MarkedBERT: Integrating Traditional IR Cues in Pre-trained Language Models for Passage Retrieval

Lila Boualili
lila.boualili@irit.fr
IRIT, University of Paul Sabatier
Toulouse, France

Jose G. Moreno
jose.moreno@irit.fr
IRIT, University of Paul Sabatier
Toulouse, France

Mohand Boughanem
mohand.boughanem@irit.fr
IRIT, University of Paul Sabatier
Toulouse, France

ABSTRACT

The Information Retrieval (IR) community has witnessed a flourishing development of deep neural networks, however, only a few managed to beat strong baselines. Among them, models like DRMM and DUET were able to achieve better results thanks to the proper handling of exact match signals. Nowadays, the application of pre-trained language models to IR tasks has achieved impressive results exceeding all previous work. In this paper, we assume that established IR cues like exact term-matching, proven to be valuable for deep neural models, can be used to augment the direct supervision from labeled data for training these pre-trained models. To study the effectiveness of this assumption, we propose MarkedBERT a modified version of one of the most popular pre-trained models via language modeling tasks, BERT. MarkedBERT integrates exact match signals using a marking technique that locates and highlights *Exact Matched* query-document terms using marker tokens. Experiments on MS MARCO *Passage Ranking* task show that our rather simple approach is actually effective. We find that augmenting the input with marker tokens allows the model to focus on valuable text sequences for IR.

KEYWORDS

Deep Learning; Passage Retrieval; Exact Matching

ACM Reference Format:
Lila Boualili, Jose G. Moreno, and Mohand Boughanem. 2020. MarkedBERT: Integrating Traditional IR Cues in Pre-trained Language Models for Passage Retrieval. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '20), July 25–30, 2020, Virtual Event, China.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3397271.3401194

1 INTRODUCTION

Neural models exploiting massive pre-training on language modeling tasks, such as BERT [2], are becoming basic tools in the domain of Natural Language Processing (NLP). The same trend is witnessed in the IR community, where several works emerged applying these models to IR tasks such as document retrieval [1, 5, 9], passage ranking [8], and question answering [17].

Nowadays, in *ad hoc* retrieval, the prevalent approach consists in deploying deep pre-trained models like BERT as re-rankers over an initial list of candidate documents retrieved using a traditional bag-of-words *term-matching* model [1]. These neural re-rankers ignore the term-matching signals that led to the retrieval of the candidate documents in the first stage, and thus, ignoring prior work in neural IR [3] that highlighted the importance of these signals for ranking. We thus hypothesis that IR cues – used in traditional IR methods and proven to be useful for neural models [3]– such as *Exact Matching* may also be beneficial for pre-trained models.

We propose considering the *Exact Match* cue to enhance the results of BERT for the passage retrieval task. Inspired by a recent work for relation extraction [13] where the use of marker tokens to highlight the entities (key tokens) highly improved BERT's performance, we propose a BERT-based model that highlights exact matched terms for the passage ranking task namely: MarkedBERT. This model helps to focus on the terms that are important for the relevance evaluation process.

Our experiments on the MS MARCO passage ranking task show that term-matching highlight is effective at improving the MRR@10 by about 9% when compared to a strong baseline. Our code is available for replication and future work[1].

2 RELATED WORK

With the advent of deep learning, the IR community witnessed a flourishing development of neural ranking models, such as DRMM [3], DUET [7], KNRM [15] and Co-PACRR [4]. [6] provide a recent overview of many of these models. This development helped the researchers to realise that *relevance matching* and *semantic matching* (e.g: paraphrase detection) are different tasks [3]. 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, e.g: in [7], 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. In spite of all the work invested in the development of neural ranking models, a recent study conducted over a 100 papers by [16] showed that most models failed against strong non-neural baselines. More recently, a new generation of pre-trained neural models via language modeling has achieved state-of-the-art results in many NLP tasks. Several works proposed using the most popular model BERT [2] for *ad hoc* retrieval. Nogueira et al. [8] report one of the first successful applications of BERT to passage re-ranking. Later studies took this work as a starting point for more development. In [10], the authors present the first use of document expansion based on neural networks that allowed the achievement

