai and Callan, 2019), we report the results obtained with the re-implementation by Li et al. (2020) where the results are improved using a BERT model fine-tuned on MS MARCO rather than Bing search log.
- Parade (Li et al., 2020), we report results obtained using both BERT and ELECTRA variant from the paper.
- T5 (Nogueira et al., 2020), the T5, also known as monoT5, with $3B$ parameters detains the state-of-the-art across many ad hoc benchmarks like Robust04. We report the original results from the paper.

4.3 Training

We use the base version (12 layers, 768 hidden size, 12 heads, 110M parameters) of BERT due to hardware limitations. We fine-tune both our vanilla baseline and our models augmented with the different marking strategies on the large publicly released MS MARCO passage dataset. We use a batch-size of 128 and the maximum sequence length ($128\ sequences \times 512\ tokens = 65536\ tokens/batch$) for $100k$ on free Google Colab TPUs[8]. We use Adam optimizer (Kingma and Ba, 2015) with the initial learning rate set to $3e^{-6}$ and linear decay of the learning rate. The drop out rate is set to 0.1 for all our experiments. We use the open source implementation of BERT by Hugging Face (Wolf et al., 2020). It is important to note that fine-tuning an augmented model with a marking strategy does not add a computational cost compared to the vanilla model.

4.4 Inference

We use a two-stage ranking pipeline. We retrieve an initial candidate list of top 1, 000 documents per query using BM25. We use the BM25 implementation from off-the-shelf Anserini open-source IR toolkit (Yang et al., 2017).

The length of BERT's input sequence cannot exceed 512 tokens due to the fact that the positional embeddings were trained on sequences of a maximum length of 512 tokens. This limitation prevents from directly applying our models to long documents. Following the strategy proposed by Dai and Callan (2019), we split each document into overlapping passages that can be handled individually by BERT. For Robust04 and GOV2, passages are generated using a sliding window of 150 words and a stride of 75 words, formally expressed as $d = \{p_1, ..., p_n\}$ where n is the number of passages in the document d. As a trade-off between latency and effectiveness, we only consider a maximum of 30 passages per document. The first and last passages are always picked while the remaining 28 are randomly chosen. The fine-tuned BERT models on exclusively out-of-domain data are used afterwards to predict the relevance of each passage w.r.t a query q independently. The best scoring passage is then taken as a proxy for the Document-level relevance:

$$R(d, q) = max(R(p_1, q), ..., R(p_n, q)) \tag{5}$$

For the queries we consider both the topic titles that are preferred by most pre-BERT models including BM25, and the descriptions that are more similar to MS MARCO's natural language questions.

[8] https://colab.research.google.com

For TREC DL Document ranking evaluation, we split each document into overlapping passages with the same maximum length of 384 and a stride of 192 following the splitting strategy in Yan et al. (2019). In addition, the title is added to the beginning of every passage if it is available. Similarly to Robust04 and GOV2, we use the best scoring passage as proxy for the whole document relevance.

5 Results and analysis

We address, in this section, our research questions. First, we investigate the effectiveness of our proposed exact match marking strategies with a BERT core on in-domain data, i.e, MS MARCO document ranking benchmark, and the robustness to out-of-domain collections, i.e, Robust04 and GOV2. Then, we study how to improve domain-transfer capabilities of our models using score interpolation with a bag-of-words model. We further investigate the contribution of additional fine-tuning on limited target-domain data in a multi-phase fine-tuning setting and how our exact match marking contributes in each phase. Finally, we verify the contribution of our exact match marking on the more effective ELECTRA model, and compare our best configurations to diverse state-of-the-art baselines.

5.1 Performance of the models augmented with exact match marking

We evaluate the contribution of our proposed exact match marking strategies and discuss our research question *RQ1 Is exact match marking beneficial to pretrained transformers exemplified by BERT?* by comparing the augmented models with exact match marking to the vanilla baseline. We consider results in the in-domain setting with MS MARCO Document dataset and the zero-shot transfer setting to out-of-domain datasets, namely: Robust04 and GOV2.

Table 6 Reranking effectiveness on the TREC DL 2019 and DL 2020 Document ranking tasks

TREC DL Doc	DL 2019				DL 2020			
Model	nDCG@10		MAP@100		nDCG@10		MAP@100	
BM25	0.5176	–	0.2434	-	0.5286	–	0.3793	–
BM25+RM3	0.5169	–	0.2772	-	0.5248	–	0.4006	–
Vanilla$_{\text{BERT}}$	0.6726	–	0.3006	-	0.6340	–	**0.4523**	–
Sim-Doc$_{\text{BERT}}$	0.6858	+2.0%	0.3038	+1.1%	0.6340	+0.0%	0.4414	−2.4%
Sim-Pair$_{\text{BERT}}$	0.6798	+1.1%	0.3057	+1.7%	0.6495	+2.4%	0.4505	−0.4%
Pre-Doc$_{\text{BERT}}$	0.6777	+0.8%	**0.3061**	+1.8%	0.6368	+0.4%	0.4513	−0.2%
Pre-Pair$_{\text{BERT}}$	**0.7025** †	+4.4%	0.3018	+1.8%	**0.6498**	+2.5%	0.4497	−0.6%

Best performances are highlighted in bold

Significant improvements over the vanilla baseline with $p < 0.05$ are indicated with †

Change rate over the vanilla baseline are reported for each metric (%)

5.1.1 In-domain effectiveness

We re-rank the initial list of candidate documents retrieved by BM25 with RM3 query expansion, using all our models and the vanilla baseline. We report the performance on the TREC DL 2019 and 2020 test sets in Table 6, in terms of the official evaluation metrics: nDCG@10 and MAP@100.

Comparison with baselines. Compared to BM25 and the first-stage retriever (BM25+ RM3), all BERT-based models perform significantly better. Interestingly, the non-neural methods perform better on DL 2020 test set while the BERT-based models perform better on the DL 2019 test set. Adding exact match marking regardless of the marking strategy, leads to better or at least the same performance as the vanilla baseline (marking ablation). The Pre-Pair$_{\text{BERT}}$ model achieves the overall best performance on DL 2019 test topics, but also on DL 2020 along with Sim-Pair $_{\text{BERT}}$.

Impact of the marker type and marking level on the performance. On TREC DL 2019, using the pair marking strategy brings substantial gains in performance when used in combination with the precise marker type, Pre-Pair $_{\text{BERT}}$ achieves +3.7% relative gain over the Pre-Doc $_{\text{BERT}}$ model. While it leads to a drop in performance when combined with the simple marker, Sim-Pair $_{\text{BERT}}$ has a relative loss of −0.9% compared to Sim-Doc $_{\text{BERT}}$. Interestingly, on TREC DL 2020 using the Pair marking level has the same impact regardless of the marker type.

Marking both the query and document segments seems to be more beneficial considering results on both test collections. Using the precise marker type brings further gains in performance on DL 2019.

5.1.2 Out-of-domain effectiveness

We use the fine-tuned models on exclusively MS MARCO passages to rerank the documents retrieved by BM25 in the first-stage. We do not train the models on the target collections (Robust04, GOV2), we use all their queries and relevance judgements as a held-out test set. Thus, this evaluation is an instance of a zero-shot transfer setting.

Table 7 shows the reranking effectiveness of our different models and baselines on the top 1, 000 candidate documents retrieved by BM25 from Robust04 and GOV2 collections using both the title and description fields of the TREC topics. We recall that titles are short key word queries preferred by traditional bag-of-words models like BM25 and descriptions are well-written natural language queries similar to MS MARCO's questions on which the BERT models are fine-tuned. We report results using the commonly used nDCG@20 and P@20 metrics to enable direct comparisons with previous work on these collections.

Comparison with baselines. All BERT-based models achieve substantially better performance on both collection compared to the traditional non-neural baselines at the only exception of GOV2 titles. We observe a discrepancy in the impact of the exact match marking on GOV2 compared to Robust04. While all our models, except Sim-Doc $_{\text{BERT}}$, significantly outperform the vanilla baseline on Robust04 descriptions or at least achieve similar performance on titles, our models have no significant impact on GOV2. Importantly, in no case a marking-based model leads to a significant degradation of performance on GOV2. The disparity in the behavior of the models on the two benchmarks is probably due to the nature of the documents involved. While Robust04 comprises well-written news articles, GOV2 documents are web pages that include navigation bars, advertisements, tables and discontinuous text. The zero-shot domain transfer –from the MS MARCO fine-

Table 7 Reranking effectiveness in the zero-shot transfer setting of the different models on Robust04 and GOV2 collections

Robust04	Title run				Description run			
Model	nDCG@20		P@20		nDCG@20		P@20	
BM25	0.4240	–	0.3631	–	0.4058	–	0.3345	–
BM25+RM3	0.4407	–	0.3821	–	0.4255	–	0.3661	–
Vanilla $_{\text{BERT}}$	0.4652	–	0.4046	–	0.4510	–	0.3851	–
Sim-Doc $_{\text{BERT}}$	0.4447*	−4.4%	0.3831*	−5.3%	0.4166*	−7.6%	0.3510*	−8.9%
Sim-Pair $_{\text{BERT}}$	**0.4773**	+2.6%	**0.4155**	+2.7%	**0.4931** ‡	+9.3%	**0.4169** ‡	+8.3%
Pre-Doc $_{\text{BERT}}$	0.4767	+2.5%	0.4084	+0.9%	0.4789‡	+6.2%	0.4026‡	+4.5%
Pre-Pair $_{\text{BERT}}$	0.4654	+0.0%	0.4024	−0.5%	0.4795‡	+6.3%	0.4034‡	+4.8%
GOV2	Title run				Description run			
Model	nDCG@20		P@20		nDCG@20		P@20	
BM25	0.4774	–	0.5362	–	0.4264	–	0.4705	–
BM25+RM3	**0.4851**	–	**0.5634**	–	0.4212	–	0.4966	–
Vanilla $_{\text{BERT}}$	0.4533	–	0.5272	–	0.4696	–	0.5248	–
Sim-Doc $_{\text{BERT}}$	0.4588	+1.2%	0.5349	+1.5%	0.4686	−0.2%	0.5262	+0.3%
Sim-Pair $_{\text{BERT}}$	0.4468	−1.4%	0.5134	−2.6%	0.4687	−0.2%	0.5326	+1.5%
Pre-Doc $_{\text{BERT}}$	0.4485	−1.1%	0.5121	−2.9%	**0.4768**	+1.5%	**0.5315**	+1.3%
Pre-Pair $_{\text{BERT}}$	0.4515	−0.4%	0.5238	−0.6%	0.4752	+1.2%	0.5285	+0.7%

Best performances are highlighted in bold

Significant improvements over the vanilla baseline with $p<0.05$ and $p<0.01$ after Bonferroni correction are indicated with † and ‡ respectively

Significant inferiority with $p<0.05$ is marked with $*$

For each measure, the improvement rate over the vanilla baseline is given (%)

tuned models to Robust04 articles– seems to be more attainable than to GOV2 web pages even though MS MARCO passages were extracted from the web. We hypothesise that further fine-tuning on domain-specific data may be required to learn better domain-specific text representations. We investigate this in-domain adaptation in Sect. 5.3.

Impact of the marker type and marking level on the performance. On Robust04, marking both the query and the document –models based on pair marking– has more impact on the simple marker than the precise marker. On the description queries, Sim-Pair $_{\text{BERT}}$ achieves an nDCG@20 of 0.4931 while Sim-Doc $_{\text{BERT}}$ has an nDCG@20 of only 0.4166, and achieve 0.4773 compared to 0.4447, respectively, on title queries. While the marking strategy has a lower impact on models using the precise markers (Pre-Doc $_{\text{BERT}}$ and Pre-Pair $_{\text{BERT}}$) especially on descriptions. On the other hand, results on the GOV2 collection are quite mitigated.

Marking both the query and the document segments with a simple marker (#) appears to be the best setting, Sim-Pair $_{\text{BERT}}$ has the best ranking accuracy among the four strategies tested, with clear margins on the Robust04 collection especially on descriptions. We, thus, choose to continue our analysis using the Sim-Pair $_{\text{BERT}}$ strategy, the full results using all the marking strategies can be found in Appendix 1.

