Match Your Words! A Study of Lexical Matching in Neural Information Retrieval

Thibault Formal[1,2(✉)], Benjamin Piwowarski[2,3], and Stéphane Clinchant[1]

[1] Naver Labs Europe, Meylan, France
{thibault.formal,stephane.clinchant}@naverlabs.com
[2] Sorbonne Université, Institute for Intelligent Systems and Robotics, UMR 7222, Paris, France
benjamin@piwowarski.fr
[3] CNRS, Paris, France

Abstract. Neural Information Retrieval models hold the promise to replace lexical matching models, e.g. BM25, in modern search engines. While their capabilities have fully shone on in-domain datasets like MS MARCO, they have recently been challenged on out-of-domain zero-shot settings (BEIR benchmark), questioning their actual generalization capabilities compared to bag-of-words approaches. Particularly, we wonder if these shortcomings could (partly) be the consequence of the inability of neural IR models to perform lexical matching off-the-shelf. In this work, we propose a measure of discrepancy between the lexical matching performed by any (neural) model and an "ideal" one. Based on this, we study the behavior of different state-of-the-art neural IR models, focusing on whether they are able to perform lexical matching *when it's actually useful*, i.e. for important terms. Overall, we show that neural IR models fail to properly generalize term importance on out-of-domain collections or terms almost unseen during training.

Keywords: Neural Information Retrieval · BERT · Lexical matching

1 Introduction

Over the last two years, the effectiveness of neural IR systems has risen substantially. Neural retrievers based on pre-trained Language Models like BERT [4] – whether dense or sparse – hold the promise to replace lexical matching models (e.g. BM25) for first-stage ranking in modern search engines. Despite this success, little is known regarding their actual inner working in the IR setting. Previous works scrutinizing BERT-based ranking models either relied on axiomatic approaches adapted to neural models [1,17], controlled experiments [11], or direct investigation of the learned representations [7,9] or attention [19]. This line of work has shown – among other findings – that these models, which rely on contextualized semantic matching, are actually still quite sensitive to lexical match and term statistics in documents/collections [7,9]. However, these observations

M. Hagen et al. (Eds.): ECIR 2022, LNCS 13186, pp. 120–127, 2022.
https://doi.org/10.1007/978-3-030-99739-7_14

are based on specifically tailored approaches that cannot directly be applied to any given model. To generalize these findings, we introduce instead an intuitive black box approach: we propose to "count" query terms appearing in top documents retrieved by various state-of-the-art neural systems, in order to compare their ability to perform *lexical matching*.

Furthermore, previous studies have been conducted on the MS MARCO dataset, on which models have been trained. The BEIR benchmark [18] has shown that the only systems improving the overall performance over BM25 in the zero-shot setting have (somehow) a lexical bias, e.g. models like doc2query-T5 [13] or ColBERT [10]. Therefore, we also propose to study the extent to which neural IR models are able to *generalize* lexical matching, for query terms that either have not been seen in the training set or with different collection statistics (e.g. common in the training set but rare on an out-of-domain evaluation set).

In this work, we first develop indicators that help measuring to what extent a lexical match is "important" for the user (user relevance) or for the model (system relevance). By comparing both values – i.e. computing the difference between the user and the system, we can look at the following research questions:

(RQ1). To what extent neural retrievers perform accurate lexical matching (Sect. 3.1)? **(RQ2).** Do they generalize term matching to unseen query terms (Sect. 3.1)? **(RQ3).** Do they generalize term matching to new collections (Sect. 3.2)?

2 Methodology

Our analysis rationale is the following: the more a term is important for a query (w.r.t. relevant documents), the more frequent the term should be retrieved by the system in top retrieved documents. Therefore, we first need to define what it means for a term to be *important for lexical matching*, and how to accurately measure frequency in top documents. Roughly speaking, we are interested in the models ability to retrieve documents containing query terms, *when they are deemed important*. Note that we are not interested in expansion mechanisms in our analysis since they are more related to semantic matching.

Intuitively, term importance w.r.t. relevance can be measured by the extent to which a term allows to distinguish relevant from non-relevant documents in a collection of documents. It is thus natural to use the Robertson-Sparck Jones (RSJ) weight [14,20]. The RSJ weights have been shown, if estimated correctly, to order documents in the optimal order w.r.t. the Probability Ranking Principle [15]. For a given user information need U, the user RSJ_U weight for term t is defined as follows (the conditioning on query q is implicit):

$$\text{RSJ}_{t,U} = \log \frac{p(t|R)p(\neg t|\neg R)}{p(\neg t|R)p(t|\neg R)} \tag{1}$$

where $P(t|R)$ is the probability that term t occurs in a relevant document. $\text{RSJ}_{t,U}$ is thus high when a term, for a document to be relevant, is both *necessary* ($p(.|R)$) and *sufficient* ($p(.|\neg R)$). Intuitively, it is low for e.g. stopwords, as they have equal *odds* to appear in relevant and irrelevant documents. The

above weight can be estimated using the set of relevant documents and collection statistics.