[1] https://github.com/BOUALILILila/markers_bert

SIGIR '20, July 25–30, 2020, Virtual Event, China

ACM ISBN 978-1-4503-8016-4/20/07…$15.00
https://doi.org/10.1145/3397271.3401194

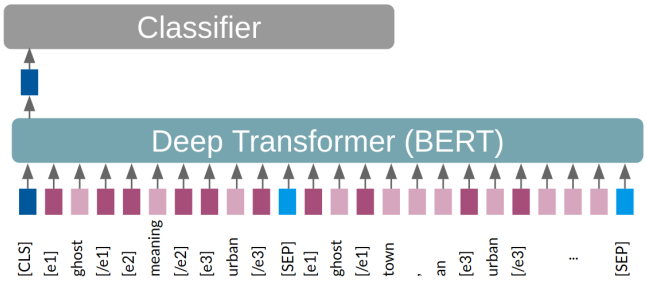


Figure 1: Architecture of MarkedBERT applied on the marked query Q:"$[e_1]$ ghost $[/e_1]$ $[e_2]$ meaning $[/e_2]$ $[e_3]$ urban $[/e_3]$", and passage P:"$[e_1]$ ghost $[/e_1]$ town, an $[e_3]$ urban $[/e_3]$ area with a fixed boundary that is smaller than a city ".

of new state-of-the-art results on MS MARCO leader board[2] at that time. Furthermore, studies such as [11] and [12] tried to understand the behaviors of BERT that led to its success and failure in passage ranking. Though this models achieve unprecedented success, they ignore the incorporation of exact matching signals regardless of their proven importance for ranking. In the light of this fact, we investigate whether these signals can benefit pre-trained neural models.

3 METHODOLOGY

We tackle the ad hoc passage ranking problem where passages are ranked for a given query according to their relevance score.

3.1 Base model

BERT is a multi-layer bidirectional Transformer encoder based on the original implementation described in [14]. It is *pre-trained* on a large corpus of unlabeled data using two unsupervised tasks: Masked Language Model (MLM) and Next Sentence Prediction (NSP). BERT relies on a fine-tuning approach where minimal task specific parameters are fine-tuned together with the pre-trained parameters for the downstream task. Fine-tuning is straightforward thanks to the self-attention mechanism that allows BERT to model many downstream tasks whether they involve single text or text pairs [2]. Indeed, it can encode multiple text segments using two special tokens ([SEP] and [CLS]), and using segment embeddings allowing the distinction between the *Segment A* and *Segment B*. While [SEP] separates the two segments, the [CLS] token is used for making judgements about the text pairs. [CLS] is pre-trained on the NSP task to predict whether *Segment A* immediately precedes *Segment B* in the original text. It can be further *fine-tuned* inexpensively on a downstream task like Passage Ranking.

The BERT-based model of [8] provides the starting point of our work. In this approach, BERT is fine-tuned using a single additional layer to perform passage-level point-wise classification. The model takes the query Q as *Segment A* and the passage P as *Segment B* to build the input sequences of BERT: $S = [[CLS], Q, [SEP], P, [SEP]]$. We follow the classical strategy that consists in fine-tuning the [CLS] token for relevance classification.

