Beneath the [MASK]: An Analysis of Structural Query Tokens in ColBERT

Ben Giacalone[1 [0009−0006−8525−959X]], Greg Paiement[1 [0009−0007−0819−5315]], Quinn Tucker[1 [0009−0007−8885−245X]], and Richard Zanibbi[1 [0000−0001−5921−9750]]

Rochester Institute of Technology, Rochester NY 14623, USA

Abstract. ColBERT is a highly effective and interpretable retrieval model based on token embeddings. For scoring, the model adds cosine similarities between the most similar pairs of query and document token embeddings. Previous work on interpreting how tokens affect scoring pay little attention to non-text tokens used in ColBERT such as [MASK]. Using MS MARCO and the TREC 2019-2020 deep passage retrieval task, we show that [MASK] embeddings may be replaced by other query and structural token embeddings to obtain similar effectiveness, and that [Q] and [MASK] are sensitive to token order, while [CLS] and [SEP] are not.

Keywords: ColBERT · interpretability · embeddings · relevance scoring

1 Introduction

The ColBERT [4] retrieval model uses BERT [2] to produce token embeddings for document and query passages. Typically, candidate documents are retrieved using dense retrieval on embedded tokens [15, 17], and then re-scored using the sum of maximum cosine similarities between each query token embedding and its most similar document token embedding via the *MaxSim* operator. Rescoring improves retrieval effectiveness, and is more interpretable than dense retrieval models that use single vectors (e.g. the BERT [CLS] token), because query tokens contribute individually to document rank scores [3], and token embeddings can be analyzed directly.

Interestingly, not all tokens used in ColBERT's scoring are text tokens. Some are *structural tokens* that mark locations and segments in a token sequence. ColBERT employs a modified BERT model to create contextualized embeddings for *every* document and query token, including structural BERT tokens. Structural tokens include [CLS], which appears at the input start, followed by [Q] or [D] to signify whether a passage is from a query or a document. Text tokens from the query are next, followed by [SEP] after the final text token. Query token sequences shorter than the input size are padded with [MASK] tokens,[1] and document token sequences are padded with [PAD] tokens. Below are example query and document passage tokenizations with input sizes of 32 and 180 tokens, respectively. Subscripts are used to indicate token position in the input.

[1] [MASK] was originally devised for BERT to represent a "hidden" input token in its masked token prediction training task.

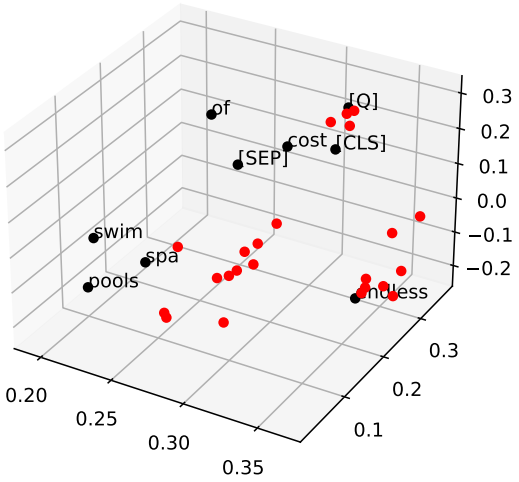


Fig. 1. PCA embeddings for the MS MARCO query "cost of endless pools swim spa". `[MASK]` tokens (red points) cluster around query words and structural tokens (black).

Q: `[CLS]`$_1$ `[Q]`$_2$ `cost`$_3$ `of`$_4$ `endless`$_5$ `pools`$_5$ `swim`$_6$ `spa`$_7$ `[SEP]`$_8$ `[MASK]`$_9$ `[MASK]`$_{10}$ `[MASK]`$_{11}$ ... `[MASK]`$_{30}$ `[MASK]`$_{31}$ `[MASK]`$_{32}$

D: `[CLS]`$_1$ `[D]`$_2$ `prices`$_3$ ... `swim`$_{12}$ `spa`$_{13}$ `.`$_{14}$ `[SEP]`$_{15}$ `[PAD]`$_{16}$ `[PAD]`$_{17}$ `[PAD]`$_{18}$ ... `[PAD]`$_{178}$ `[PAD]`$_{179}$ `[PAD]`$_{180}$