Title versus description queries. Since we are in a reranking configuration, it is important to note that the first stage retriever BM25, as most pre-BERT ranking models, prefers short key word queries to longer natural language descriptions (Dai and Callan, 2019; Nogueira et al., 2020). Table 8 shows the recall at rank 1, 000 of BM25 for both title and description queries, where we notice a substantial difference in recall affecting the quality of the candidate documents that the reranking models receive. Despite this disadvantageous initialization, the reranking models manage to reduce the gap between title and description runs. The improvement rate over BM25 is much higher for description queries compared to title queries on both collections especially on GOV2 where vanilla $_{\text{BERT}}$ has a change rate of −5.0% over BM25, while it achieves over +10% gain on descriptions. The descriptions that are longer natural language queries carrying richer information that could not be fully harnessed by the traditional bag-of-words method, are more effectively leveraged in the reranking stage. This BERT ability was already noted in previous work (Dai and Callan, 2019), and Sim-Pair $_{\text{BERT}}$ follows the same preference, as it improves the search accuracy of the description runs more effectively than the title runs. The overall performance reported for our model using descriptions clearly surpasses that obtained using titles by +4.1% on average, despite the lower recall in the initial stage.

Impact of the initial stage retriever. Considering that first stage ranker BM25 has higher recall on title queries, and that the marking-based models prefer description queries, we propose a hybrid reranking pipeline where the documents retrieved by BM25 using title queries are reranked with the BERT-based models using the description queries. Using this hybrid pipeline allow as to obtain a higher recall in the first stage since BM25 performs better on short keyword queries, and thus better candidate documents for reranking. Description queries are longer statements of information needs more suitable for pretrained reranking models to fulfill their potential. This pipeline remains realistic as language queries may be generated from standard key-word queries (Padaki et al., 2020). This hybrid approach is also adopted in recent state-of-the-art ranking model based on T5 Nogueira et al. (2020).

Table 9 shows the results obtained using the hybrid reranking pipeline on both test collections. Unsurprisingly, using better candidate documents for reranking with descriptions yields even better accuracy. The vanilla $_{\text{BERT}}$ model achieves an improvement rate of +14% over BM25 on Robust04 and +3.4% on GOV2 (we recall that BM25 results are obtained using titles). Adding exact match marking in the hybrid reranking pipeline outperforms the vanilla baseline on both collections; significantly on Robust04 with a gain of over +8%.

5.1.3 In-domain versus out-of-domain effectiveness.

Results on both in-domain and out-of-domain benchmarks clearly indicate that exact match marking, aside from the Sim-Doc marking strategy which significantly underperforms the

Table 8 Recall of BM25 on Robsut04 and GOV2 collections on both title and description queries

Collection	Title	Description
Robust04	0.6989	0.6519
GOV2	0.7106	0.6024

Table 9 Reranking effectiveness in the zero-shot transfer setting of the different models on Robust04 and GOV2 collections using the hybrid pipeline

	Robust04				GOV2			
Model	nDCG@20		P@20		nDCG@20		P@20	
BM25	0.4240	–	0.3631	–	0.4774	–	0.5362	–
BM25+RM3	0.4407	–	0.3821	–	0.4851	–	0.5634	–
$\text{Vanilla}_{\text{BERT}}$	0.4845	–	0.4147	–	0.4937	–	0.5611	–
$\text{Sim-Pair}_{\text{BERT}}$	**0.5239** ‡	+8.1%	**0.4446** ‡	+7.2%	**0.4991**	+1.1%	**0.5695**	+1.5%

Best performances are highlighted in bold

Significant improvements over the vanilla baseline with $p < 0.05$ and $p < 0.01$ after Bonferroni correction are indicated with † and ‡ respectively

For each measure, the improvement rate over the vanilla baseline is given (%)

vanilla baseline on Robust04, is more beneficial than using a vanilla baseline. Using Sim-Pair (especially for out-of-domain experiments) or Pre-Pair (especially for in-domain experiments) marking strategies seems to be working best.

In the next two sections, we focus on out-of-domain effectiveness and study common techniques used in the literature to enhance the effectiveness of BERT-based models, and how our models behave in combination with these techniques. Therefore, the MS MARCO document ranking benchmark is not suitable and thus we only report results on Robust04 and GOV2 collections.

5.2 Contribution of the first-stage retriever scores to the end-to-end effectiveness

Our experimental design is based on a two-stage ranking architecture also known as a retrieve-then-rerank architecture where our BERT-based models rerank the documents retrieved by the BM25 model. In this section we evaluate the contribution of the best match scores from the initial bag-of-words retriever to the end-to-end effectiveness by simply combining BM25's document-level scores with the passage-level evidence from the reranker using linear interpolation. We follow the linear combination defined in the Birch model (Akkalyoncu Yilmaz et al., 2019).

Birch uses a monoBERT sentence-level relevance classifier at its core. To determine document relevance s_f, inference is applied over each individual sentence s_i in a candidate document d, and then the top n sentence scores are combined with the original document score s_{doc} given by the first-stage retriever as follows:

$$s_f = \alpha . s_{doc} + (1 - \alpha) . \sum_{i=1}^{n} w_i . s_i \qquad (6)$$

where s_i is the i-th top scoring sentence according to monoBERT. The parameters α and w_i's are tuned via cross-validation. In other words, the relevance score of a document comes from the combination of its document-level term-matching score and evidence contributions from the top sentences in the documents as determined by monoBERT.

For our experiments, the linear interpolation is applied to the results obtained in the zero-shot transfer setting with the best-scoring passage ($n = 1$). In other words, we use the score combination defined in Eq. (6) on the document scores obtained by the BM25 retriever at cutoff $1,000$ and their corresponding scores estimated with the best-scoring passage method by the reranking models. Table 10 first shows the results of the traditional BM25 model alone, then the second and third sections are each dedicated to a reranker: vanilla and Sim-Pair BERT models. For both rerankers, we remind the results of the model alone obtained in the zero-shot transfer setting and then present the end-to-end effectiveness after interpolating BM25 scores (+ BM25) with the indication of the change rate (%) over the reranker-only effectiveness. These results allow us to answer our research question *RQ2 Do exact match scores from the first-stage retriever contribute to end-to-end effectiveness of the pretrained transformers and how exact match marking affects this contribution?*

5.2.1 Impact of interpolating BM25 scores

Interpolating BM25 scores (Best Match) that are solely based on surface-level features such as TF and IDF leads to a significant gain in performance, indicating that BM25 document-level scores provide an additional relevance signal that the BERT-based models alone could not effectively capture. We notice that the improvement rate resulting from interpolating BM25 scores is much substantial on the GOV2 collection (+15% in average) compared to Robust04 (+5.7% in average). The fact that the BERT models outperform BM25 by a large margin on Robust04 while this margin is much smaller on the GOV2 can explain why BM25 scores have more incidence on the end-to-end effectiveness on GOV2 than on Robust04.

5.2.2 Impact of exact match marking

From Table 10, we can clearly see that for Robust04, where the exact match marking is effective, the improvement rate over the reranker-only effectiveness is lower when using exact match marking, about +12% in average, compared to the vanilla model with +22% gain in average. In other words, the impact of the BM25 scores is more important on the vanilla model compared to the Sim-Pair model. While on GOV2 the improvement rate after BM25 scores interpolation compared to the reranker-only performance is either comparable or slightly higher when using exact match marking compared to the vanilla baseline. However, the performance of the Sim-Pair $_{\mathrm{BERT}}$ model with BM25 scores interpolation is, in all cases, higher than the vanilla $_{\mathrm{BERT}}$ + BM25 performance regardless of the improvement rate brought by the score combination. Since we use the results obtained in the zero-shot domain transfer setting where, we recall, the exact the marking is more effective, the gains of the Sim-Pair $_{\mathrm{BERT}}$+BM25 configuration over the vanilla $_{\mathrm{BERT}}$+BM25 are more substantial on Robust04 than on GOV2.

5.2.3 Contribution of BM25 scores

The contribution of BM25 scores is controlled by the parameter α in Eq. (6) which we tuned via 5-fold in-collection cross validation. In all scenarios, the weight put on α is non-negligible, in other words, the contribution of BM25 signals remain important, this observation was also reported for the Birch model (Lin et al., 2020). However, we notice that the weight of α is always lighter when combining with the Sim-Pair $_{\mathrm{BERT}}$ model that uses exact match marking. For Robust04 descriptions, the vanilla $_{\mathrm{BERT}}$+BM25 baseline puts

Table 10 Reranking effectiveness of the different models before and after interpolating BM25 scores on Robust04 and GOV2 collections

Robust04	Title run				Description run				Hybrid run			
Model	nDCG@20		P@20		nDCG@20		P@20		nDCG@20		P@20	
BM25	0.4240	–	0.3631	–	0.4058	–	0.3345	–	0.4240	–	0.3631	–
Vanilla $_{\mathrm{BERT}}$	0.4652	–	0.4046	–	0.4510	–	0.3851	–	0.4845	–	0.4147	–
+ BM25	0.4932	+6.0%	0.4255	+5.2%	0.4856	+7.7%	0.4062	+5.5%	0.5266	+8.7%	0.4488	+8.2%
Sim–Pair $_{\mathrm{BERT}}$	0.4773	–	0.4155	–	0.4931	–	0.4169	–	0.5239	–	0.4446	–
+ BM25	**0.4947**	+3.6%	**0.4265**	+2.6%	**0.5098**	+3.4%	**0.4279**	+2.6%	**0.5497**	+4.9%	**0.4707**	+5.9%
GOV2	Title run				Description run				Hybrid run			
Model	nDCG@20		P@20		nDCG@20		P@20		nDCG@20		P@20	
BM25	0.4774	–	0.5362	–	0.4264	–	0.4705	–	0.4774	–	0.5362	–
Vanilla $_{\mathrm{BERT}}$	0.4533	–	0.5272	–	0.4696	–	0.5248	–	0.4937	–	0.5611	–
+ BM25	0.5320	+17.%	0.5987	+13.	0.5166	+10.%	0.5742	+9.4%	0.5722	+16.%	0.6383	+14%
Sim-Pair $_{\mathrm{BERT}}$	0.4468	–	0.5134	–	0.4687	–	0.5326	–	0.4991	–	0.5695	
+ BM25	**0.5327**	+19.%	**0.6000**	+17%	**0.5235**	+12.%	**0.5893**	+11.%	**0.5778**	+16.%	**0.6497**	+14.%

Best performances are highlighted in bold

For each measure, the improvement rate over the reranking performance without BM25 scores interpolation is given (%)

a weight of $\alpha \in \{0.3, 0.4\}$ on BM25 scores, when Sim-Pair $_{\mathrm{BERT}}$+BM25 only consider a contribution of $\alpha = 0.2$ from BM25, while achieving substantially better performance. This indicates that the vanilla model relies more on BM25 to complete its relevance estimation unlike the marking-based model that is able to effectively capture more relevance signals and thus needing less contribution from BM25 scores.

Figure 2 visualizes the end-to-end ranking accuracy measured by nDCG@20 for $\alpha \in [0, 1]$ on both Robust04 and GOV2 collections. On Robust04, we can clearly see that Sim-Pair $_{\mathrm{BERT}}$+BM25 reaches the most effective combination with smaller contribution from BM25 scores (smaller α), while the vanilla baseline requires more intervention from BM25 and still cannot reach the performance of Sim-Pair $_{\mathrm{BERT}}$+BM25, especially on descriptions. It is only logical that the most performing model, that outperforms BM25 by a large margin, requires less contribution from this later. Nevertheless, if we take the example of the GOV2 descriptions, despite the similar starting performance at $\alpha = 0.0$ of vanilla and Sim-Pair BERT models, the gap between their performance starts getting wider at only $\alpha = 0.1$ to reach its peak at $\alpha = 0.2$.

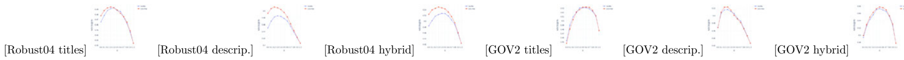
Combining the original document score obtained in the first-stage retriever with passage-level evidence from BERT-based reranking models to determine the final relevance score of a document yields substantial gains in performance. Relevance scores based on traditional IR axioms complete the relevance signals captured by contextual pretrained LMs such as BERT. Moreover, using our simple marking strategy to highlight the exact matching signals in the query-document pairs enhance BERT's own ability to estimate relevance and thus, requires less contribution from BM25 to achieve the best performance.

5.3 Multi-phase fine-tuning

In previous experiments, we leveraged out-of-domain relevance assessments to fine-tune our BERT models. This fine-tuning aims at providing the model with general notions of relevance matching. However, transferring these relevance patterns to the target corpus may, in some cases, be ineffective. To overcome this domain-transfer limitation, we use additional fine-tuning on labeled data drawn from the same distribution as the final task, in other words, in-domain labeled data fine-tuning. This approach is known as "stage-wise" or "multi-phase" fine-tuning (Lin et al., 2020).