[2]https://microsoft.github.io/msmarco/

3.2 MarkedBERT

Our strategy consists in augmenting the input sequence S with special tokens to mark the terms that match exactly in Q and P. We introduce the tokens $[e_k]$ and $[/e_k]$ ($k = \{1, ..., |Q|\}$) that mark the start and the end of each occurrence of the query terms[3] in P. These markers identify query terms, if a term is repeated in the query it will have the same identifier as the first occurrence. Let us consider a query $Q = \{q_1, \ldots, q_{|Q|}\}$ and a passage $P = \{p_1, \ldots, p_{|P|}\}$, if $\{p_i, p_j\}$ are occurrences of q_n and p_l is the only occurrence of q_m in P with $n < m, i < j < l$, the augmented input sequence $\tilde{S}$ is then:

$$\tilde{S} = [[CLS], \ldots, [e_n], q_n, [/e_n], \ldots, [e_m], q_m, [/e_m], \ldots, [SEP], \ldots, [e_n], p_i, [/e_n], \ldots, [e_n], p_j, [/e_n], \ldots, [e_m], p_l, [/e_m], \ldots, [SEP]]$$

For readability, q_i/p_j refers to q_i/p_j word piece tokens.

Figure 1 presents the architecture of MarkedBERT. The standard classification token $[CLS]$ is used to determine whether the passage P is relevant or not w.r.t the query Q.

4 EXPERIMENTAL SETUP

4.1 Dataset

Microsoft MAchine Reading COmprehension dataset (MS MARCO) is a large scale dataset obtained from no less than half a million queries sampled from Bing's search query logs. Each query is associated with sparse relevance judgments by human editors. Our work focuses on the passage ranking dataset which contains over than $8.8M$ passages. The training set contains approximately $400M$ triples of a query, relevant and non-relevant passages with approximately $500K$ relevant pairs. We use the small training set that comprises a subset of $40M$ triples with $400K$ relevant query-passage pairs. Each query has, on average, one relevant passage. The development set contains 6980 queries. The official metric for this dataset is MRR@10 on the development and test set. The test set is not publicly available.

4.2 Training

We use the base version (12 layers, 768 hidden size, 12 heads, 110M parameters) of BERT ($\text{BERT}_{\text{base}}$) due to hardware limitations. We fine-tune both the Base model and MarkedBERT with a batch-size of 32 and the maximum sequence length ($32\ sequences \times 512\ tokens = 16{,}384\ tokens/batch$) for 2 epochs. To avoid biasing the model towards predicting non-relevant labels, that are approximately 95 times more frequent in the training set after discarding redundant query-passage pairs, we sample an equal amount of relevant and non-relevant pairs to obtain a balanced training set as suggested by [9]. We use Adam optimizer with the initial learning rate set to 3×10^{-6} and linear decay of the learning rate. The drop out rate is set to 0.1 for all our experiments. We use an open source implementation of BERT by Hugging Face[4].

4.3 Inference

We use a two-stage cascade ranking pipeline. The first stage produces a top-1000 candidate passages per query using BM25[5]. Prior to indexing, we expand each passage with a set of generated queries

[3]In our experiments, a query term with no occurrences in the passage is not marked.
[4]https://github.com/huggingface/transformers
[5]https://github.com/castorini/anserini

Table 1: MRR@10 percentage of the Base model and MarkedBERT on the MS MARCO development set. † Indicates statistical significance w.r.t the Base model.

Model	MRR@10
BM25 [10]	18.4
Doc2query + BM25 [10]	22.1
Base model (Ours)	30.2
MarkedBERT	32.8†

Table 2: Average MRR@10 percentage per query length (QL). D: Doc2query + BM25, B: Base, M: MarkedBERT, #: Query count, d1, d2(%): improvement of MarkedBERT over Doc2query + BM25 and Base model respectively.

QL	2	3	4	5	6	7	8	9	10
#	273	621	1047	1156	1053	759	506	360	489
D	27.8	24.2	24.5	25.5	27.6	22.4	24.9	21.5	21.2
B	38.1	36.5	36.0	33.7	34.5	32.0	31.9	31.1	26.3
M	41.7	40.7	39.5	37.3	37.2	33.5	34.0	33.2	28.6
d1	50.2	68.1	60.7	46.7	34.8	49.8	36.7	54.8	35.1
d2	09.5	11.6	09.5	11.2	07.8	04.7	06.6	06.7	08.8