Figure 1 shows the token embeddings for the query above.[2] As seen in Figure 1, `[MASK]` tokens tend to cluster around other query tokens, giving them additional weight [4, 13]. The original ColBERT paper suggests `[MASK]` tokens provide a form of query augmentation through term re-weighting and query expansion. Wang et al. [12, 13] study a version of ColBERT that performs query expansion using pseudo-relevance feedback, and find that `[MASK]` tokens generally do *not* expand the query by matching terms outside the query, and instead need to add them explicitly. In this way, `[MASK]` tokens primarily weight query tokens by matching query text tokens in documents. Wang et al. [14] also find that for many ColBERT based models, using only query structural tokens for retrieval (`[CLS]`, `[SEP]`, `[Q]`, `[MASK]`) is nearly as effective as using all token embeddings for retrieval, and outperforms using only low IDF query token embeddings.

Previous studies of ColBERT's retrieval behavior have focused on text tokens. In considering why ColBERT's ranking mechanism outperforms standard lexical models such as BM25, Formal et al. [3] focus on query text tokens, and find that tokens with high Inverse Document Frequency (IDF) produce more *exact* matches in ColBERT query/document token alignments (e.g. `(Q:pool,D:pool)`) while low IDF terms produce more *inexact* matches (e.g. `(Q:is,D:and)`). Low IDF token embeddings also tend to shift position more, and removing high IDF tokens perturbs ranking more than removing low IDF tokens. MacAvaney et al.

[2] Interactive version: `https://cs.rit.edu/~bsg8294/colbert/query_viz.html`

[5] also found a sensitivity for text tokens in ColBERT, with misspellings harming retrieval more than in lexical models. Curiously, they also find ColBERT increasing document scores when *non-relevant* tokens are appended to a document token sequence, while appending *relevant* terms decreases rank scores even after controlling for document length. Perhaps appending relevant terms produces an 'unnatural' token sequence for the embedding language model which interferes with token embedding/contextualization and *MaxSim* scoring.

In this paper, we extend inquiries into how tokens impact retrieval in ColBERT by shifting focus to structural tokens, and [MASK] in particular. In the next Section we present experiments to address the following research questions:

RQ1. Do [MASK] tokens perform more than just term weighting?
RQ2. How sensitive are [CLS], [SEP], [Q], and [MASK] to query token order?

2 Methodology and Experimental Designs

For our experiments, we use the ColBERT v1 model integrated within PyTerrier [6]. The state-of-the-art ColBERT v2 [9] model adds index compression and training with hard negatives and distillation to improve rank quality. Index compression and embedding modifications may alter retrieval candidates and token cosine similarities, and we plan to check this in future work. However, we wish to first study the simpler, original ColBERT model. We also believe insights into the workings of ColBERT v1 and models inspired by it (e.g. the text/image model FILIP [16]) will be beneficial for the research community.

Implementation, Datasets, and Metrics. We use a ColBERT v1 checkpoint from the University of Glasgow trained on passage ranking triples for 44k batches,[3] and run experiments on a server with 4 Intel Xeon E5-2667v4 CPUs, 4 NVIDIA RTX2080-Ti GPUs, and 512 GB RAM. For our experiments, we use two datasets:

1. MS MARCO [7]'s passage retrieval dev set (8.8 million documents, 1 million queries, binary relevance judgements). Each query has at most 1 matching document. We use this dataset for query statistics (e.g. cosine distances between query embeddings).
2. A dataset combining test queries from the TREC 2019 [11] and 2020 [1] deep passage retrieval task (99 queries, graded relevance judgements). Collection documents are the same as MS MARCO. We use this dataset for experiments focused upon retrieval quality.[4]

For the TREC 2019-2020 collection, the relevance scale is between 0 and 3 with a score of 2 considered relevant for metrics using binary relevance (e.g. MAP). We examine relevance scores thresholded at 1, 2, and 3 to see the effect of binarizing at different relevance grades. We use MRR@10 to characterize effectiveness for

[3] http://www.dcs.gla.ac.uk/ craigm/ecir2021-tutorial/colbert_model_checkpoint.zip
[4] Running the TREC test queries takes roughly 15 minutes to complete using a multithreaded Rust program: https://github.com/Boxxfish/IR2023-Project

top results, MAP to characterize effectiveness for complete rankings, and to complement MAP we use nDCG@k measures ($k \in \{10, 1000\}$) to utilize graded relevance labels from the TREC data.