Once the models are fine-tuned on the MS MARCO passage dataset following the training setting described in Sect. 4.3, we further fine-tune them on the target task using 5-fold cross validation for both Robust04 and GOV2 collections. We use the folds from (Yang et al., 2019) for Robust04 and the 5-folds configuration adopted by Li et al. (2020).

Following prior work by Dai and Callan (2019), we consider a maximum of 30 passages per document as a trade-off between latency and effectiveness. During training, passages issued from the top 1, 000 documents retrieved by BM25 for queries in the training folds are sub-sampled to avoid catastrophic forgetting. Aside from the first passage, passages in a document are randomly preserved with a probability of 0.1. Passages from a relevant document according to the ground-truth (TREC relevance judgements) are taken as positive



[Robust04 titles] [Robust04 descrip.] [Robust04 hybrid] [GOV2 titles] [GOV2 descrip.] [GOV2 hybrid]

Fig. 2 The end-to-end ranking accuracy of the vanilla $_{\mathrm{BERT}}$ and Sim-Pair $_{\mathrm{BERT}}$ models with BM25 scores interpolation on Robust04 and GOV2 collections. $\alpha = 0.0$ indicates the reranking model effectiveness only without BM25 scores, and $\alpha = 1.0$ means that only BM25 scores are used

examples and passages issued from the other remaining documents as negative examples. We use a *pointwise cross entropy loss* and fine-tune the models for 1 single epoch with a batch size of 32 training instances comprising a query and a passage. We use the Adam optimizer with a learning rate of $1e^{-5}$ with warm up over the first 10% of the total training steps.

For queries in the left-out test fold, we set the rerank threshold to 100 as a trade-off between latency and effectiveness. We report the average performance across all test folds measured in terms of P@20 and nDCG@20 using pytrec_eval[9]. In this setting, our vanilla baseline corresponds to the pointwise trained BERT-MaxP model (Dai and Callan, 2019) initialized with monoBERT fine-tuned on MS MARCO instead of Google's BERT pre-trained checkpoint without any prior fine-tuning on the text ranking task.

Table 11 reports the reranking effectiveness obtained using the multi-phase fine-tuning setting compared to the single-phase MS MARCO fine-tuning (zero-shot transfer setting) for both Robust04 and GOV2 collections. We report results obtained for reranking the top 100 documents retrieved by BM25 in both settings. Thanks to the additional in-domain fine-tuning on the target collection, the performance on both collections improves regardless of the topic field used. We notice in this setting that Sim-Pair $_{\text{BERT}}$ is able to achieve significant gains over the vanilla baseline on the GOV2 collection, confirming our hypothesis that the zero-shot domain transfer from MS MARCO was not sufficient for this collection.

Using the multi-phase fine-tuning setting BERT-based models are able to achieve better performance on descriptions compared to titles on Robust04 by +7.5% and +8.3% for the vanilla and Sim-Pair models respectively, despite the lower retrieval effectiveness of BM25 on descriptions compared titles (−4.3%). On the other hand, the difference in BM25 retrieval effectiveness between descriptions compared to titles is more important on GOV2, about −11%. The BERT-based rerankers reduce this gap to −5.5% and −5.9% for the vanilla and Sim-Pair models respectively but not enough to reverse the tendency. The end-to-end effectiveness on this collection is thus higher on titles than descriptions as observed in previous state-of-the-art models such as BERT-MaxP (Dai and Callan, 2019) or Parade (Li et al., 2020)(see results in Sect. 5.5). Still, the hybrid pipeline outperforms both title and description runs on both collections. The reranking accuracy achieved by the hybrid runs are the highest reported results using a BERT-based model on both collections, at the time this article was written.

5.3.1 Phase-wise marking

Previous results of the Sim-Pair $_{\text{BERT}}$ model presented in Table 11 in the multi-phase setting are obtained using the exact match marking through out the two fine-tuning phases. While the first phase fine-tuning focuses on learning general notions of relevance from a large passage collection, the goal of adding in-domain fine-tuning is to learn directly from labeled data with the same distribution as the target task. It is important to determine on which of the two phases, the marking strategy is more beneficial and at which phase it can be omitted. To this aim, we conduct an ablation study on the Sim-Pair $_{\text{BERT}}$ model. Table 12 shows the results of the marking-strategy ablation on Robust04 and GOV2 collections using the different topic fields. With these results, we can now discuss our research question *RQ3 At which phase the exact match marking is the most beneficial in a multi-phase fine-tuning configuration?*

[9] https://pypi.org/project/pytrec-eval/

Table 11 Reranking effectiveness in the multi-phase vs. zero-shot transfer setting for the Sim-Pair and vanilla models on Robust04 and GOV2 collections

Robust04	Title run				Description run				Hybrid run			
Model	nDCG@20		P@20		nDCG@20		P@20		nDCG@20		P@20	
BM25	0.4240	–	0.3631	–	0.4058	–	0.3345	–	0.4240	–	0.3631	–
BM25+RM3	0.4407	–	0.3821	–	0.4255	–	0.3661	–	0.4407	–	0.3821	–
Zero-shot transfer												
Vanilla $_{\text{BERT}}$	0.4764	–	0.4096	–	0.4611	–	0.3867	–	0.4989	–	0.4245	–
Sim-Pair $_{\text{BERT}}$	0.4763	−0.0	0.4129	+0.8	0.4923‡	+6.8	0.4084‡	+5.6	0.5273‡	+5.7	0.4434‡	+4.5
Multi-phase												
Vanilla $_{\text{BERT}}$	0.4995	–	0.4275	–	0.5368	–	0.4492	–	0.5546	–	0.4715	–
Sim-Pair $_{\text{BERT}}$	**0.5058**	+1.3%	**0.4371**	+2.2%	**0.5479** †	+2.1%	**0.4574** †	+1.8%	**0.5701** ‡	+2.8%	**0.4815** ‡	+2.1%

GOV2	Title run				Description run				Hybrid run			
Model	nDCG@20		P@20		nDCG@20		P@20		nDCG@20		P@20	
BM25	0.4774	–	0.5362	–	0.4264	–	0.4705	–	0.4774	–	0.5362	–
BM25+RM3	0.4851	–	0.5634	–	0.4212	–	0.4966	–	0.4851	–	0.5634	–
Zero-shot transfer												
Vanilla $_{\text{BERT}}$	0.5098	–	0.5916	–	0.4928	–	0.556	–	0.5510	–	0.6312	–
Sim-Pair $_{\text{BERT}}$	0.5181	+1.6	0.599	+1.3	0.4904	−0.5	0.5597	+0.7	0.5531	+0.4	0.6346	+0.5
Multi-phase												
Vanilla $_{\text{BERT}}$	0.5476	–	0.6302	–	0.5175	–	0.5772	–	0.5909	–	0.6604	–
Sim-Pair $_{\text{BERT}}$	**0.5743** ‡	+4.9%	**0.6540** ‡	+3.8%	**0.5406** ‡	+4.5%	**0.6084** ‡	+5.4%	**0.5998**	+1.5%	**0.6758**	+2.3%

Best performances are highlighted in bold

Significant improvements over the vanilla baseline with $p < 0.05$ and $p < 0.01$ after Bonferroni correction are indicated with † and ‡ respectively for the same setting

Change rate over the vanilla baseline in the same setting are reported for each metric (%)

Table 12 Reranking effectiveness with exact matching ablation at different phases of the multi-phase fine-tuning configuration of Sim-Pair $_{\text{BERT}}$ on Robust04 and GOV2 collections

Robust04	Marking		Title run				Description run				Hybrid run			
Run	MS	ID	nDCG@20		P@20		nDCG@20		P@20		nDCG@20		P@20	
Vanilla $_{\text{BERT}}$	–	–	0.4995	–	0.4275	–	0.5368	–	0.4492	–	0.5546	–	0.4715	–
Sim-Pair $_{\text{BERT}}$	✓	✓	**0.5058**	+1.3%	**0.4371**	+2.2%	0.5479†	+2.1%	0.4574†	+1.8%	**0.5701**‡	+2.8%	**0.4815**‡	+2.1%
A	✓	–	0.4978	−0.3%	0.4281	+0.1%	**0.5521**‡	+2.9%	**0.4592**†	+2.2%	0.5678‡	+2.4%	0.4811†	+2.0%
B	–	✓	0.4896*	−2.0%	0.4239	−0.8%	0.5344	−0.4%	0.4504	+0.3%	0.5500	−0.8%	0.4665	−1.0%

GOV2	Marking		Title run				Description run				Hybrid run			
Run	MS	ID	nDCG@20		P@20		nDCG@20		P@20		nDCG@20		P@20	
Vanilla $_{\text{BERT}}$	–	–	0.5476	–	0.6302	–	0.5175	–	0.5772	–	0.5909	–	0.6604	–
Sim-Pair $_{\text{BERT}}$	✓	✓	**0.5743**‡	+4.9%	**0.6540**‡	+3.8%	0.5406‡	+4.5%	0.6084‡	+5.4%	0.5998	+1.5%	**0.6758**	+2.3%
A	✓	–	0.5665†	+3.5%	0.6430	+2.0%	**0.5509**‡	+6.5%	**0.6161**‡	+6.7%	**0.6027**	+2.0%	0.6728	+1.9%
B	–	✓	0.5503	+0.5%	0.6312	+0.2%	0.5218	+0.8%	0.5785	+0.2%	0.5761	−2.5%	0.6517	−1.3%

MS refers to the MS MARCO fine-tuning phase and ID to the in-domain fine-tuning

Best performances are highlighted in bold

Significant improvements over the vanilla baseline with $p < 0.05$ and $p < 0.01$ after Bonferroni correction are indicated with † and ‡ respectively for the same setting

Significant inferiority with $p < 0.05$ is marked with $*$

Change rate over the vanilla baseline are reported for each metric (%)

MS marking (labelled run A in Table 12), uses exact match marking in the MS MARCO (MS) fine-tuning phase only then use the original data without further marking for the in-domain (ID) fine-tuning phase. We can see, in Table 12, that using the marking strategy in the general fine-tuning phase is sufficient to outperform the vanilla baseline or at least perform similarly for Robust04 titles. In other words, initializing BERT with the weights learnt from marked inputs is better than those learnt from non-marked inputs. Ablating marking in the in-domain fine-tuning phase can even surpass the performance of the Sim-Pair $_{\mathrm{BERT}}$ that uses marking across the two fine-tuning phases as observed for descriptions on both collections and the hybrid run on GOV2.

ID marking (labelled run B in Table 12), uses the marking strategy to augment the inputs during fine-tuning on the in-domain data while the BERT model was initialized with the weights learnt from non-marked MS MARCO inputs. The results of this first-phase marking ablation either has no substantial impact on the model's performance or leads to a degradation in performance. This behavior is predictable, since there is not enough in-domain data for BERT to learn useful representations of the marker tokens and their contribution to the relevance prediction.

Using a marking strategy during first general-purpose fine-tuning phase (MS marking) is already enough to outperform the vanilla baseline without requiring additional marking during the in-domain fine-tuning phase. At the end, the fine-tuned model using the Sim-Pair marking strategy on MS MARCO is able to use the relevance matching patterns learned using out-of-domain data, with explicit marking, for later phases even without the guidance of the explicit markers. Nevertheless, additional marking in the in-domain fine-tuning phase used in the classical Sim-Pair $_{\mathrm{BERT}}$ approach is beneficial for title queries where it brings and additional gain of $+1.6\%$ and $+1.4\%$ over the MS marking only (run A) on Robust04 and GOV2, respectively.

5.4 Impact of exact match marking on ELECTRA variant

While BERT is the most famous and largely adopted pretrained language model, additional variants such as RoBERTa (Liu et al., 2019) or ELECTRA (Clark et al., 2020) were proposed in order to improve the model from different aspects. Recent state-of-the-art results reported on Robust04 and GOV2 collections were achieved using the ELECTRA model that appears to outperform BERT. ELECTRA (Clark et al., 2020) replaces the Masked Language Modeling (MLM) with a novel more sample-efficient pretraining task called replaced token detection. In this task, the model learns to distinguish real input tokens from plausible but synthetically generated replacements by a small "generator" model. This approach uses two components: the generator, a small two-layer BERT model that predicts masked tokens and the ELECTRA discriminator model that both require training. However, the new objective allows the model to learn from all input positions rather than only 15% of the positions in the MLM task.

In order to be confident in our approach, we investigate if exact match marking is beneficial for a BERT variant pretrained on a more robust task and study *RQ4 Is exact match marking beneficial in alternative transformer-based models such as ELECTRA?*

For our experiments, we use the base version of the ELECTRA model as the core of our model architecture illustrated in Fig. 1 as a replacement of the BERT model. We use the same single-layer neural network that estimates a score $R(d, q)$ quantifying how relevant the candidate document d is to the query q. We also use the same fine-tuning hyper parameters used with BERT.