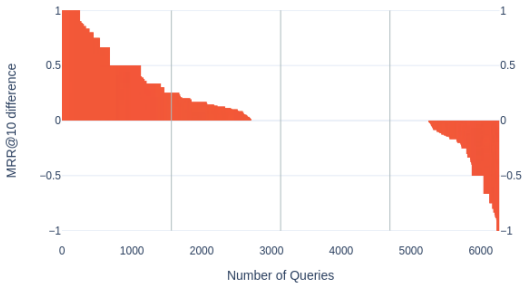
using Doc2query [10] trained on MS MARCO data to overcome the vocabulary mismatch problem. In the second stage, we re-rank the candidate passages (without expansion) using the Base model and MarkedBERT.

5 RESULTS AND DISCUSSION

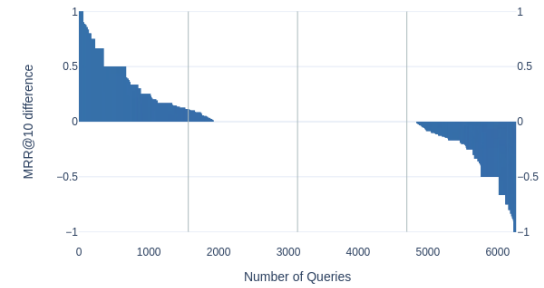
The performance of the Base model and MarkedBERT on the entire development set of MS MARCO is shown in Table 1. Note that the dev set was not used during training. Both BERT-based models outperform the existing traditional retrieval models by a large margin. Interestingly, MarkedBERT significantly outperforms the Base model by about 9%. In order to understand the improvements, we analyse the performances of the two BERT-based models on the development set.

After the initial retrieval using Doc2query + BM25, only 6264 out of the original 6980 queries had their relevant passage in the top-1000 candidates list. Thus we only consider the subset containing these 6264 queries in our following analysis.

Per Query Analysis. We analyze the per query performance of MarkedBERT compared to the first stage ranker Doc2query + BM25 and the Base model. Figure 2 reports the ΔMRR per query, sorted in descending order. (a) MarkedBERT vs. Doc2query + BM25: Among the 6264 queries, 2708 (43.23%) gain in performance with the exact term-matching highlight while for 1012 (16.15%)queries, this highlight causes a deterioration in performance, for 2544 queries, both models perform similarly. (b) MarkedBERT vs. Base: 1927 (30.76%) queries gain in performance with the exact term-matching highlight while for 1428 (22.79%) others, it causes a deterioration in performance, for 2909 queries, both base and MarkedBERT model perform similarly.



(a) MRR@10$_{\text{MarkedBERT}}$ − MRR@10$_{\text{Doc2query + BM25}}$



(b) MRR@10$_{\text{MarkedBERT}}$ − MRR@10$_{\text{Base}}$

Figure 2: Per query MRR@10 difference on MS MARCO Dev set of MarkedBERT compared to (a) Doc2query + BM25 ranker, (b) the Base model. Vertical lines indicating 25%, 50% and 75% of the queries were added for visual readability.

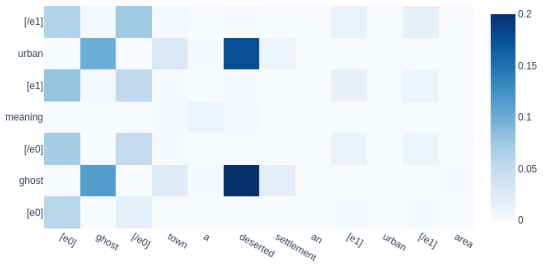
Per Query Length Analysis. Since our approach is based on augmenting the input sequence with marker tokens, we want to evaluate the impact of the marking technique on longer queries. Table 2 reports the average *MRR*@10 per query length for Doc2query + BM25, Base model and MarkedBERT. We notice that MarkedBERT performs better across all query lengths. However, these improvements ($d1$ and $d2$) tend to decrease as the query length increases from 2 to 10. The improvement $d1$ of MarkedBERT over the Base model goes from an average of 10.4% for small queries ($QL < 6$) to 6.9% for longer queries ($QL \geq 6$). The Doc2query + BM25 model is the least sensitive to query length with a decrease in performance of only 23.9%. For the BERT-based models, the decrease in performance is slightly more prominent for MarkedBERT (31.4%) compared to the Base model (30.9%). This result is consistent with the the findings in [11] where the authors compared BERT to simple BM25 and found that BERT performs better for shorter queries. This is an encouraging result since, in average, adding marker tokens does not hurt the performance of longer queries, since BERT does not perform well for these queries originally.