RQ1: Do `[MASK]` tokens perform more than just term weighting? Figure 1 illustrates how `[MASK]` token embeddings cluster around query terms, which was consistent for MS MARCO queries we examined. As mentioned previously, `[MASK]` tokens have been identified as having two roles in scoring: (1) query term weighting through matching document terms to `[MASK]` tokens with embeddings similar to non-`[MASK]` query tokens, and (2) query expansion through `[MASK]` embeddings shifting toward potentially relevant tokens outside of the query.

In this experiment, we test whether the clustering of `[MASK]` tokens around query tokens indicates that term weighting is the *only* role `[MASK]` tokens actually play in ColBERT scoring. To do this, we replace structural token embeddings in a query with text token embeddings from the same query. This forces ColBERT to perform term weighting: replacing structural token embeddings by their nearest text embeddings cannot introduce new terms or perform "soft weighting" by increasing the weight of multiple query tokens. We use the TREC 2019-2020 collection, and compare four token remapping conditions: (1) no remapping, (2) remapping `[MASK]` tokens to text tokens, (3) remapping all structural token embeddings (`[CLS]`, `[SEP]`, `[Q]`, and `[MASK]`) to text tokens, and finally (4) remapping each `[MASK]` to its most similar embedded text token *or* non-`[MASK]` structural token (i.e. `[CLS]`, `[SEP]`, or `[Q]`). We hypothesize that replacing `[MASK]` embeddings by non-`[MASK]` embeddings in queries will reduce effectiveness by preventing matches with terms that do not appear in the query (i.e. by preventing query expansion).

RQ2: How sensitive are `[CLS]`, `[SEP]`, `[Q]`, and `[MASK]` to query token order? As shown in the example above, ColBERT begins every query token sequence passed to BERT with the structural tokens `[CLS]` and `[Q]`, followed by the text tokens and the structural token `[SEP]` marking the end of the text tokens, and finally zero or more `[MASK]` tokens to fill out the fixed-size input (e.g. 32 tokens). `[CLS]` is included in BERT's training objective function, and aggregates context across entire query and document passages resulting in a "summary" representation. We thus expect queries with similar wording and intent to produce similar `[CLS]` embeddings, even when the query word order changes. We expect the same pattern to hold for `[SEP]` which terminates every query and document passage. In contrast, we expect `[MASK]` embeddings to vary more than `[CLS]` and `[SEP]` tokens when words are re-ordered, because of their observed clustering around query terms and resulting weighting of individual terms in scoring. We expect `[Q]` embeddings to also vary more than `[CLS]` and `[SEP]`, because `[Q]` is absent in the original BERT training objective.

To study how query word order influences contextualization for query structural tokens, we reorder query text terms prior to contextualization similar to Rau et. al [8]. Randomly shuffling query tokens may alter the meaning of a query, so we limit the permutations considered. Specifically, we transform queries of the

Table 1. Replacing structural token embeddings with other query token embeddings (TREC 2019-2020, RQ1). Maximum values are in bold; significant differences from "None" are shown with a dagger ($p < 0.05$, Bonferroni-corrected t-tests).

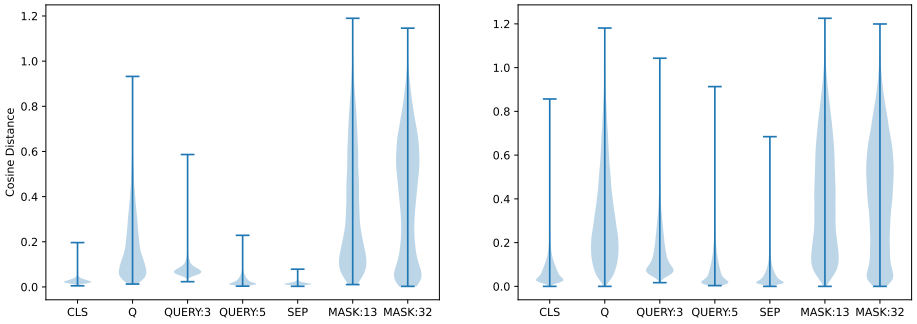
Metric	Structural Token Remapping			
	None	**All [X] → Text**	**[MASK] → Text**	**[MASK] → Str. & Text**
Binary Relevance				
MAP(rel≥1)	0.447	0.454	†**0.462**	†**0.462**
MRR(rel≥1)@10	**0.930**	0.924	0.929	0.923
MAP(rel≥2)	0.450	0.444	0.454	**0.457**
MRR(rel≥2)@10	**0.851**	0.820	0.835	0.837
MAP(rel≥3)	0.366	0.362	**0.373**	0.372
MRR(rel≥3)@10	0.557	0.560	**0.563**	**0.563**
Graded Relevance				
nDCG@10	0.689	0.685	0.691	**0.694**
nDCG@1000	0.680	0.673	0.683	**0.684**