5.4.1 In-domain effectiveness

Using the same setting used for the BERT-based models, we report the results obtained on TREC DL2019 and 2020 test collections in Table 13. For clarity, we only show results with the Sim-Pair marking strategy, full results with all the strategies can be found in Appendix 3.

Interestingly using the ELECTRA core in place of BERT in the vanilla baseline does not lead to increased performance and we even observe a slight drop in performance in TREC DL 2020. Adding exact match marking, using both cores, leads to similar gains over the vanilla baselines. While the gain in average precision is more pronounced with ELECTRA on both DL 2019 and 2020, the effectiveness in terms of nDCG@10 is more interesting with the BERT core on the DL 2020 test collection.

5.4.2 Zero-shot transfer setting

We use the fine-tuned models on exclusively out-of-domain data, i.e MS MARCO passage dataset, and apply inference on the window-passages obtained by splitting each document using the same passage length of 150 words and a 75 words stride used in the BERT experiments. Table 14 shows the results obtained at cutoff 1, 000 on both Robust04 and GOV2 collections. We recall the results of the Vanilla and Sim-Pair models with the BERT core for comparison.

Exact Match Marking on ELECTRA Results indicate clearly that adding exact match marking is still beneficial for the ELECTRA variant. As for the BERT version, Sim-Pair $_{\text{ELECTRA}}$ is more effective on Robust04 with an average improvement rate of +5% compared to only half, +2.5%, on GOV2. However, exact match marking has more notable impact on titles rather than descriptions, when the vanilla $_{\text{ELECTRA}}$ baseline prefers clearly description queries.

ELECTRA versus BERT core The Sim-Pair$_{\text{ELECTRA}}$ variant achieves better performance than its BERT counterpart regardless of the topic field on the GOV2 collection. In contrast, using the BERT core is more effective on Robust04 on both titles, descriptions and the hybrid pipeline. The same tendency can be observed for the vanilla baseline with smaller margins.

Table 13 Reranking effectiveness on the TREC DL 2019 and DL 2020 Document ranking tasks for Sim-Pair and vanilla models with both BERT and ELECTRA cores

TREC DL Doc	DL 19				DL 20			
Model	nDCG@10		MAP		nDCG@10			
BM25	0.5176	–	0.2434	–	0.5286	–	0.3793	–
BM25+RM3	0.5169	–	0.2772	–	0.5248	–	0.4006	–
Vanilla $_{\text{BERT}}$	0.6726	–	0.3006	–	0.6340	–	0.4523	–
Sim-Pair $_{\text{BERT}}$	0.6798	+1.1%	0.3057	+1.7%	**0.6495**	+2.4%	0.4505	−0.4%
Vanilla $_{\text{ELECTRA}}$	0.6738	–	0.2976	–	0.6236	–	0.4297	–
Sim-Pair $_{\text{ELECTRA}}$	**0.6816**	+1.2%	**0.3062**	+2.9%	0.6331	+1.5%	**0.4543** †	+5.7%

Best performances are highlighted in bold

Significant improvements over the vanilla baseline with $p < 0.05$ are indicated with †, for the same core

Change rate over the vanilla baseline for the same core type are reported for each metric (%)

Table 14 Reranking effectiveness in the zero-shot transfer setting for the Sim-Pair and vanilla models on Robust04 and GOV2 collections using both BERT and ELECTRA cores

Robust04	Title run				Description run				Hybrid run			
Model	nDCG@20		P@20		nDCG@20		P@20		nDCG@20		P@20	
BM25	0.4240	–	0.3631	–	0.4058	–	0.3345	–	0.4240	–	0.3631	–
BM25+RM3	0.4407	–	0.3821	–	0.4255	–	0.3661	–	0.4407	–	0.3821	–
Vanilla $_{\text{BERT}}$	0.4652	–	0.4046	–	0.4510	–	0.3851	–	0.4845	–	0.4147	–
Sim–Pair $_{\text{BERT}}$	**0.4773**	+2.6%	**0.4155**	+2.7%	**0.4931** ‡	+9.3%	**0.4169** ‡	+8.3%	**0.5239** ‡	+8.1%	**0.4446** ‡	+7.2%
Vanilla $_{\text{ELECTRA}}$	0.4416	–	0.3833	–	0.4482	–	0.3831	–	0.4782	–	0.4141	–
Sim-Pair $_{\text{ELECTRA}}$	0.4717‡	+6.8%	0.4124	+7.6%	0.4597	+2.6%	0.3886	+1.4%	0.5043‡	+5.5%	0.4263	+2.9%
GOV2	Title run				Description run				Hybrid run			
Model	nDCG@20		P@20		nDCG@20		P@20		nDCG@20		P@20	
BM25	0.4774	–	0.5362	–	0.4264	–	0.4705	–	0.4774	–	0.5362	–
BM25+RM3	0.4851	–	0.5634	–	0.4212	–	0.4966	–	0.4851	–	0.5634	–
Vanilla $_{\text{BERT}}$	0.4533	–	0.5272	–	0.4696	–	0.5248	–	0.4937	–	0.5611	–
Sim-Pair $_{\text{BERT}}$	0.4468	−1.4%	0.5134	−2.6%	0.4687	−0.2%	0.5326	+1.5%	0.4991	+1.1%	0.5695	+1.5%
Vanilla $_{\text{ELECTRA}}$	0.4668	–	0.5332	–	0.4986	–	0.5601	–	0.5147	–	0.5765	–
Sim-Pair $_{\text{ELECTRA}}$	**0.4881** ‡	+4.6%	**0.5577** ‡	+4.6%	**0.5030**	+0.9%	**0.5634**	+0.6%	**0.5249**	+2.0%	**0.5923**	+2.7%

Best performances are highlighted in bold

Significant improvements over the vanilla baseline with $p < 0.05$ and $p < 0.01$ are indicated with † and ‡ respectively for the same core

Change rate over the vanilla baseline for the same core type are reported for each metric (%)

5.4.3 Multi-phase fine-tuning

Table 15 shows the results obtained using the multi-phase fine-tuning on both MS MARCO passage dataset and in-domain labeled data, described in Sect. 5.3 for BERT. The ELECTRA-based models outperform the BERT-based models on both collections regardless of the topic field used indicating that ELECTRA is a more effective core PLM than BERT in a multi-phase fine-tuning setting. However, adding exact match marking has no significant impact in this setting. Sim-Pair $_{\text{ELECTRA}}$ performs slightly better than the vanilla $_{\text{ELECTRA}}$ baseline on the Robust04 collection across title, description and hybrid runs. On the other hand, exact match marking leads to better ranking accuracy on GOV2 titles, but provokes a slight degradation in performance when the description field is used for reranking (description and hybrid runs).

Exact match marking is indeed beneficial for the ELECTRA model, especially in a zero-shot transfer setting where no labeled data is available in the target domain. Sim-Pair $_{\text{ELECTRA}}$ is able to achieve significant gains on titles, where Sim-Pair $_{\text{BERT}}$ is less effective. However, for description and hybrid runs that use descriptions for reranking, exact match marking appears to have more substantial impact when using a BERT core. On TREC DL 2019 and 2020 benchmarks, both vanilla and Sim-Pair models perform similarly with both BERT and ELECTRA cores. The only advantage of the ELECTRA core is increased average precision with Sim-Pair. Finally, we can say that, in most cases, the ELECTRA-based versions of our models are more effective compared to their BERT counterparts.

5.5 Comparison with state-of-the-art baselines

In this section we try to situate our approach with regard to what has already been proposed for document ranking. In a first part, we try to conduct comparative evaluations with models presenting a similar experimental setup for a fair comparison. Then in a second part, we compare our best runs to a wide variety of SOTA approaches with different configurations.

5.5.1 Comparison in the same experimental design.

In order to fairly compare a novel approach with previously proposed ones, it is important to conduct the evaluation in the same experimental conditions. Here, we try to reproduce as much of the original settings used to produce the results of the Birch and BERT-maxP baselines, respectively.

Birch (MS) This baseline is fine-tuned exclusively on MS MARCO passages, therefore we use our Sim-Pair $_{\text{BERT}}$ + BM25 model equally fine-tuned on MS MARCO passages and augmented with BM25 scores interpolation following the same Equation 6 used in Birch (Akkalyoncu Yilmaz et al., 2019). All Robust04 and GOV2 topics and relevance judgements are used as a held-out test set.

Table 16 shows the results of our Sim-Pair $_{\text{BERT}}$ + BM25 model compared to the Birch (MS) baseline. The results clearly indicate that our model outperforms Birch (MS). Since our model already outperforms the baseline with a BERT $_{\text{Base}}$ version, it unnecessary to conduct the same experiment with a BERT $_{\text{Large}}$ whose computational cost is, unfortunately, beyond our hardware limitations.

BERT-MaxP (MS) The configuration of this baseline is the same we used in the multi-phase fine-tuning setting. We compare the results of Sim-Pair $_{\text{BERT}}$ fine-tuned first on MS MARCO and then further fine-tuned on the target task obtained with a 5-fold cross

Table 15 Reranking effectiveness in the multi-phase fine-tuning setting for the Sim-Pair and vanilla models on Robust04 and GOV2 collections using both BERT and ELECTRA cores

Robust04	Title run				Description run				Hybrid run			
Model	nDCG@20		P@20		nDCG@20		P@20		nDCG@20		P@20	
BM25	0.4240	–	0.3631	–	0.4058	–	0.3345	–	0.4240	–	0.3631	–
BM25+RM3	0.4407	–	0.3821	–	0.4255	–	0.3661	–	0.4407	–	0.3821	–
Vanilla $_{\text{BERT}}$	0.4995	–	0.4275	–	0.5368	–	0.4492	–	0.5546	–	0.4715	–
Sim-Pair $_{\text{BERT}}$	0.5058	+1.3%	0.4371	+2.2%	0.5479†	+2.1%	0.4574†	+1.8%	0.5701‡	+2.8%	0.4815‡	+2.1%
Vanilla $_{\text{ELECTRA}}$	0.5375	–	0.4560	–	0.5676	–	0.4663	–	0.5901	–	0.4902	–
Sim-Pair $_{\text{ELECTRA}}$	**0.5380**	+0.1%	**0.4564**	+0.1%	**0.5686**	+0.2%	**0.4705**	+0.9%	**0.5927**	+0.4%	**0.4942**	+0.8%
GOV2	Title run				Description run				Hybrid run			
Model	nDCG@20		P@20		nDCG@20		P@20		nDCG@20		P@20	
BM25	0.4774	–	0.5362	–	0.4264	–	0.4705	–	0.4774	–	0.5362	–
BM25+RM3	0.4851	–	0.5634	–	0.4212	–	0.4966	–	0.4851	–	0.5634	–
Vanilla$_{\text{BERT}}$	0.5476	–	0.6302	–	0.5175	–	0.5772	–	0.5909	–	0.6604	–
Sim-Pair $_{\text{BERT}}$	0.5743‡	+4.9%	0.6540‡	+3.8%	0.5406‡	+4.5%	0.6084‡	+5.4%	0.5998	+1.5%	0.6758	+2.3%
Vanilla $_{\text{ELECTRA}}$	0.5784	–	0.6621	–	**0.5629**	–	**0.6279**	–	**0.6149**	–	0.6862	–
Sim-Pair $_{\text{ELECTRA}}$	**0.5868**	+1.5%	**0.6661**	+0.6%	0.5552	−1.4%	0.6225	−0.9%	0.6133	−0.3%	**0.6926**	+0.9%

Best performances are highlighted in bold

Significant improvements over the vanilla baseline with $p < 0.05$ and $p < 0.01$ are indicated with † and ‡ respectively for the same core

Change rate over the vanilla baseline for the same core type are reported for each metric (%)

Table 16 Reranking effectiveness of the Sim-Pair $_{\text{BERT}}$ with interpolating BM25 scores vs. Birch (MS) baseline on both Robust04 and GOV2 collections

Robust04	Title run				Description run			
Model	nDCG@20		P@20		nDCG@20		P@20	
BM25	0.4240	–	0.3631	–	0.4058	–	0.3345	–
Birch (MS)	0.4227	–	0.3616	–	0.4053	–	0.3341	–
Sim-Pair $_{\text{BERT}}$ + BM25	**0.4947**	+17.%	**0.4265**	+18.%	**0.5098**	+26.%	**0.4279**	+28.%
GOV2	**Title run**				**Description run**			
Model	nDCG@20		P@20		nDCG@20		P@20	
BM25	0.4774	–	0.5362	–	0.4264	–	0.4705	–
Birch (MS)	0.4722	–	0.5352	–	0.4260	–	0.4701	–
Sim-Pair $_{\text{BERT}}$ + BM25	**0.5327**	+13.%	**0.6000**	+12.%	**0.5235**	+23..%	**0.5893**	+25.%