Per Query Type Analysis. In order to understand the performance improvement of MarkedBERT across different query types, we follow the comparison method used in [11]. We classify the queries based on the lexical answer type using a rule-based classifier[6]. The queries are classified into 6 possible answer types, namely numerical, human, location, description, entity and abbreviation. Table 3 report the average *MRR*@10 across these 6 query answer

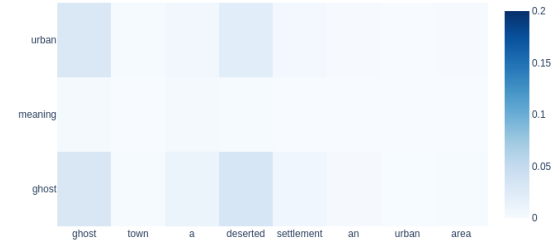
[6]https://github.com/superscriptjs/qtypes

Table 3: Average MRR@10 percentage per query answer type. D2q refers to Doc2query.

Type	NUM	HUM	LOC	DESC	ENTY	ABBR
#Queries	1039	500	527	2063	363	9
D2q+BM25	22.2	25.1	32.8	22.6	23.2	36.1
Base	29.1	35.6	42.0	33.2	30.5	49.4
Marked	33.0	38.4	44.6	35.0	31.5	47.9



(a) MarkedBERT



(b) Base model

Figure 3: Attention map head between the query: "ghost meaning urban" and the passage: "ghost town a deserted settlement an urban area ..." tokens in the head 6 from layer 12 of MarkedBERT (a) and the Base model(b).

types for the 4501 queries that have a valid answer type. We notice that adding exact term-match highlight benefits the most *numerical* type queries while it slightly deteriorates the performance of *abbreviation* type queries compared to the Base model.

Attention Analysis. The intuition behind introducing marker tokens in the input is to bring focus on the exact matched-terms. To check this, we visualize the attention map on the same layer and head of both MarkedBERT and the Base model for the query "ghost meaning urban" and its relevant passage "ghost town a deserted settlement an urban area ..." shown in Figure 3. Darker color indicates higher attention weight. For this example, the Base model without marker tokens could not retrieve the passage while the MarkedBERT retrieved it in the first rank ($\Delta MRR@10 = 1$). We notice that in the absence of the marker tokens, the Base model attends broadly to the whole sequence. Even if it captures some relationships between the key words "ghost" and "urban" the weights are really low (clear color), and the occurrence of "urban" in the passage is totally ignored. Whereas, in the presence of marker tokens, MarkedBERT captures interesting relationships between the key words with higher weights but it also puts focus on the marker tokens themselves. This new capability allows the model to focus on the occurrence of the word " urban" in the passage (when the Base model failed) through the marker tokens surrounding it even if the weights are lower. Moreover, MarkedBERT captures a strong semantic relationship between the query terms "ghost" and "urban" with the passage term "deserted"(empty of people[7]).

6 CONCLUSION

In this paper, we demonstrate that a traditional IR cue such as *Exact Term-Matching* can help a state-of-the-art re-ranking model based on BERT to achieve even better results. We proposed MarkedBERT that incorporates Exact Matching signals via a simple yet effective marking technique that only modifies the model input. Our analysis confirms that introducing marker tokens induces more focus on the exact matched terms that we consider important to the ranking task and thus improve BERT's relevance evaluation. This way, we can hypothesis that the marking technique can be used to attract attention on different tokens and provide BERT-like models with additional information to accomplish several tasks.