form "what is ..." into "... is what", moving the first two text tokens to the end of the query in the opposite order. To further avoid accidental semantic shifting, we only examine queries that are 3-8 tokens long. 12,513 queries in the MS MARCO dev set fit this criteria. As a baseline, we also apply the same reordering pattern to *all* queries 3-8 tokens long, without requiring the first two tokens to be "what is". 68,318 queries in the dev set fit this criteria. For the reasons given above, we hypothesize that [Q] and [MASK] embeddings will change more than [SEP] and [CLS] under this reordering. We use the cosine distance to quantify the shift in token embeddings after reordering the query text tokens.

3 Results

RQ1: Do [MASK] tokens perform more than just term weighting? In Table 1 we see replacing embeddings for *all* structural tokens with their closest text token embedding produces non-significant reductions in metrics other than MRR@10 (rel≥3). The two conditions mapping only [MASK] have very similar metrics, but surprisingly produce slight increases in MAP and nDCG@10/@1000 over both the "None" and "All" conditions. For MAP(rel≥1), the increase is significant (1.5%). MRR@10(rel≥3) is also slightly higher than standard ColBERT (but not significantly so). These small increases are likely from incorporating additional context through the [CLS], [Q], and [SEP] tokens (especially [CLS]). This contradicts our hypothesis that remapping [MASK] embeddings would harm performance, and is also interesting because [MASK] tokens comprise most of the input for short queries. In other words, its appears that strong retrieval performance with ColBERT is possible even when using only a few text token embeddings, provided that term weighting is taken into account.

RQ2: How sensitive are [CLS], [SEP], [Q], and [MASK] to query token order? In Figure 2(a), [QUERY:3] is the third token and first text token (always "what")



(a) "`what is ...`" → "`... is what`" (b) All queries: "`1 2 ...`" → "`... 2 1`"

Fig. 2. Distribution of cosine distance $(1 - \cos(\boldsymbol{e}, \boldsymbol{e'}))$ for token embeddings before and after query token reordering (MS MARCO, RQ2). For brevity not all tokens are shown, but the general trend of higher variance holds for all `[MASK]` tokens. **Left:** Cosine distances for queries starting with "what is". **Right:** Cosine distances without requiring queries to start with "what is". `[QUERY:3]` and `[QUERY:5]` are the first and third text tokens, respectively; `[MASK:13]` represents the `[MASK]` token at position 13, and `[MASK:32]` represents the final `[MASK]` input token at position 32.

while `[QUERY:5]` is the fifth token containing the text token after "what is." We see distinct differences in how cosine distances are distributed for `[CLS]`, `[SEP]`, `[QUERY:3]`, and `[QUERY:5]` versus `[Q]`, `[MASK:13]`, and `[MASK:32]`. The first group shows barely any shift, while the latter group shows large shifts, with higher variation. Figure 2(b) shows results for queries reordered similarly, but without requiring them to start with "what is". For example, *"airplane flights to florida"* produces the somewhat unnatural query *"to florida flights airplane"*. All tokens show larger representational shifts in this condition; however, we again find that `[CLS]`, `[SEP]`, and the `[QUERY:3/5]` text token embeddings vary far less than the `[Q]` and `[MASK]` embeddings.

Despite our efforts to avoid it, some "what is" queries have their meaning altered by our reordering. For example, *"what is some examples homogeneous"* becomes *"some examples homogeneous is what"*, which may change the query from a request for examples to asking for a definition. When we filtered out queries containing the word "example", the variance of `[QUERY:3]` dropped from $8.53 \cdot 10^{-4}$ to $7.73 \cdot 10^{-4}$, while the variance of `[Q]` had less of a proportional drop ($2.07 \cdot 10^{-2}$ to $2.06 \cdot 10^{-2}$), indicating some of the variance of non-`[Q]` or `[MASK]` embeddings may be due to these edge cases.