Best performances are highlighted in bold

validation with BERT-MaxP (MS) in Table 17. We report the results when using the exact match marking during fine-tuning on MS MARCO passages only [MS], and the results with the full marking on both MS MARCO and in-domain data, i.e., Sim-Pair $_{\text{BERT}}$. Our approach outperforms clearly the BERT-maxP baseline on titles, and performs slightly better on descriptions. It is important to notice that the BERT-MaxP results reported by Li et al. (2020) are better than our vanilla $_{\text{BERT}}$ baseline in the multi-phase fine-tuning setting,

Table 17 Reranking effectiveness of the Sim-Pair $_{\text{BERT}}$ with multi-phase fine-tuning vs. BERT-MaxP (MS) baseline on both Robust04 and GOV2 collections

Robust04	Title run				Description run			
Model	nDCG@20		P@20		nDCG@20		P@20	
BM25	0.4240	–	0.3631	–	0.4058	–	0.3345	–
BERT–MaxP (MS)	0.4931	–	0.4277	–	0.5453	–	0.4522	–
Sim-Pair $_{\text{BERT}}$	**0.5058**	+2.6%	**0.4371**	+2.2%	0.5479	+0.5%	0.4574	+1.1%
Sim-Pair $_{\text{BERT}}$ [MS]	0.4978	+1.0%	0.4281	+0.1%	**0.5521**	+1.2%	**0.4592**	+1.5%
GOV2	**Title run**				**Description run**			
Model	nDCG@20		P@20		nDCG@20		P@20	
BM25	0.4774	–	0.5362	–	0.4264	–	0.4705	–
BERT–MaxP (MS)	0.5600	–	0.6352	–	0.5506	–	0.6087	–
Sim-Pair $_{\text{BERT}}$	**0.5743**	+2.6%	**0.6540**	+3.0%	0.5406	−1.8%	0.6084	−0.0%
Sim-Pair $_{\text{BERT}}$ [MS]	0.5665	+1.2%	0.6430	+1.2%	**0.5509**	+0.1%	**0.6161**	+1.2%

[MS] indicates that the run uses MS marking: exact match marking is only used during fine-tuning on MS MARCO and ablated in the in-domain fine-tuning phase

Best performances are highlighted in bold especially on GOV2. This slight difference can be exaplined by the the traditional use of the pointwise loss function (monoBERT (Nogueira and Cho, 2019)) while they use a pairwise loss function.

5.5.2 Comparison with different experimental designs

Each approach has the optimal experimental conditions that lead to the best ranking accuracy possible, and these optimal conditions are hardly the same for the different models we want to compare. Independently of the experimental framework employed to obtain the results, or the nature of the approach, Table 18 compares our best runs with both BERT and ELECTRA cores obtained in the multi-phase fine-tuning setting, with the best baseline runs. While Table 19 compares our best in-domain runs to both TREC best runs from the TREC DL 2019 and 2020 tracks and the SOTA baselines.

Robust04 and GOV2 collections nsurprisingly, the reranking models achieve the best results and largely outperform all other baselines. For a fair comparison with the sparse and dense retrieval methods (runs [03-07]) which do not use target-domain fine-tuning, we add our runs in the zero-shot setting on descriptions (runs [08-09]). Nevertheless, our rerankers still outperform the retrievers.

Results obtained using the best Sim-Pair $_{\mathrm{BERT}}$, run [17] in Table 18, outperform all the BERT-based models that represent the state of the art and achieves better performance than T5 for both base and large versions on robust04. The Sim-Pair $_{\mathrm{ELECTRA}}$ variant (run [18]) achieves comparable performance with the T5-3B model while using only 3.6% of its parameters and outperforms the Parade $_{\mathrm{ELECTRA}}$ model on both Robust04 and GOV2 collections by a varying margin from +3% to more than +4%. The T5 baseline is by far the strongest baseline, it is important to note that it uses a zero-shot transfer setting without the need for in-domain fine-tuning as opposed to BERT-MaxP, Parade and our best runs [17-18], however, its large size make it unpractical compared to a BERT$_{\mathrm{Base}}$ or ELECTRA $_{\mathrm{Base}}$.

TREC DL Document Ranking task imilarly to the Robust04 and GOV2 results, the best TREC runs which are cross-encoding rerankers outperform all other baselines. For TREC DL 2019, we include the best `idst_bert_r1` run (Yan et al., 2019) which uses StructBERT (Wang et al., 2020), a BERT model which better models sentence relationships thanks to an improved Next Sentence Prediction task, and `ucas_runid1` Chen et al. (2019) which uses BERT-MaxP Dai and Callan (2019). We also include Parade results Li et al. (2020). Our runs outperform Parade and `ucas_runid1` but cannot outperform `idst_bert_r1` –StructBERT core– in terms of nDCG@10. In TREC DL 2020, the best run `d_d2q_duo` Pradeep et al. (2020) is a large multi-stage ranking model including a BM25 retriever, DocT5Query document expansion and two cascading T5-3B rerankers, making hard to outperform. The `ICIP_run1` Chen et al. (2020), uses a BERT-Large model at its core with a refined fine-tuning process including passage filtering and better negative sampling which explains its higher performance. Nevertheless, our runs are still competitive and outperform Parade which has the same model size as our models. Interestingly, the performance on TREC DL 2020 are lower in terms on nDCG@10 compared to TREC DL 2019 for the same model as observed for both our runs and the Parade run.

Table 18 Reranking effectiveness on Robust04 and GOV2 of our best runs versus the best baseline runs

Runs	Robust04					GOV2				
	nDCG@20		P@20		Field	nDCG@20		P@20		Field
Lexical Retrieval										
[01] BM25	0.4240	+40.%	0.3631	+36.%	Title	0.4774	+28.%	0.5362	+29.%	Title
[02] BM25+RM3	0.4407	+34.%	0.3821	+29.%	Title	0.4851	+26.%	0.5634	+23.%	Title
Sparse Retrieval										
[03] BOW+DeepCT-Query	0.4450	+33.%	–	–	Desc	0.4300	+43.%	–	–	Desc
[04] BM25+DocT5Query	0.4076	+45.%	0.3361	+47.%	Title	–	–	–	–	–
Dense Retrieval										
[05] DPR	0.1723	+244%	0.1333	+271%	Desc	0.1618	+279%	0.1644	+321%	Desc
[06] ANCE	0.3517	+69.%	0.2767	+79.%	Desc	0.3604	+70.%	0.3738	+85.%	Desc
[07] ColBERT	0.3619	+64.%	0.2924	+69.%	Desc	–	–	–	–	–
Zero-shot Setting										
[08] Sim-Pair $_{\text{BERT}}$	0.4931	+20.%	0.4169	+19.%	Desc	0.4687	+31.%	0.5326	+30.%	Desc
[09] Sim-Pair $_{\text{ELECTRA}}$	0.4597	+29.%	0.3886	+27.%	Desc	0.5030	+22.%	0.5634	+23.%	Desc
Reranking										
[10] Birch (MS-MB)	0.5137	+15.%	0.4404	+12.%	Title	0.5608	+9.4%	0.6409	+8.1%	Title
[11] BERT-MaxP (MS)	0.5453	+8.7%	0.4522	+9.3%	Desc	0.5600	+9.5%	0.6356	+9.0%	Title
[12] Parade	0.5605	+5.7%	0.4661	+6.0%	Desc	0.5750	+6.7%	0.6530	+6.1%	Title
[13] Parade $_{\text{ELECTRA}}$	0.5713	+3.7%	0.4717	+4.8%	Desc	0.5851	+4.8%	0.6678	+3.7%	Title
[14] T5-base	0.5298	+12.%	–	–	Hybrid	–	–	–	–	–
[15] T5-large	0.5345	+11.%	–	–	Hybrid	–	–	–	–	–
[16] T5-3B	**0.6091**	−2.7%	–	–	Hybrid	–	–	–	–	–
Best runs										
[17] Sim-Pair $_{\text{BERT}}$	0.5701	–	0.4815	–	Hybrid	0.5998	–	0.6758	–	Hybrid

Table 18 continued

Runs	Robust04					GOV2				
	nDCG@20		P@20		Field	nDCG@20		P@20		Field
[18] Sim-Pair $_{\text{ELECTRA}}$	**0.5927**	–	0.4942	–	Hybrid	**0.6133**	–	0.6926	–	Hybrid

The change rate (%) of our best run, Sim-Pair $_{\text{ELECTRA}}$, over each baseline is indicated for both metrics if available. We use the multi-phase fine-tuning for our runs, the same multi-phase fine-tuning is adapted in Parade and BERT-maxP baselines. For a fair comparison with sparse and dense retrieval models we add Sim-Pair runs in the zero-shot setting on descriptions

Best performances are highlighted in bold

Table 19 Reranking effectiveness on TREC DL 2019 and 2020 Document ranking tasks of our Sim-Pair models with both BERT and ELECTRA cores versus the best TREC runs and baselines

Runs	DL 2019				DL 2020			
	nDCG@10		MAP@100		nDCG@10		MAP@100	
Lexical Retrieval								
[01] BM25	0.5176	+31.%	0.2434	+26.%	0.5286	+23.%	0.3793	+19.%
[02] BM25+RM3	0.5170	+32.%	0.2774	+10.%	0.5225	+24.%	0.4014	+12.%
Sparse Retrieval								
[03] BM25+HDCT	0.4523	+50.%	0.2067	+13.%	0.4506	+44.%	0.3022	+49.%
[04] BM25+DocT5Query	0.5968	+14.%	0.2700	+13.%	0.5885	+10.%	0.4230	+6.5%
Dense Retrieval								
[05] DPR*	0.5570	+22.%	–	–	–	–	–	–
[06] ANCE*	0.6150	+10.%	–	–	–	–	–	–
[07] ColBERT	0.5756	+18.%	0.1914	+60.%	0.5481	+18.%	0.2963	+52.%
Reranking-TREC								
[08] ucas_runid1	0.6437	+5.7%	0.2642	+16.%	–	–	–	–
[09] idst_bert_r1	**0.7189**	−5.4%	0.2915	+5.0%	–	–	–	–
[10] Parade	0.6500	+4.6%	0.2740	+12.%	0.6010	+8.1%	0.4030	+12.%
[11] d_d2q_duo	–	–	–	–	**0.6934**	−6.3%	**0.5422**	−17.%
[12] ICIP_run1	–	–	–	–	0.6623	−1.9%	0.4333	+4.0%
Our Runs								
[13] Sim-Pair $_{\text{BERT}}$	0.6798	–	0.3057	–	0.6495	–	0.4505	–
[14] Sim-Pair $_{\text{ELECTRA}}$	0.6801	–	**0.3061**	–	0.6331	–	0.4543	–

The change rate (%) of our best run, over each baseline is indicated for both metrics if available

Best performances are highlighted in bold

DPR* and ANCE* results were copied from the ANCE paper (Xiong et al., 2021)

6 Discussion and future work

Our research is related to effectively harnessing the exact matching signals from the query-document pairs to enhance document ranking with pretrained language models (PLMs) exemplified by BERT. We have shown through the empirical experiments reported in this paper that PLMs such as BERT can benefit from explicit exact match cues conveyed via marker tokens to be more effective for ad hoc ranking.

BERT as the most famous PLM, was successfully applied to text ranking as well as a wide range of other tasks without requiring any specialized neural architectural components to capture different relevance signals as opposed to pre-BERT neural ranking models. Previous work by Qiao et al. (2019) study the behaviour of BERT for ranking and find that it is able to capture semantic matching signals between paraphrase tokens. However, research from the pre-BERT era have proven that, in addition to semantic matching, exact matching is still an important cue for neural ranking models (Guo et al., 2016; Mitra et al., 2017). Guo at al. Guo et al. (2016) argue that "exact matching of terms in documents with those in queries is still *the most important signal* in ad hoc retrieval due to the indexing and search paradigm in modern search engines". This is why, (Boualili et al., 2020) suggest to emphasize the exact match signals for BERT using a marking technique that does not involve redesigning the model's architecture, which will cost the immense benefits of self-supervised pretraining.

In this paper we extend (Boualili et al., 2020) and study four research questions that aim to investigate the effectiveness of our newly proposed marking strategies for ad hoc document ranking.

