NAVER LABS EUROPE (SPLADE) @ TREC DEEP LEARNING 2021

Carlos Lassance
Naver Labs Europe
France
carlos.lassance@naverlabs.com

Thibault Formal
Naver Labs Europe and Sorbonne Université, ISIR
France
thibault.formal@naverlabs.com

Benjamin Piwowarski
Sorbonne Université, ISIR, CNRS
France
benjamin@piwowarski.fr

Arnaud Sors, Stéphane Clinchant
Naver Labs Europe
France
{arnaud.sors, stephane.clinchant}@naverlabs.com

ABSTRACT

This paper describes our participation to the 2021 TREC Deep Learning challenge. We submitted runs to both passage and document *full ranking* tasks, with a focus on the passage task, where the goal is to retrieve and rank a set of 100 passages directly from the new MS MARCO v2 collection, containing around 138M entries. We rely on SPLADE for first-stage retrieval. For the second stage, we use an ensemble of BERT re-rankers, trained using hard negatives selected by SPLADE. Three runs were submitted, coming from a diverse set of experiments: i) a fast retriever *without* re-ranking nor query encoding (SPLADE-doc model), ii) a SPLADE model fully trained on MS MARCO v1, and re-ranked by an ensemble of models, and iii) an ensemble of SPLADE models trained on both new and old MS MARCO, re-ranked by an ensemble of models also trained on both datasets. Much to our surprise, out of the 3 runs, the one trained only on MS MARCO v1 obtained the best results on the TREC competition and is very competitive when compared to the median and best results. More surprising to us is that the results on the dev set of MS MARCO v2 did not correlate with TREC results, *contrary to previous years*.

1 Introduction

In this paper, we detail our TREC 2021 Deep Learning track submission, based on the SPLADE model [Formal et al., 2021b]. We introduce several improvements for SPLADE, that are further detailed in the updated version of the model [Formal et al., 2021a]. We submitted three runs to both passage and document full-ranking tasks, with a focus on the first one. For the document task, we simply score a document by taking the maximum score over its passages.

MS MARCO v2 The TREC organizers introduced a new dataset for the Deep Learning track (TREC DL), that we refer to as MS MARCO v2 (and thus MS MARCO v1 for the previous one). The main differences between the two datasets are: i) the size of their respective collection, with v2

containing around fifteen times more passages than v1, and ii) how documents are cut into passages: we now have a mapping from all passages to documents, allowing our participation to the document task, where we do everything in the passage domain and then map *ids* to the document ones.

Note that the increased collection size introduces some issues. First, indexing and searching in the collection is more expensive than before. Second, the relevance assignment (*qrels*) changed a lot due to the way the passages are extracted from documents, causing issues for training and evaluation (Section 3.3). Finally, the amount of documents from the collection that are not seen during training increased a lot, as the amount of queries (and positives/negatives per query) did not change.

2 Methodology

In the following, we introduce the models we consider for both candidate generation as well as re-ranking. We also describe our training procedure, and detail the submitted runs.

2.1 First Stage - Candidate mining with SPLADE

SPLADE [Formal et al., 2021b] is a transformer-based retrieval model, that relies on the Masked Language Modeling (MLM) head and max pooling over the tokens to represent documents and queries. Thus, document/query representations have the same number of dimensions as the amount of tokens in the transformer vocabulary (in this case BERT [Devlin et al., 2018], with $|V| \approx 30000$). The model is trained by jointly optimizing ranking and regularization losses; consequently, retrieval can be done in a sparse fashion, as only a few dimensions are activated by SPLADE for a given document (or query).

A variant of such model consists in having only a document encoder: the ranking score is then a simple sum over query (sub-)words. This model is referred to as SPLADE-doc in [Formal et al., 2021b]. In this case, the only cost associated with retrieval is the index search, as there is no inference on query side.

The training of SPLADE is done via optimization of a contrastive loss (InfoNCE) using hard-negatives (from BM25) and in-batch negatives, constrained by a regularization that aims at reducing the amount of expected floating-point operations during retrieval. For our models trained on MS MARCO v1, we build upon recent distillation work [Hofstätter et al., 2020], and train SPLADE with the MarginMSE loss[1] (thus replacing the InfoNCE loss). Unfortunately we could not generate the data (i.e. the teacher scores) in time for MS MARCO v2, and thus all SPLADE models trained on v2 do not rely on distillation. Also, note that while it has been shown that hard-negative mining [Xiong et al., 2021] and intelligent query sampling [Hofstätter et al., 2021] may improve the results of retrieval, we were not able to integrate it in time for TREC.

2.2 Second Stage - Re-ranking with cross-attention models

In recent years it has been shown that re-ranking based on cross-attention models [Nogueira and Cho, 2019] is paramount for the TREC competitions [Craswell et al., 2020, Craswell et al., 2021]. In our case, we build upon a recent work [2] [Gao et al., 2021] in order to train our re-rankers.