This study is encouraging future work on (a) using the same marking technique for other tasks, (b) further adapting advanced neural models to IR. We are looking forward to extending our work using a wide range of other IR cues.

REFERENCES

[1] Z. Akkalyoncu Yilmaz, W. Yang, H. Zhang, and J. Lin. 2019. Cross-Domain Modeling of Sentence-Level Evidence for Document Retrieval. In *Proc. of the 2019 EMNLP-IJCNLP Conf.* ACL, Hong Kong, China, 3488–3494.
[2] J. Devlin, M. Chang, K. Lee, and K. Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proc. of the 2019 NAACL-HLT Conf., Volume 1*. ACL, 4171–4186.
[3] J. Guo, Y. Fan, Q. Ai, and W. Croft. 2016. A deep relevance matching model for ad-hoc retrieval. In *Proc. of the 25th CIKM Conf.* 55–64.
[4] K. Hui, A. Yates, K. Berberich, and G. De Melo. 2018. Co-PACRR: A context-aware neural IR model for ad-hoc retrieval. In *Proc. of the 11th WSDM Conf.* 279–287.
[5] S. MacAvaney, A. Yates, A. Cohan, and N. Goharian. 2019. CEDR: Contextualized embeddings for document ranking. In *Proc. of the 42nd SIGIR Conf.* 1101–1104.
[6] B. Mitra, N. Craswell, and others. 2018. An introduction to neural information retrieval. *Foundations and Trends® in Information Retrieval* 13, 1 (2018), 1–126.
[7] B. Mitra, F. Diaz, and N. Craswell. 2017. Learning to match using local and distributed representations of text for web search. In *Proc. of the 26th WWW Conf.* 1291–1299.
[8] R. Nogueira and K. Cho. 2019. Passage Re-ranking with BERT. *CoRR* (2019). arXiv:1901.04085
[9] R. Nogueira, W. Yang, K. Cho, and J. Lin. 2019. Multi-Stage Document Ranking with BERT. arXiv:cs.IR/1910.14424
[10] R. Nogueira, W. Yang, J. Lin, and K. Cho. 2019. Document Expansion by Query Prediction. *CoRR* (2019). arXiv:1904.08375
[11] H. Padigela, H. Zamani, and W. Croft. 2019. Investigating the successes and failures of BERT for passage re-ranking. arXiv:1905.01758
[12] Yifan Qiao, Chenyan Xiong, Zhenghao Liu, and Zhiyuan Liu. 2019. Understanding the Behaviors of BERT in Ranking. arXiv:1904.07531
[13] L. Soares, N. FitzGerald, J. Ling, and T. Kwiatkowski. 2019. Matching the Blanks: Distributional Similarity for Relation Learning. In *Proc. of the 57th ACL Conf.* 2895–2905.
[14] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, and I. Polosukhin. 2017. Attention is all you need. In *Advances in NIPS*. 5998–6008.
[15] C. Xiong, Z. Dai, J. Callan, Z. Liu, and R. Power. 2017. End-to-end neural ad-hoc ranking with kernel pooling. In *Proc. of the 40th SIGIR Conf.* 55–64.
[16] W. Yang, K. Lu, P. Yang, and J. Lin. 2019. Critically Examining the" Neural Hype" Weak Baselines and the Additivity of Effectiveness Gains from Neural Ranking Models. In *Proc. of the 42nd SIGIR Conf.* 1129–1132.
[17] W. Yang, Y. Xie, A. Lin, X. Li, L. Tan, K. Xiong, M. Li, and J. Lin. 2019. End-to-End Open-Domain Question Answering with BERTserini. In *Proc. of the 2019 NAACL Conf. (Demonstrations)*. ACL, Minneapolis, Minnesota, 72–77.

[7] https://dictionary.cambridge.org/dictionary/english/deserted