First, we investigated the benefits of exact match marking for a BERT-based model in both in-domain and zero-shot transfer settings. The results of the experiment showed that combining a simple soft marker with a pair marking strategy (Sim-Pair) is the most simple yet effective marking strategy. Moreover, experiments on Robust04 and GOV2 showed this exact match marking approach has a higher effectiveness on the description field of the topic compared to the title field. This preference for well-written natural language questions is in line with BERT's preference for descriptions revealed by Dai and Callan (2019). On the other hand, we follow a retrieve-then-rerank architecture where the retriever is a bag-of-words model that prefers short key word queries while the reranker is a BERT-based model that prefers long natural language questions (Dai and Callan, 2019; Nogueira et al., 2020). In order to get the best of the two stages, we propose a hybrid pipeline where titles are used during the retrieving stage and then replaced by descriptions in the reranking stage which leads to substantial gains in performance.

Second, we investigate how to improve effectiveness on the out-of-domain collections using two methods: (1) linear interpolation of BM25 document-level scores with BERT-based passage-level scores, and (2) adding in-domain fine-tuning on the target collection. With the first method, we find exact term matching scores from traditional bag-of-words models like BM25 are still beneficial for BERT-based document reranking for out-of-domain collections. Indeed, combining document-level scores from BM25 with passage-evidence from out BERT-based models with a simple linear interpolation leads to substantial gains in performance. The document-level scores from initial BM25 retrieval based on traditional IR cues (TF, IDF) provide additional relevance signals that complete the passage-level scores from BERT-based models. Furthermore, using exact match marking appears to better take advantage of the combination with BM25 scores to achieve better performance than the vanilla model.

With the second method, when adding in-domain fine-tuning on top of the first general-purpose fine-tuning phase on out-of-domain data, we demonstrated through an ablation study that using exact match marking in the general-purpose fine-tuning phase on large out-of-domain data is enough to achieve substantial leaps in performance especially on descriptions. We publish our fine-tuned checkpoints on MS MARCO so it can be accessible to the community as a more effective alternative to a vanilla checkpoint.

Third, we study the contribution of our exact match marking strategy on a BERT variant, ELECTRA, that has been recently used in state of the art models such as Parade (Li et al., 2020). Experiments showed that exact match marking is indeed beneficial for ELECTRA, especially in the zero-shot transfer setting where no in-domain annotated data is used for training. In addition, the ELECTRA-based models were able to outperform their BERT counterparts in most cases.

Finally, we compared our best runs using both BERT and ELECTRA to a wide range of transformer-based ranking models that represent the state of the art at the time this article was written. On the one hand, the comparative evaluation showed that our exact match marking approach combined with the hybrid pipeline, that uses titles for BM25 retrieval and descriptions for BERT reranking, achieves near state-of-the-art results on Robust04 compared to the strong and larger T5-3B baseline, and outperforms previously proposed models on GOV2. On the other hand, the comparative evaluation on the TREC DL Document rankings tasks of 2019 and 2020, showed that our marking-based approach is a competitive model compared to the best TREC runs. Even if this evaluation is an in-domain setting, the benefits from exact match marking seem to be less prominent than those observed on Robust04 or GOV2. Differently from the title and description queries used with Robust04 and GOV2, the TREC DL queries are questions. Additionally, documents and other aspects of evaluation also differ. Further analysis is needed to determine the factors behind these discrepancy.

At the end, what does this mean for a deployment choice of a vanilla BERT vs. Sim-Pair $_{\mathrm{BERT}}$? We would argue Sim-Pair $_{\mathrm{BERT}}$ induces focus on exact match signals leading to better performance than the vanilla BERT (in 24 comparisons, with 9 being significant), or at least to comparable performance (only in 4 comparisons, with no significant loss). Importantly, our extensive experiments did not show a single case where Sim-Pair $_{\mathrm{BERT}}$ perform significantly worse, thus we would recommend it. On the efficiency side, our approach inherits the efficiency issue of the monoBERT cross-encoder. However, we do not add more complexity to the model making our approach a better substitute for a vanilla BERT with the exact same number of parameters (110M).

Our approach was empirically proven to be effective on standard ad hoc benchmarks, however in terms of explicability, there is still a lot of analysis that need to be done in order to understand how exactly the marking conveys the exact match signals to BERT and how are they integrated in the relevance prediction process. To this day, only so little is understood about the inner workings of BERT and PLMs in general regardless of all the efforts put into studying their behaviors. Previous research attempted to reveal insights about how BERT "works" in the limited context of passage retrieval, but studies lack when it comes to long documents ranking. Aside from the explicability limitation, our approach is rather simple and considers all query terms to be of equal importance when, in reality, they hardly have the same importance in the query especially in long descriptions.

For future work, we plan to develop diagnostic tests in attempt to shed light on the contribution of the exact match marking to the inner workings of BERT. Once the intervention of the markers determined, their representations can be leveraged for relevance classification in addition or instead of the current standard [CLS]. Identifying the subset of queries that are most likely to be improved by adding explicit exact match cues can be used to choose whether to use marking or not. Furthermore, our approach could be further improved by integrating the query term importance. Finally, other methods may be investigated to better integrate exact match signals into BERT.

7 Conclusion

Pretrained language models perform well on an impressively wide range of tasks. They were proven to excel at semantic matching, nevertheless exact matching is essential for relevance matching. In the light of this fact, we proposed to use marker tokens to convey exact match cues from the textual input that yield strong performance while maintaining the same architecture i.e number of parameters. We showed through empirical experiments that using a simple marker combined with a pair marking level is the most simple strategy that yields the best effectiveness. We show that applying this marking strategy in a hybrid retrieve-than-rerank pipeline that uses short key word queries for the first bag-of-words retriever and then adopts long natural language queries for reranking with PLMs like BERT and ELECTRA produce competitive effectiveness compared to state-of-the-art models. We published our fine-tuned checkpoints on marked data on the HuggingFace model hub so it can be easily used by the community via the famous “transformers” library without changes to their setups while benefiting from the improvements brought by exact match marking and build upon them.

Appendix 1. Results using the BERT core with all marking strategies

Appendix 1.1 Zero-shot transfer setting

Table 20 shows the full results obtained using all the proposed strategies on Robust04 and GOV2 collections at cutoff 100 and 1, 000. We report results using the title, description and hybrid runs. The results at cutoff 1, 000 complement the reported results in Tables 7 and 9. For the 100-cutoff results, they complement the results of Table 11 for the zero-shot transfer section. We report the results at cutoff 100 for direct comparison with the multi-phase fine-tuning setting where we only rerank the top-100 documents retrieved by BM25 as a trade-off between effectiveness and efficiency.

Appendix 1.2 Multi-phase fine-tuning setting

Table 21 shows the results obtained using the multi-phase fine-tuning setting described in Sect. 5.3 for all our models using all proposed marking strategies. These results expand those presented in Table 11.

Appendix 2. Results using the ELECTRA core with all marking strategies

Appendix 2.1 Zero-shot transfer setting

Table 22 resumes the results of applying all proposed marking strategies on the ELECTRA core model for Robust04 and GOV2 collections. This table complements the results presented in Table 14. We add the results at the reranking cutoff 100 in order to give an idea about the zero-shot setting results without in-domain fine-tuning directly comparable with

Table 20 Reranking effectiveness in the zero-shot transfer setting of all our models on Robust04 and GOV2 collections

Robust04	Title run				Description run				Hybrid run			
Model	nDCG@20		P@20		nDCG@20		P@20		nDCG@20		P@20	
BM25	0.4240	–	0.3631	–	0.4058	–	0.3345	–	0.4240	–	0.3631	–
BM25+RM3	0.4407	–	0.3821	–	0.4255	–	0.3661	–	0.4407	–	0.3821	–
Top-100												
Vanilla $_{\mathrm{BERT}}$	0.4764	–	0.4096	–	0.4611	–	0.3867	–	0.4989	–	0.4245	–
Sim-Doc $_{\mathrm{BERT}}$	0.4678	−1.8%	0.4042	−1.3%	0.4616	+0.1%	0.3865	−0.1%	0.4912	−1.5%	0.4129	−2.7%
Sim-Pair $_{\mathrm{BERT}}$	0.4763	−0.0%	**0.4129**	+0.8%	**0.4923** ‡	+6.8%	**0.4084** ‡	+5.6%	**0.5273** ‡	+5.7%	**0.4434** ‡	+4.5%
Pre-Doc $_{\mathrm{BERT}}$	**0.4781**	+0.4%	0.4078	−0.4%	0.4867‡	+5.6%	0.4016‡	+3.9%	0.5205‡	+4.3%	0.4294‡	+1.2%
Pre-Pair $_{\mathrm{BERT}}$	0.4700	−1.3%	0.4064	−0.8%	0.4812‡	+4.4%	0.3974‡	+2.8%	0.5132‡	+2.9%	0.4410‡	+3.9%
Top-1000												
Vanilla $_{\mathrm{BERT}}$	0.4652	–	0.4046	–	0.4510	–	0.3851	–	0.4845	–	0.4147	–
Sim-Doc $_{\mathrm{BERT}}$	0.4447*	−4.4%	0.3831*	−5.3%	0.4166*	−7.6%	0.3510*	−8.9%	0.4476*	−7.6%	0.3817*	−7.9%
Sim-Pair $_{\mathrm{BERT}}$	**0.4773**	+2.6%	**0.4155**	+2.7%	**0.4931** ‡	+9.3%	**0.4169** ‡	+8.3%	**0.5239** ‡	+8.1%	**0.4446** ‡	+7.2%
Pre-Doc $_{\mathrm{BERT}}$	**0.4767**	+2.5%	**0.4084**	+0.9%	0.4789‡	+6.2%	0.4026‡	+4.5%	0.5035‡	+3.9%	0.4235	+2.1%
Pre-Pair $_{\mathrm{BERT}}$	0.4654	+0.0%	0.4024	−0.5%	0.4795‡	+6.3%	0.4034‡	+4.8%	0.5086‡	+5.0%	0.4319‡	+4.1%
GOV2	Title run				Description run				Hybrid run			
Model	nDCG@20		P@20		nDCG@20		P@20		nDCG@20		P@20	
BM25	0.4774	–	0.5362	–	0.4264	–	0.4705	–	0.4774	–	0.5362	–
BM25+RM3	0.4851	–	0.5634	–	0.4212	–	0.4966	–	0.4851	–	0.5634	–
Top-100												
Vanilla $_{\mathrm{BERT}}$	0.5098	–	0.5916	–	0.4928	–	0.556	–	0.5510	–	0.6312	–
Sim-Doc $_{\mathrm{BERT}}$	0.5146	+0.9%	0.5936	+0.3%	0.4884	−0.9%	0.5557	−0.1%	0.5497	−0.2%	**0.6359**	+0.7%
Sim-Pair $_{\mathrm{BERT}}$	**0.5181**	+1.6%	**0.5990**	+1.3%	0.4904	−0.5%	0.5597	+0.7%	0.5531	+0.4%	0.6346	+0.5%
Pre-Doc $_{\mathrm{BERT}}$	0.5100	−0.0%	0.5903	−0.2%	**0.4952**	+0.5%	**0.5601**	+0.7%	**0.5568**	+1.1%	0.6322	+0.2%

Table 20 continued

GOV2	Title run				Description run				Hybrid run			
Model	nDCG@20		P@20		nDCG@20		P@20		nDCG@20		P@20	
Pre-Pair $_{\text{BERT}}$	0.5168	+1.4%	0.5936	+0.3%	0.4920	−0.2%	0.5584	+0.4%	0.5559	+0.9%	0.6332	+0.3%
Top-1000												
Vanilla $_{\text{BERT}}$	0.4533	–	0.5272	–	0.4696	–	0.5248	–	0.4937	–	0.5611	–
Sim-Doc $_{\text{BERT}}$	0.4588	+1.2%	0.5349	+1.5%	0.4686	−0.2%	0.5262	+0.3%	0.4943	+0.1%	0.5607	−0.1%
Sim-Pair $_{\text{BERT}}$	0.4468	−1.4%	0.5134	−2.6%	0.4687	−0.2%	0.5326	+1.5%	0.4991	+1.1%	**0.5695**	+1.5%
Pre-Doc $_{\text{BERT}}$	0.4485	−1.1%	0.5121	−2.9%	**0.4768**	+1.5%	**0.5315**	+1.3%	**0.5013**	+1.5%	0.5668	+1.0%
Pre-Pair $_{\text{BERT}}$	0.4515	−0.4%	0.5238	−0.6%	0.4752	+1.2%	0.5285	+0.7%	0.4979	+0.9%	0.5594	−0.3%