The procedure can be summarized shortly: for each annotated query of the training set (either MS MARCO v1 or v2), we draw negatives from the first-stage SPLADE model used for the run, either from the top-100 or top-1000. This training procedure differs from most prior works where re-rankers are systematically trained with BM25 negatives, even though being applied on top of other models like dense bi-encoders, introducing a shift between train and test distributions.

2.3 Ensembling

We also have applied ensembling in order to improve our results. We proceeded in a very standard way: the score for a document is the mean score over all the individual model scores.

[1] using the provided cross-encoder teacher scores from `https://github.com/sebastian-hofstaetter/neural-ranking-kd`

[2] code available at `https://github.com/luyug/Reranker`

2.4 Runs submitted to TREC

For our TREC submission, we considered three approaches:

- *Quick*: Based only on the SPLADE-doc model, trained with InfoNCE on the MS MARCO v1 triplets;
- *V1*:
  - First stage: Based on the distilled SPLADE model, trained with MarginMSE on the MS MARCO v1 model. The top-1000 documents are kept for the next stage.
  - Second stage: An ensemble of *seven* re-rankers, where: i) 1 is taken "off-the-shelf"[3], based on MINILM [Wang et al., 2020], ii) 2 models are trained on the top-100 results from SPLADE, on the training queries of MS MARCO v1. These models use ELECTRA-large [Clark et al., 2020] and ROBERTA-large [Liu et al., 2019] as pre-trained checkpoints, and iii) 4 models are trained on the top-1000 results from SPLADE, on the training queries of MS MARCO v1. Three of these models use the ROBERTA-large [Liu et al., 2019] as their backbone with different hyperparameters during fine-tuning; the last one relies on ELECTRA-large [Clark et al., 2020].
- *V1 + V2*:
  - First stage: An ensemble of 5 SPLADE models. These models are i) the same as the *Quick* model, ii) the first-stage of *V1*, iii) a sparser model using the same training as *V1*, iv) a SPLADE model trained with InfoNCE on the MS MARCO v1 dataset, and v) a SPLADE model trained with InfoNCE on the MS MARCO v2 dataset. We get the top-1000 documents of each model and then perform ensembling via the mean score over all models, in order to choose which documents to consider for re-ranking;
  - Second stage: An ensemble of ten re-rankers, where: i) 7 are the same as the ones used in *V1*, and ii) 3 models are trained on the top-1000 results of BM25 on the training queries of MS MARCO v2. The pre-trained checkpoints used are ROBERTA-large [Liu et al., 2019], MINILM [Wang et al., 2020] and ELECTRA-large [Clark et al., 2020].

3 Analysis on MS MARCO v1 and v2

In the following we discuss our experimental process and we analyze the performance of different models on the dev sets of MS MARCO v1 and v2. *Note that these analyses are for the passage track.*

3.1 *V1* on MS MARCO v1

First we inspect the results of our *V1* model on MS MARCO v1. This will serve as a baseline for our runs, as we analyze results in numbers that we can compare with prior works and TREC competitions.

Our first-stage model achieves an impressive (at the time of TREC submission) result of 0.355 MRR@10 and 97.6 R@1k, which compared to other models available at that time was pretty substantial. As far as we are aware, the state of the art for a single non-cross-attention model in June 2021 were i) ColBERT [Khattab and Zaharia, 2020], with 0.36 MRR@10 and 97% R@1k, which would be difficult to scale for TREC before the inclusion of binarization[4], and ii) TCT-ColBERT [Lin et al., 2021] with 0.359 MRR@10 and 97% R@1k. We also evaluate on TREC-2019, where our model was also very competitive with the state of the art at the time.

By applying our re-ranking ensemble, we are able to jump from 0.355 to 0.425 MRR@10, which is quite an impressive jump, at the expense of a large inference cost.

Note that in between the beginning of TREC and the writing of this contribution, the state of the art for both single models and two-stage ones has been evolving quickly, with models such as CoCondenser [Gao and Callan, 2021] (0.382 MRR@10 for single-stage) and AR2 [Zhang et al., 2021]

[3] `https://huggingface.co/cross-encoder/ms-marco-MiniLM-L-12-v2`

[4] `https://github.com/stanford-futuredata/ColBERT/tree/binarization`

(0.395 MRR@10 for single-stage). Even our SPLADE model has been able to achieve an MRR@10 of 0.393 by using a combination of pre-training techniques from [Gao and Callan, 2021] and better hard-negatives [5]. For two-stage models, the state of the art has evolved significantly with models such as RocketQAv2 [Ren et al., 2021] claiming 0.419 MRR@10 with only a dense model and a top-50 re-ranker, as well as other models on the official leaderboard [6] reaching up to 0.44 MRR@10, but without much specification on how exactly the re-rankers are trained.