4 Discussion and Conclusion

To our surprise, replacing `[MASK]` token embeddings in queries with either their most similar text token embedding in the same query, or the most similar text or non-`[MASK]` structural token embedding from the query yielded similar effec-

tiveness to standard ColBERT for the TREC 2019-2020 dataset. There is even a small significant increase in MAP when weakly-relevant documents are considered relevant (i.e. MAP(rel≥1)). The differences between mapping [MASK] to only text tokens or to both text and non-MASK structural tokens was statistically insigificant for all metrics observed. So if [MASK] tokens have effects other than term weighting in scoring, their role appears to be minor (RQ1).

This suggests a possible optimization. We can multiply each non-[MASK] query token embedding's score contribution by the number of [MASK] token embeddings most similar to it. Regarding interpretability, using [MASK] only to weights tokens in this manner simplifies the ColBERT scoring model both conceptually and computationally. For short queries, most of the input to ColBERT is [MASK] tokens, and so the number of query tokens to match against document tokens with *MaxSim* may be a fraction of the full token input size. A related approach is described by Tonellotto et al. [10], where query embeddings are dropped after contextualization based on frequency statistics. However, rather than pruning a set number of tokens based on collection frequency, we would use all token embeddings to retrieve candidates, and then weight non-[MASK] query tokens using fewer nearest neighbor lookups during scoring.

However, the question of [MASK] tokens' role in retrieving candidates still remains, as this paper has focused on the final scoring step; all query tokens were used to retrieve candidates in our experiments. How might limiting or removing the use of [MASK] tokens in the first-stage dense retrieval impact performance? We wonder about the small statistically insignificant improvements seen in MAP and MRR for highly relevant documents (rel≥3), as well as nDCG@10, nDCG@1000, and MAP. Are these stable and/or significant in other collections? To better understand [MASK], one possible experiment is appending different numbers of [MASK] tokens to each query. This may reveal whether having fewer [MASK] tokens causes them to move closer to non-[MASK] embeddings, and whether having more [MASK] tokens might improve term weighting.

Regarding the effect of token ordering on contextualized embeddings (RQ2), our findings are consistent with our original hypothesis: [CLS] and [SEP] embeddings do not vary greatly for similar queries with a different token ordering, while [Q] and [MASK] do. The shift in [Q] is the most interesting result here; the model may be treating [Q] similar to another [MASK] token. Some early analysis suggests that a query [CLS] tends to match a document [CLS], a query [SEP] tends to match ending punctuation, and [MASK] tends to match tokens other than [CLS] or [SEP] (see our interactive visualization for ColBERT scoring[2]). We have not observed [Q] matching to any specific document tokens.

In the future we would also like to validate our results using ColBERT v2. We believe that our results *should* hold for the newer model – if [MASK]s continue to cluster around query word embeddings, we expect [MASK]s will continue to act as term weights, and the training process in ColBERT v2 should not alter how [Q] is processed. We would also like to extend our evaluation to additional datasets, since we have only focused on MS MARCO and the MS MARCO-derived TREC 2019-2020 datasets in our experiments.