We now want to compute the same weight, when relevance is defined by the *system* (and not the *user*). In other words, we would like to measure how much a model "retrieves" term t. One way to proceed is to suppose that top-K documents are *relevant from the point of view of the system*, for a suitable K. While a more accurate definition of system relevance could be used, we found out in our preliminary analysis that results were not very sensitive to the choice of K. We hence define the system RSJ_S weight for term t as:

$$\text{RSJ}_{t,S} = \log \frac{p(t|\text{top-}K)p(\neg t|\neg\text{top-}K)}{p(\neg t|\text{top-}K)p(t|\neg\text{top-}K)} \tag{2}$$

Intuitively, it gives us a mean to properly count occurrences of query terms in retrieved documents – taking into account collection statistics. It is estimated similarly to Eq. 1. Once RSJ_U and RSJ_S have been computed, we can look at the difference between both, i.e. $\Delta\text{RSJ}_t = \text{RSJ}_{t,S} - \text{RSJ}_{t,U}$. If $\Delta\text{RSJ}_t > 0$ (resp. $\Delta\text{RSJ}_t < 0$), it means that the model overestimates (resp. underestimates) the importance of term t when considering its document ordering. In other words, the model retrieves "too much" (resp. "too few") this term. Please note that a high correlation between RSJ_S and RSJ_U **is not** indicative of the absolute performance of a model, as RSJ_U is neither a perfect model nor performance measure. However, we argue that it can still indicate partly the performance of the model w.r.t. lexical matching, especially for terms whose RSJ_U are high.

3 Experiments

We conducted experiments by analyzing models trained on MS MARCO [12], using public model parameters when available (indicated by $\star$). We evaluated models on the *in-domain* TREC Deep Learning 2019–2020 datasets [2,3] (97 queries in total), and two *out-of-domain* datasets from the BEIR [18] benchmark (`TREC-COVID` (bio-medical) and `FiQA-2018` (financial), with respectively 50 and 648 test queries). For all our experiments, we measure the system relevance by using top-$K = 100$. For the term-level analysis, we keep stopwords, and use standard tokenization and Porter stemming. We solely focus on first-stage retrievers (and not re-rankers), for which lexical matching might be more critical. We thus compare various state-of-the-art models (based on the BEIR benchmark), considering different types of approaches (sparse and dense). We include two lexical models, the standard `BM25` [16] and `doc2query-T5` ($\star$) [13]; `SPLADE` ($\star$) [5,6], an expansion-based sparse approach; `ColBERT` [10], an interaction-based architecture; two dense retrievers, `TAS-B` ($\star$) [8] and a standard `Bi-encoder` trained with contrastive loss and in-batch negatives.

3.1 Lexical Match in Neural IR

In Fig. 1, we plot the relationship between the user weight and ΔRSJ, for each term in the test queries appearing at least 10 times in the training queries (left,