Best results, for each cutoff, are highlighted in bold

Significant improvements over the Vanilla baseline with $p < 0.05$ and $p < 0.01$ are indicated with † and ‡ respectively, for the same cutoff

Significant inferiority with $p < 0.05$ is marked with $*$

For each measure, the improvement rate over the Vanilla baseline is given (%)

Table 21 Reranking effectiveness in the multi-phase fine-tuning setting of the different models on Robust04 and GOV2 collections

Robust04	Title run				Description run				Hybrid run			
Model	nDCG@20		P@20		nDCG@20		P@20		nDCG@20		P@20	
BM25	0.4240	–	0.3631	–	0.4058	–	0.3345	–	0.4240	–	0.3631	–
BM25+RM3	0.4407	–	0.3821	–	0.4255	–	0.3661	–	0.4407	–	0.3821	–
Vanilla $_{\text{BERT}}$	0.4995	–	0.4275	–	0.5368	–	0.4492	–	0.5546	–	0.4715	–
Sim–Doc $_{\text{BERT}}$	0.4976	−0.4%	0.4273	−0.0%	0.5378	+0.2%	0.4470	−0.5%	0.5632	+1.6%	0.4783	+1.4%
Sim-Pair $_{\text{BERT}}$	**0.5058**	+1.3%	**0.4371**	+2.2%	0.5479†	+2.1%	0.4574†	+1.8%	**0.5701** ‡	+2.8%	**0.4815** ‡	+2.1%
Pre-Doc $_{\text{BERT}}$	0.5039	+0.9%	0.4331	+1.3%	0.5462	+1.8%	0.4568	+1.7%	0.5607	+1.1%	0.4757	+0.9%
Pre-Pair $_{\text{BERT}}$	0.5021	+0.5%	0.4333	+1.4%	**0.5532** ‡	+3.1%	**0.4631** ‡	+3.1%	**0.5699** ‡	+2.8%	**0.4821** ‡	+2.2%
GOV2	Title run				Description run				Hybrid run			
Model	nDCG@20		P@20		nDCG@20		P@20		nDCG@20		P@20	
BM25	0.4774	–	0.5362	–	0.4264	–	0.4705	–	0.4774	–	0.5362	–
BM25+RM3	0.4851	–	0.5634	–	0.4212	–	0.4966	–	0.4851	–	0.5634	–
Vanilla $_{\text{BERT}}$	0.5476	–	0.6302	–	0.5175	–	0.5772	–	0.5909	–	0.6604	–
Sim-Doc $_{\text{BERT}}$	0.5413	−1.2%	0.6248	−0.9%	0.5151	−0.5%	0.5799	+0.5%	0.5754	−2.6%	0.6513	−1.4%
Sim-Pair $_{\text{BERT}}$	**0.5743** ‡	+4.9%	**0.6540** ‡	+3.8%	0.5406‡	+4.5%	**0.6084** ‡	+5.4%	**0.5998**	+1.5%	**0.6758**	+2.3%
Pre-Doc $_{\text{BERT}}$	0.5635†	+2.9%	0.6470†	+2.7%	**0.5432** ‡	+5.0%	**0.6074** ‡	+5.2%	**0.6002**	+1.6%	0.6715	+1.7%
Pre-Pair $_{\text{BERT}}$	0.5705‡	+4.2%	0.6513‡	+3.3%	0.5387‡	+4.1%	0.6034‡	+4.5%	0.5966	+1.0%	0.6708	+1.6%

Best results are highlighted in bold

Significant improvements over the Vanilla baseline with $p < 0.05$ and $p < 0.01$ are indicated with † and ‡ respectively

Table 22 Reranking effectiveness in the zero-shot transfer setting of the different models on Robust04 and GOV2 collections

Robust04	Title run				Description run				Hybrid run			
Model	nDCG@20		P@20		nDCG@20		P@20		nDCG@20		P@20	
BM25	0.4240	–	0.3631	–	0.4058	–	0.3345	–	0.4240	–	0.3631	–
BM25+RM3	0.4407	–	0.3821	–	0.4255	–	0.3661	–	0.4407	–	0.3821	–
Top–100												
Vanilla $_{\text{ELECTRA}}$	0.4712	–	0.4108	–	0.4721	–	0.3988	–	0.5103	–	0.4323	–
Sim-Doc $_{\text{ELECTRA}}$	0.4680	−0.7%	0.4054	−1.3%	**0.4804** †	+1.8%	**0.4040**	+1.3%	0.5231‡	+2.5%	**0.4422** †	+2.3%
Sim-Pair $_{\text{ELECTRA}}$	**0.4820** †	+2.3%	**0.4181**	+1.8%	0.4749	+0.6%	0.3964	−0.6%	**0.5235** †	+2.6%	0.4418†	+2.2%
Pre-Doc $_{\text{ELECTRA}}$	0.4663	−1.0%	0.4080	−0.7%	0.4789	+1.4%	0.4016	+0.7%	0.5182	+1.5%	0.4378	+1.3%
Pre-Pair $_{\text{ELECTRA}}$	0.4668	−0.9%	0.4064	−1.1%	0.4740	+0.4%	0.4002	+0.4%	0.5169	+1.3%	0.4416	+2.2%
Top-1000												
Vanilla $_{\text{ELECTRA}}$	0.4416	–	0.3833	–	0.4482	–	0.3831	–	0.4782	–	0.4141	–
Sim-Doc $_{\text{ELECTRA}}$	0.4479	+1.4%	0.3878	+1.2%	0.4640†	+3.5%	**0.3948** †	+3.1%	0.4970‡	+3.9%	0.4247	+2.6%
Sim-Pair $_{\text{ELECTRA}}$	**0.4717** ‡	+6.8%	**0.4124** ‡	+7.6%	0.4597	+2.6%	0.3886	+1.4%	**0.5043** ‡	+5.5%	**0.4263**	+2.9%
Pre-Doc $_{\text{ELECTRA}}$	0.4500	+1.9%	0.3912	+2.1%	**0.4662** †	+4.0%	**0.3948** †	+3.1%	0.4996‡	+4.5%	0.4251	+2.7%
Pre-Pair $_{\text{ELECTRA}}$	0.4511	+2.2%	0.3934	+2.6%	0.4537	+1.2%	0.3878	+1.2%	0.4936†	+3.2%	0.4245	+2.5%

GOV2	Title run				Description run				Hybrid run			
Model	nDCG@20		P@20		nDCG@20		P@20		nDCG@20		P@20	
BM25	0.4774	–	0.5362	–	0.4264	–	0.4705	–	0.4774	–	0.5362	–
BM25+RM3	0.4851	–	0.5634	–	0.4212	–	0.4966	–	0.4851	–	0.5634	–
Top–100												
Vanilla $_{\text{ELECTRA}}$	0.5278	–	0.6094	–	0.5153	–	0.5785	–	**0.5803**	–	**0.6617**	–
Sim-Doc $_{\text{ELECTRA}}$	0.5342	+1.2%	**0.6188**	+1.5%	0.5120	−0.6%	0.5795	+0.2%	0.5761	−0.7%	0.6527	−1.4%
Sim-Pair $_{\text{ELECTRA}}$	**0.5387**	+2.1%	0.6171	+1.3%	**0.5207**	+1.0%	**0.5859**	+1.3%	0.5801	−0.0%	0.6587	−0.5%
Pre-Doc $_{\text{ELECTRA}}$	0.5350	+1.4%	0.6148	+0.9%	0.5086	−1.3%	0.5711	−1.3%	0.5779	−0.4%	0.6557	−0.9%

Table 22 continued

GOV2	Title run				Description run				Hybrid run			
Model	nDCG@20		P@20		nDCG@20		P@20		nDCG@20		P@20	
Pre-Pair $_{\mathrm{ELECTRA}}$	0.5306	+0.5%	0.6131	+0.6%	0.5108	−0.9%	0.5775	−0.2%	0.5760	−0.7%	0.6557	−0.9%
Top-1000												
Vanilla $_{\mathrm{ELECTRA}}$	0.4668	–	0.5332	–	0.4986	–	0.5601	–	0.5147	–	0.5765	–
Sim-Doc $_{\mathrm{ELECTRA}}$	0.4796	+2.7%	0.5530†	+3.7%	0.4958	−0.6%	0.5544	−1.0%	0.5198	+1.0%	**0.5930**	+2.9%
Sim-Pair $_{\mathrm{ELECTRA}}$	**0.4881** †	+4.6%	**0.5577** ‡	+4.6%	**0.5030**	+0.9%	**0.5634**	+0.6%	**0.5249**	+2.0%	0.5923	+2.7%
Pre-Doc $_{\mathrm{ELECTRA}}$	0.4845†	+3.8%	0.5530‡	+3.7%	0.4981	−0.1%	0.5560	−0.7%	0.5212	+1.3%	0.5883	+2.0%
Pre-Pair $_{\mathrm{ELECTRA}}$	0.4820	+3.3%	0.5513†	+3.4%	0.4828*	−3.2%	0.5419	−3.2%	0.5075	−1.4%	0.5711	−0.9%

Best results, for each cutoff, are highlighted in bold

Significant improvements over the Vanilla baseline with $p<0.05$ and $p<0.01$ are indicated with † and ‡ respectively, for the same cutoff

Significant inferiority with $p<0.05$ is marked with $*$

For each measure, the improvement rate over the Vanilla baseline is given (%)

Table 23 Reranking effectiveness in the multi-phase fine-tuning setting of the different models on Robust04 and GOV2 collections

Robust04	Title run				Description run				Hybrid run			
Model	nDCG@20		P@20		nDCG@20		P@20		nDCG@20		P@20	
BM25	0.4240	–	0.3631	–	0.4058	–	0.3345	–	0.4240	–	0.3631	–
BM25+RM3	0.4407	–	0.3821	–	0.4255	–	0.3661	–	0.4407	–	0.3821	–
Vanilla $_{\text{ELECTRA}}$	0.5375	–	0.4560	–	0.5676	–	0.4663	–	0.5901	–	0.4902	–
Sim-Doc $_{\text{ELECTRA}}$	0.5367	−0.1%	0.4560	+0.0%	0.5662	−0.2%	0.4683	+0.4%	0.5893	−0.1%	0.4912	+0.2%
Sim-Pair $_{\text{ELECTRA}}$	0.5380	+0.1%	0.4564	+0.1%	0.5686	+0.2%	**0.4705**	+0.9%	**0.5927**	+0.4%	0.4942	+0.8%
Pre-Doc $_{\text{ELECTRA}}$	0.5338	−0.7%	**0.4590**	+0.7%	**0.5705**	+0.5%	0.4697	+0.7%	0.5889	−0.2%	0.4926	+0.5%
Pre-Pair $_{\text{ELECTRA}}$	**0.5390**	+0.3%	0.4566	+0.1%	0.5677	+0.0%	0.4699	+0.8%	**0.5930**	+0.5%	**0.4970**	+1.4%
GOV2	Title run				Description run				Hybrid run			
Model	nDCG@20		P@20		nDCG@20		P@20		nDCG@20		P@20	
BM25	0.4774	–	0.5362	–	0.4264	–	0.4705	–	0.4774	–	0.5362	–
BM25+RM3	0.4851	–	0.5634	–	0.4212	–	0.4966	–	0.4851	–	0.5634	–
Vanilla $_{\text{ELECTRA}}$	0.5784	–	0.6621	–	**0.5629**	–	**0.6279**	–	0.6149	–	0.6862	–
Sim-Doc $_{\text{ELECTRA}}$	0.5891	+1.8%	0.6685	+1.0%	0.5044*	−10.4%	0.5758*	−8.3%	0.6120	−0.5%	0.6926	+0.9%
Sim-Pair $_{\text{ELECTRA}}$	0.5868	+1.5%	0.6661	+0.6%	0.5552	−1.4%	0.6225	−0.9%	0.6133	−0.3%	**0.6926**	+0.9%
Pre-Doc $_{\text{ELECTRA}}$	0.5841	+1.0%	0.6634	+0.2%	0.5524	−1.9%	0.6188	−1.4%	0.6130	−0.3%	0.6852	−0.1%
Pre-Pair $_{\text{ELECTRA}}$	**0.5920** †	+2.4%	**0.6718**	+1.5%	0.5486*	−2.5%	0.6134*	−2.3%	**0.6207**	+0.9%	**0.6956**	+1.4%

Best results are highlighted in bold

Significant improvements over the Vanilla baseline with $p<0.05$ and $p<0.01$ are indicated with † and ‡ respectively

Significant inferiority with $p<0.05$ is marked with $*$

Table 24 Reranking effectiveness on the TREC DL 2019 and DL 2020 Document ranking tasks