References

1. Craswell, N., Mitra, B., Yilmaz, E., Campos, D.: Overview of the TREC 2020 deep learning track. In: Voorhees, E.M., Ellis, A. (eds.) Proc. Text REtrieval Conference (TREC). NIST Special Publication, vol. 1266 (2020)
2. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proc. North American Chapter of the Association for Computational Linguistics (NAACL). pp. 4171–4186 (2019). https://doi.org/10.18653/v1/N19-1423, https://aclanthology.org/N19-1423
3. 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.) Proc. European Conference on Information Retrieval (ECIR). LNCS, vol. 12657, pp. 257–263. Springer (2021)
4. Khattab, O., Zaharia, M.: ColBERT: Efficient and effective passage search via contextualized late interaction over BERT. In: Huang, J.X., Chang, Y., Cheng, X., Kamps, J., Murdock, V., Wen, J., Liu, Y. (eds.) Proc. SIGIR. pp. 39–48 (2020). https://doi.org/10.1145/3397271.3401075
5. MacAvaney, S., Feldman, S., Goharian, N., Downey, D., Cohan, A.: ABNIRML: Analyzing the Behavior of Neural IR Models. Transactions of the Association for Computational Linguistics **10**, 224–239 (03 2022). https://doi.org/10.1162/tacl_a_00457, https://doi.org/10.1162/tacl_a_00457
6. Macdonald, C., Tonellotto, N., MacAvaney, S., Ounis, I.: PyTerrier: Declarative experimentation in Python from BM25 to dense retrieval. In: Proc. Intl. Conf. Information & Knowledge Management (CIKM). pp. 4526—4533 (2021). https://doi.org/10.1145/3459637.3482013
7. Nguyen, T., Rosenberg, M., Song, X., Gao, J., Tiwary, S., Majumder, R., Deng, L.: MS MARCO: A human generated machine reading comprehension dataset. In: Besold, T.R., Bordes, A., d'Avila Garcez, A.S., Wayne, G. (eds.) Proceedings of the Workshop on Cognitive Computation: Integrating neural and symbolic approaches 2016 co-located with the 30th Annual Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain, December 9, 2016. CEUR Workshop Proceedings, vol. 1773. CEUR-WS.org (2016), https://ceur-ws.org/Vol-1773/CoCoNIPS_2016_paper9.pdf
8. Rau, D., Kamps, J.: The role of complex NLP in transformers for text ranking. In: Proc. ICTIR. pp. 153–160 (2022). https://doi.org/10.1145/3539813.3545144, http://dx.doi.org/10.1145/3539813.3545144
9. Santhanam, K., Khattab, O., Saad-Falcon, J., Potts, C., Zaharia, M.: ColBERTv2: Effective and efficient retrieval via lightweight late interaction. In: Carpuat, M., de Marneffe, M.C., Meza Ruiz, I.V. (eds.) Proc. North American Chapter of the Association for Computational Linguistics (NAACL). pp. 3715–3734 (2022). https://doi.org/10.18653/v1/2022.naacl-main.272
10. Tonellotto, N., Macdonald, C.: Query embedding pruning for dense retrieval. In: Proc. Intl. Conf. Information & Knowledge Management (CIKM). pp. 3453–3457 (2021). https://doi.org/10.1145/3459637.3482162, https://doi.org/10.1145/3459637.3482162
11. Voorhees, E.M., Ellis, A. (eds.): Proceedings of the Twenty-Eighth Text REtrieval Conference, TREC 2019, Gaithersburg, Maryland, USA, November 13-15, 2019, NIST Special Publication, vol. 1250. National Institute of Standards and Technology (NIST) (2019), https://trec.nist.gov/pubs/trec28/trec2019.html

12. Wang, X., Macdonald, C., Ounis, I.: Improving zero-shot retrieval using dense external expansion. Information Processing Management **59**(5), 103026 (2022). https://doi.org/https://doi.org/10.1016/j.ipm.2022.103026, https://www.sciencedirect.com/science/article/pii/S0306457322001364
13. Wang, X., MacDonald, C., Tonellotto, N., Ounis, I.: ColBERT-PRF: Semantic pseudo-relevance feedback for dense passage and document retrieval. ACM Trans. Web **17**(1) (2023). https://doi.org/10.1145/3572405, https://doi.org/10.1145/3572405
14. Wang, X., Macdonald, C., Tonellotto, N., Ounis, I.: Reproducibility, replicability, and insights into dense multi-representation retrieval models: From ColBERT to Col*. In: Proc. SIGIR. pp. 2552–2561. ACM (2023). https://doi.org/10.1145/3539618.3591916
15. Xiong, L., Xiong, C., Li, Y., Tang, K., Liu, J., Bennett, P.N., Ahmed, J., Overwijk, A.: Approximate nearest neighbor negative contrastive learning for dense text retrieval. In: Proc. Int. Conf. Learning Representations (ICLR). OpenReview.net (2021), https://openreview.net/forum?id=zeFrfgyZln
16. Yao, L., Huang, R., Hou, L., Lu, G., Niu, M., Xu, H., Liang, X., Li, Z., Jiang, X., Xu, C.: FILIP: fine-grained interactive language-image pre-training. In: The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022. OpenReview.net (2022), https://openreview.net/forum?id=cpDhcsEDC2
17. Zhan, J., Mao, J., Liu, Y., Zhang, M., Ma, S.: RepBERT: Contextualized text embeddings for first-stage retrieval. CoRR **abs/2006.15498** (2020), https://arxiv.org/abs/2006.15498