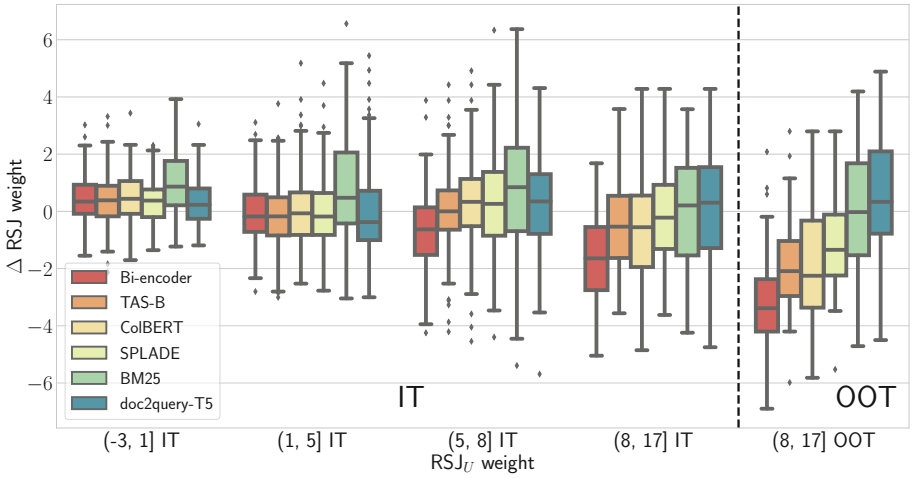

Fig. 1. ΔRSJ with respect to user RSJ$_U$ (x-axis, binned), splitting according to query terms seen during training (`IT`, left) or not (`OOT`, right). We consider that terms appearing in less than 10 training queries are `OOT`, leading to 499 and 42 terms in TREC queries, for `IT` and `OOT` respectively. Note that due to the fact that `OOT` terms are also generally rare in the collection, their RSJ$_U$ is always > 8, hence the single bin.

`IT` for In-Training). We first note that lexical-based models tend to overestimate the importance of query terms (ΔRSJ > 0). The second observation is that models are roughly similar in their estimations for low user RSJ$_U$ weights (below 5). Then, there is a clear distinction between the bi-encoder and other neural models (both dense and sparse): we can see that it retrieves less documents, on average, containing precisely the important query terms. Comparing dense and sparse/interaction models overall – by considering the average ΔRSJ over terms – we observe that, interestingly, dense models underestimate RSJ$_U$ ($\overline{\Delta\text{RSJ}} = -0.07$ for TAS-B and -0.26 for the bi-encoder) while sparse/interaction slightly overestimate it ($\overline{\Delta\text{RSJ}} = 0.03$ for ColBERT and SPLADE). Note again, as mentioned in Sect. 2, that the measure is not necessarily indicative of performance: for instance, TAS-B performs better than BM25 on TREC, suggesting that the model is better for semantic search. To illustrate the above, let us consider a query from the TREC DL set: "`does` (-1.12) `legionella` (14.85) `pneumophila` (13.12) `cause` (4.34) `pneumonia` (8.34)" (terms with associated RSJ$_U$). BM25 is able to correctly estimate importance for `legionella` (RSJ$_S$ = 15.08) contrary to neural approaches which tend to under-estimate it (RSJ$_S$ = 10.63, 13.42, 13.65 for the bi-encoder, SPLADE and ColBERT respectively).

We now shift our attention to the behavior of models for query words that are *(almost) not* in the training set. In Fig. 1, we show the distribution of ΔRSJ for terms appearing in less than 10 training queries (out of $> 500k$) (right, `OOT` for Out-Of-Training). Comparing with ΔRSJ for terms in the training set, we