TREC DL Doc	DL 19				DL 20			
Model	nDCG@10		MAP		nDCG@10			
BM25	0.5176	–	0.2434	–	0.5286	–	0.3793	–
BM25+RM3	0.5169	–	0.2772	–	0.5248	–	0.4006	–
Vanilla $_{\text{BERT}}$	0.6726	–	0.3006	–	0.6340	–	**0.4523**	–
Sim-Doc $_{\text{BERT}}$	0.6858	+2.0%	0.3038	+1.1%	0.6340	+0.0%	0.4414	−2.4%
Sim-Pair $_{\text{BERT}}$	0.6798	+1.1%	0.3057	+1.7%	0.6495	+2.4%	0.4505	−0.4%
Pre-Doc $_{\text{BERT}}$	0.6777	+0.8%	**0.3061**	+1.8%	0.6368	+0.4%	0.4513	−0.2%
Pre-Pair $_{\text{BERT}}$	**0.7025** †	+4.4%	0.3018	+1.8%	**0.6498**	+2.5%	0.4497	−0.6%
TREC DL Doc	DL 19				DL 20			
Model	nDCG@10		MAP		nDCG@10			
BM25	0.5176	–	0.2434	–	0.5286	–	0.3793	–
BM25+RM3	0.5169	–	0.2772	–	0.5248	–	0.4006	–
Vanilla $_{\text{ELECTRA}}$	0.6738	–	0.2976	–	0.6236	–	0.4297	–
Sim-Doc $_{\text{ELECTRA}}$	**0.6889**	+2.2%	**0.3082**	+3.6%	0.6369	+2.1%	0.4482†	+4.3%
Sim-Pair $_{\text{ELECTRA}}$	0.6816	+1.2%	0.3062	+2.9%	0.6331	+1.5%	0.4543†	+5.7%
Pre-Doc $_{\text{ELECTRA}}$	0.6801	+0.9%	0.3061	+2.9%	**0.6453**	+3.5%	**0.4582** †	+6.6%
Pre-Pair $_{\text{ELECTRA}}$	0.6763	+0.4%	0.2886	−3.0%	0.6234	−0.0%	0.4306	+0.2%

Best performances are highlighted in bold

Significant improvements over the vanilla baseline with $p < 0.05$ are indicated with †, for the same core

Change rate over the vanilla baseline for the same core type are reported for each metric (%)

the multi-phase fine-tuning setting that uses the same reranking threshold of 100 in Table 23.

Appendix 2.2 Multi-phase fine-tuning setting

Table 23 results complements the results presented in Table 15 obtained in the multi-phase fine-tuning setting using all exact match marking strategies proposed in this paper.

Appendix 3. TREC deep learning track document ranking task

Table 24 reports the results of all our models with both BERT and ELECTRA cores on the TREC DL 2019 and 2020 Document ranking tasks.

References

Akkalyoncu Yilmaz, Z., 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-IJCN)*, ACL, Hong Kong, China, (pp. 3488–3494).

Boualili, L., Moreno, J. G., & Boughanem, M. (2020). *MarkedBERT: Integrating traditional IR cues in pre-trained language models for passage retrieval* (pp. 1977–1980). New York, NY, USA: Association for Computing Machinery.

Câmara, A., & Hauff, C. (2020). Diagnosing bert with retrieval heuristics. In J. M. Jose, E. Yilmaz, J. Magalhães, P. Castells, N. Ferro, M. J. Silva, & F. Martins (Eds.), *Advances in information retrieval* (pp. 605–618). Cham: Springer International Publishing.

Chen, X., Li, C., He, B., & Sun, Y. (2019). UCAS at TREC-2019 deep learning track. In Voorhees EM, Ellis A (eds) *Proceedings of the Twenty-Eighth Text REtrieval Conference, TREC 2019*, (Vol. 1250). 2019, National Institute of Standards and Technology (NIST), NIST Special Publication: Gaithersburg, Maryland, USA.

Chen, X., He, B., Sun, L., & Sun, Y. (2020). ICIP at TREC-2020 deep learning track. In Voorhees EM, Ellis A (eds) *Proceedings of the Twenty-Ninth Text REtrieval Conference, TREC 2020*, (Vol. 1266) . National Institute of Standards and Technology (NIST), NIST Special Publication: Gaithersburg, Maryland, USA.

Clark, K., Luong, M.T., Le, Q.V., & Manning, C.D. (2020). Electra: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations*.

Craswell, N., Mitra, B., Yilmaz, E., Campos, D., & Voorhees, E.M. (2020). *Overview of the TREC 2019 deep learning track*. arXiv:2003.07820

Craswell, N., Mitra, B., Yilmaz, E., & Campos, D. (2021). *Overview of the trec 2020 deep learning track*. arXiv:2102.07662

Dai, Z., & Callan, J. (2019a). *Context-aware sentence/passage term importance estimation for first stage retrieval*. arXiv preprint arXiv:1910.10687

Dai, Z., & Callan, J. (2019b). Deeper text understanding for ir with contextual neural language modeling. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, (pp. 985–988).

Dai, Z., & Callan, J. (2020a). Context-aware document term weighting for ad-hoc search. In *Proceedings of The Web Conference 2020*, Association for Computing Machinery: New York, NY, USA, (pp. 1897-1907).

Dai, Z., & Callan, J. (2020b). Context-aware term weighting for first stage passage retrieval. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. Association for Computing Machinery: New York, NY, USA, (pp. 1533-1536).

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 Eleventh ACM International Conference on Web Search and Data Mining*, Association for Computing Machinery: New York, NY, USA, WSDM '18, (pp. 126-134)

Devlin, J., Chang, M., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 NAACL-HLT Conference*, *1*, 4171–4186.

Guo, J., Fan, Y., Ai, Q., & Croft, W. (2016). A deep relevance matching model for ad-hoc retrieval. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, (pp. 55–64).

Humeau, S., Shuster, K., Lachaux, M.A., & Weston, J. (2020). Poly-encoders: Architectures and pre-training strategies for fast and accurate multi-sentence scoring. In *International Conference on Learning Representations*.

Johnson, J., Douze, M., & Jégou, H. (2017). *Billion-scale similarity search with gpus*. arXiv:1702.08734

Karpukhin, V., Oguz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., & Yih, W.t. (2020). Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Association for Computational Linguistics, Online, (pp. 6769–6781).

Khattab, O., & Zaharia, M. (2020). *ColBERT: Efficient and effective passage search via contextualized late interaction over BERT* (pp. 39–48). New York, NY, USA: Association for Computing Machinery.

Kingma, D.P., & Ba, J. (2015) Adam: A method for stochastic optimization. In: Bengio Y, LeCun Y (eds) *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Lavrenko, V., & Croft, W.B. (2001). Relevance based language models. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Association for Computing Machinery: New York, NY, USA, SIGIR '01, (pp. 120-127).

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

Li, H. (2011). *Learning to Rank for Information Retrieval and Natural Language Processing*. Morgan & Claypool Publishers.

Lin, J., Nogueira, R., & Yates, A. (2020). *Pretrained transformers for text ranking: BERT and beyond*. arXiv:2010.06467

Liu, T. Y. (2009). Learning to rank for information retrieval. *Foundations and Trends in Information Retrieval, 3*(3), 225–331.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). *Roberta: A robustly optimized BERT pretraining approach*. arXiv:1907.11692

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

MacAvaney, S., Yates, A., Cohan, A., & Goharian, N. (2019). 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).

MacAvaney, S., Feldman, S., Goharian, N., Downey, D., & Cohan, A. (2020a). *ABNIRML: analyzing the behavior of neural IR models*. arXiv:2011.00696

MacAvaney, S., Nardini, F.M., Perego, R., Tonellotto, N., Goharian, N., & Frieder, O. (2020b). Efficient document re-ranking for transformers by precomputing term representations. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, Association for Computing Machinery: New York, NY, USA, (pp. 49-58).

Mitra, B., Diaz, F., & Craswell, N. (2017). Learning to match using local and distributed representations of text for web search. In *Proceedings of the 26th International Conference on World Wide Web*. (pp. 1291–1299).

Mitra, B., Craswell, N., et al. (2018). An introduction to neural information retrieval. *Foundations and Trends in Information Retrieval, 13*(1), 1–126.

Nguyen, T., Rosenberg, M., Song, X., Gao, J., Tiwary, S., Majumder, R., & Deng, L. (2016). *MS MARCO: A human generated machine reading comprehension dataset*. arXiv:1611.09268

Nogueira, R., & Cho, K. (2019). *Passage re-ranking with BERT*. arXiv:1901.04085,

Nogueira, R., Lin, J., & Epistemic, A. (2019). *From doc2query to docttttquery*. Online preprint 6.

Nogueira, R., Jiang, Z., Pradeep, R., & Lin, J. (2020). Document ranking with a pretrained sequence-to-sequence model. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, Association for Computational Linguistics, Online, (pp. 708–718).

Onal, K. D., Zhang, Y., Altingovde, I. S., Rahman, M. M., Karagoz, P., Braylan, A., Dang, B., Chang, H. L., Kim, H., Mcnamara, Q., Angert, A., Banner, E., Khetan, V., Mcdonnell, T., Nguyen, A. T., Xu, D., Wallace, B. C., Rijke, M., & Lease, M. (2018). Neural information retrieval: At the end of the early years. *Information Retrieval Journal, 21*(2–3), 111–182.

Padaki, R., Dai, Z., & Callan, J. (2020). Rethinking query expansion for bert reranking. In J. M. Jose, E. Yilmaz, J. Magalhães, P. Castells, N. Ferro, M. J. Silva, & F. Martins (Eds.), *Advances in Information Retrieval* (pp. 297–304). Cham: Springer International Publishing.

Padigela, H., Zamani, H., & Croft, W.B. (2019). *Investigating the successes and failures of BERT for passage re-ranking*. arXiv:1905.01758

Pradeep, R., Ma, X., Zhang, X., Cui, H., Xu, R., Nogueira, R., & Lin, J. (2020). H2oloo at trec 2020: When all you got is a hammer... deep learning, health misinformation, and precision medicine. In *Text Retrieval Conference (TREC)*.

Qiao, Y., Xiong, C., Liu, Z., & Liu, Z. (2019). *Understanding the behaviors of BERT in ranking*. arXiv:1904.07531

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., & Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research, 21*(140), 1–67.

Reimers, N., & Gurevych, I. (2019). 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. arXiv:1908.10084

Reimers, N., & Gurevych, I. (2020). Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics*.

Rennings, D., Moraes, F., & Hauff, C. (2019). An axiomatic approach to diagnosing neural ir models. In L. Azzopardi, B. Stein, N. Fuhr, P. Mayr, C. Hauff, & D. Hiemstra (Eds.), *Advances in Information Retrieval* (pp. 489–503). Cham: Springer International Publishing.

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*, (pp. 5998–6008).

Voorhees, E., Alam, T., Bedrick, S., Demner-Fushman, D., Hersh, W.R., Lo, K., Roberts, K., Soboroff, I., & Wang, L.L. (2021). Trec-covid: Constructing a pandemic information retrieval test collection. SIGIR Forum 54(1).

Wang, W., Bi, B., Yan, M., Wu, C., Xia, J., Bao, Z., Peng, L., Si, L. (2020). Structbert: Incorporating language structures into pre-training for deep language understanding. In *International Conference on Learning Representations*.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Le Scao, T., Gugger, S., Drame, M., Lhoest, Q., & Rush, A. (2020). Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, Association for Computational Linguistics, Online, (pp. 38–45).

Xiong, C., Dai, Z., Callan, J., Liu, Z., & Power, R. (2017). End-to-end neural ad-hoc ranking with kernel pooling. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, Association for Computing Machinery, New York, NY, USA, SIGIR '17,(pp. 55-64).

Xiong, L., Xiong, C., Li, Y., Tang, K., Liu, J., Bennett, P.N., Ahmed, J., & Overwijk, A. (2021). Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *9th International Conference on Learning Representations, ICLR 2021*, Virtual Event, Austria, May 3-7, 2021, OpenReview.net.

Yan, M., Li, C., Xia, J., & Wang, W. (2019). Idst at trec 2019 deep learning track: Deep cascade ranking with generation-based document expansion and pre-trained language modeling. In:*TREC*.

Yang, P., Fang, H., & Lin, J. (2017). Anserini: Enabling the use of lucene for information retrieval research. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, (pp. 1253–1256).

Yang, W., Lu, K., Yang, P., & Lin, J. (2019a). Critically examining the "neural hype" weak baselines and the additivity of effectiveness gains from neural ranking models. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, (pp. 1129–1132).

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

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.