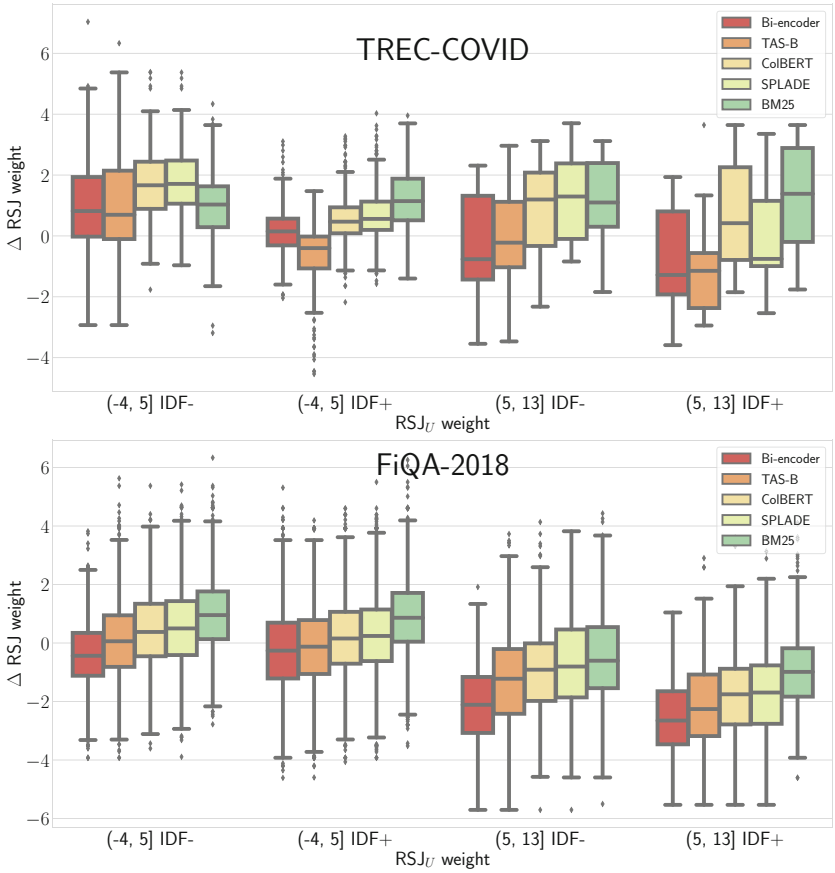

Fig. 2. ΔRSJ with respect to RSJ$_U$ (x-axis, binned) in the zero-shot setting. `IDF-` includes 108 and 933 terms, while `IDF+` includes 112 and 428 terms for respectively `TREC-COVID` and `FiQA-2018`. Note that bins are not similar compared to Fig. 1, as RSJ weights have different distributions on BEIR datasets.

can see that all neural models are affected somehow, showing that lexical match does not fully generalize to "new" terms. For the $(8, 17]$ bin, and for every model (except BM25), the difference in mean between `IT`/`OOT` is significant, based on a t-test with $p = 0.01$.

Finally, we also looked at the relationship between `IT`/`OOT` and model performance. More precisely, for terms in the $(8, 17]$ bin, we computed the mean ndcg@10 for queries containing at least one term either in `IT` or `OOT` (respectively 55 and 37 queries out of the 97, with 9 queries in both sets). We found that BM25 and doc2query-T5 performance increased by 0.1 and 0.02 respectively, while for all neural models the performance decreased (≈ 0 for TAS-B, –0.11 for SPLADE, -0.27 for the bi-encoder and -0.38 for ColBERT). The fact that BM25 performance increased is likely due to the fact that the mean IDF

increased (from 7.3 to 10.9), i.e. important terms are more discriminative in the OOT query set. With this in mind, the decrease of all neural models might suggest that a potential reason for the relative performance decrease (w.r.t. BM25) is due to a worse estimate of high RSJ_U.

3.2 Lexical Match and Zero-Shot Transfer Learning

We now analyze whether term importance can generalize to the zero-shot setting[1]. We distinguish two categories of words, namely those that occurred 5 times more in the target collection than in MS MARCO (IDF+), or those for which term statistics were more preserved (IDF-), allowing us to split query terms in sets of roughly equal size. Since term importance is related to the collection frequency (albeit loosely), we can compare ΔRSJ in those two settings. Figure 2 shows the ΔRSJ with respect to RSJ_U for the TREC-COVID and FiQA-2018 collections from the BEIR benchmark [18].

We can first observe that neural models underestimate RSJ_U for terms that are more frequent in the target collection than in the training one (IDF+). It might indicate that models have learned a dataset-specific term importance – confirming the results obtained in the previous section on out-of-training terms. When comparing dense and sparse/interaction models overall – by considering the average ΔRSJ over terms – we observe than dense models underestimate even more RSJ_U than on in-domain ($\overline{\Delta\text{RSJ}} = -0.17$ for TAS-B and -0.38 for the bi-encoder) while sparse/interaction seem to overestimate ($\overline{\Delta\text{RSJ}} = 0.18$ for ColBERT and 0.30 for SPLADE), but however to a lesser extent than BM25 ($\overline{\Delta\text{RSJ}} = 0.83$). Finally, we observed that when transferring, all the models have a higher ΔRSJ variance compared to their trained version on MS MARCO: in all cases, the standard deviation (when normalized by BM25 one) is around 0.8 for MS MARCO, but around 1.1 for TREC-COVID and FiQA-2018. This further strengthens our point on the issue of generalizing lexical matching to out-of-domain collections.

4 Conclusion

In this work, we analyzed how different neural IR models predict the importance of lexical matching for query terms. We proposed to use the Robertson-Sparck Jones (RSJ) weight as an appropriate measure to compare term importance w.r.t. the user and system relevance. We introduce a black box approach that enables a systematic comparison of different models w.r.t. term matching. We have also investigated the behavior of lexical matching in the zero-shot setting. Overall, we have shown that lexical matching properties are heavily influenced by the presence of the term in the training collection. The rarer the term, the harder it is to find documents containing that term for most neural models. Furthermore, this phenomenon is amplified if term statistics change across collections.

[1] We excluded doc2query-T5 from the analysis, due to the high computation cost for obtaining the expanded collections.

References

1. Camara, A., Hauff, C.: Diagnosing BERT with Retrieval Heuristics. In: ECIR. p. 14 (2020), zSCC: NoCitationData[s0]
2. Craswell, N., Mitra, B., Yilmaz, E., Campos, D.: Overview of the trec 2020 deep learning track (2021)
3. Craswell, N., Mitra, B., Yilmaz, E., Campos, D., Voorhees, E.M.: Overview of the trec 2019 deep learning track (2020)
4. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Burstein, J., Doran, C., Solorio, T. (eds.) Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2–7, 2019, Volume 1 (Long and Short Papers). pp. 4171–4186. Association for Computational Linguistics (2019). 10.18653/v1/n19-1423, https://doi.org/10.18653/v1/n19-1423
5. Formal, T., Lassance, C., Piwowarski, B., Clinchant, S.: SPLADE v2: sparse lexical and expansion model for information retrieval. arXiv:2109.10086 [cs], September 2021. http://arxiv.org/abs/2109.10086, arXiv: 2109.10086
6. Formal, T., Piwowarski, B., Clinchant, S.: Splade: sparse lexical and expansion model for first stage ranking. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2021, pp. 2288–2292. Association for Computing Machinery, New York (2021). https://doi.org/10.1145/3404835.3463098, https://doi.org/10.1145/3404835.3463098
7. Formal, T., Piwowarski, B., Clinchant, S.: A white box analysis of ColBERT. In: Hiemstra, D., Moens, M.-F., Mothe, J., Perego, R., Potthast, M., Sebastiani, F. (eds.) ECIR 2021. LNCS, vol. 12657, pp. 257–263. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-72240-1_23
8. Hofstätter, S., Lin, S.C., Yang, J.H., Lin, J., Hanbury, A.: Efficiently Teaching an Effective Dense Retriever with Balanced Topic Aware Sampling. In: SIGIR, July 2021
9. Jiang, Z., Tang, R., Xin, J., Lin, J.: How does BERT rerank passages? an attribution analysis with information bottlenecks. In: EMNLP Workshop, Black Box NLP, p. 14 (2021)
10. Khattab, O., Zaharia, M.: ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT. arXiv:2004.12832 [cs], April 2020. http://arxiv.org/abs/2004.12832, arXiv: 2004.12832
11. MacAvaney, S., Feldman, S., Goharian, N., Downey, D., Cohan, A.: ABNIRML: analyzing the behavior of neural IR models. arXiv:2011.00696 [cs], Nov 2020. http://arxiv.org/abs/2011.00696, zSCC: 0000000 arXiv: 2011.00696
12. Nguyen, T., Rosenberg, M., Song, X., Gao, J., Tiwary, S., Majumder, R., Deng, L.: Ms marco: a human generated machine reading comprehension dataset. CoRR abs/1611.09268 (2016). http://dblp.uni-trier.de/db/journals/corr/corr1611.html#NguyenRSGTMD16
13. Nogueira, R., Lin, J.: From doc2query to docTTTTTquery, p. 3, zSCC: 0000004
14. Robertson, S.E., Jones, K.S.: Relevance weighting of search terms. J. Am. Soc. Inf. Sci. **27**(3), 129–146 (1976). 10/dvgb84, https://onlinelibrary.wiley.com/doi/abs/10.1002/asi.4630270302,_eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/asi.4630270302
15. Robertson, S.E.: The probability ranking principle in IR. J. Documentation **33**(4), 294–304 (1977). 10/ckqfpm, https://doi.org/10.1108/eb026647, publisher: MCB UP Ltd

16. Robertson, S.E., Zaragoza, H.: The Probabilistic Relevance Framework: BM25 and Beyond. Foundations and Trends in Information Retrieval (2009)
17. Sciavolino, C., Zhong, Z., Lee, J., Chen, D.: Simple entity-centric questions challenge dense retrievers. In: Empirical Methods in Natural Language Processing (EMNLP) (2021)
18. Thakur, N., Reimers, N., Rücklé, A., Srivastava, A., Gurevych, I.: BEIR: a heterogenous benchmark for zero-shot evaluation of information retrieval models. arXiv:2104.08663 [cs], September 2021. http://arxiv.org/abs/2104.08663
19. Yates, A., Nogueira, R., Lin, J.: Pretrained transformers for text ranking: Bert and beyond. In: Proceedings of the 14th ACM International Conference on Web Search and Data Mining, WSDM 2021, pp. 1154–1156. Association for Computing Machinery, New York (2021). https://doi.org/10.1145/3437963.3441667, https://doi.org/10.1145/3437963.3441667
20. Yu, C.T., Salton, G.: Precision weighting - an effective automatic indexing method. J. ACM **23**(1), 76–88 (1976). 10/d3fgsz, https://doi.org/10.1145/321921.321930