3.2 Experiments on MS MARCO v2 dev1

Result analysis We then started experimenting with training on the MS MARCO v2 dev1. The final result table for our three runs (split between single-stage and two-stage retrieval) is available in Table 1, alongside comparison with TCT-ColBERT[7]. Note that while in full ranking mode, (*V1+V2*) achieves a better result than TCT-ColBERT, it does so with a lot more complexity. On the first-stage front, TCT-ColBERT trained on v2 clearly outperforms (*V1+V2*), while the one trained on v1 is pretty comparable to our *V1*. In the next paragraphs we explain our reasoning and how we ended with our final *V1+V2* run.

Table 1: Experiments on the MS MARCO v2 dev1 dataset. MRR numbers are multiplied by 100 for ease of presentation.

Run	First Stage			Full Ranking			TCT-ColBERT v2	
Metric	*Quick*	*V1*	*V1+V2*	*Quick*	*V1*	*V1+V2*	MS MARCO v1	MS MARCO v2
MRR@10	12.2	13.0	14.1	12.2	17.7	**21.6**	-	-
MRR@100	13.2	14.1	15.5	13.2	18.8	**22.7**	14.7	20.0
R@10	26.7%	28.3%	30.2%	26.7%	35.4%	**39.7%**	27.5%	-
R@100	54.0%	57.2%	57.9%	54.0%	64.3%	**67.4%**	58.8%	64.0%
R@1000	75.7%	80.7%	81.8%	75.7%	80.7%	81.8%	83.2%	**84.5%**

Experimental process This year, there was no triplets file that was provided, so we first extracted the top-100 negatives from BM25 (given by the organizers) and generated triplets with them. In doing so, we noticed that training SPLADE was very dependent on the negatives, and that top-100 negatives does not seem to be sufficient; we expect the same to be true for dense models, but were not able to test it due to time constraints.

We then extracted the top-1000 negatives from BM25 and re-trained our SPLADE models. This time, results were a little bit better, but not enough to justify a run trained solely on MS MARCO v2. The next step would be to generate new hard-negative triplets and re-ranker scores of those triplets (for distillation training), but due to time constraints and dataset size, we were not able to do so. Thus, we took an ensemble of SPLADE models trained on MS MARCO v1 alongside our v2 trained model for our first-stage ranker (*V1+V2*), which improved slightly the results over the *V1* run.

Finally as a re-ranker, we would like to use the same reasoning as for the *V1* run. However due to timing constraints, we had to use BM25 to train re-rankers on the MS MARCO v2. Individually, the re-rankers did not achieve very strong results, but their ensemble was able to improve effectiveness by a large margin. Finally, we decided to consider an ensemble of re-rankers trained solely on v2 and the ones trained solely on v1, in order to reduce the train/test discrepancy, which allowed us to get some final points on MS MARCO v2 dev1.

Conclusion Overall, our experiments with the v2 dev dataset were quite unsuccessful, with our v2-trained model achieving almost the same performance as the *V1* one. We could achieve significant improvements in the dev dataset only by ensembling first-stage models and v2-trained re-rankers. In the next paragraphs, we perform an analysis to better understand what are the reasons for this difference, and why training with the v2 dataset was so complex.

[5] from `https://huggingface.co/datasets/sentence-transformers/msmarco-hard-negatives`

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

[7] available at `https://github.com/castorini/pyserini/blob/3e4c2837a72ef72b1bee08e4132765db51aa7714/docs/experiments-msmarco-v2-tct_colbert-v2.md`

3.3 Analysis of MS MARCO v2 labels

In order to get an idea of how noisy the MS MARCO v2 labels are, we randomly drew 52 training queries and manually re-evaluated the relevance of positive and negative passages. For the positive passage we evaluated whether it was relevant or not for answering the query. For the negative passages, we only looked up to the top-20 'hardest' (from BM25), and evaluated whether at least one false negative was present in this top-20. For 7 of the 52 queries, we were not confident in our manual evaluation, usually because the query was unclear, too general, or passages too short. We excluded these 7 queries. Figure 1 shows the result of our annotation on the 45 remaining queries. About a quarter of positives passages are actually negatives. Moreover, about two third of the queries have *at least* one false negative in the top-20 of BM25. Out of these, we also observed that many had more than one, although we cannot provide an average number because we did not look at all 20 passages for all queries. Overall, only about a quarter of all queries have correct annotation for the top-20. In the next section, we posit that these labelling issues could be the cause of the difference in performance from dev1 to TREC 2021. Also, note that while we expect a lot of false negatives in the MS MARCO collection, the presence of false positives is something new that could possibly change a lot how we perform training on this dataset.

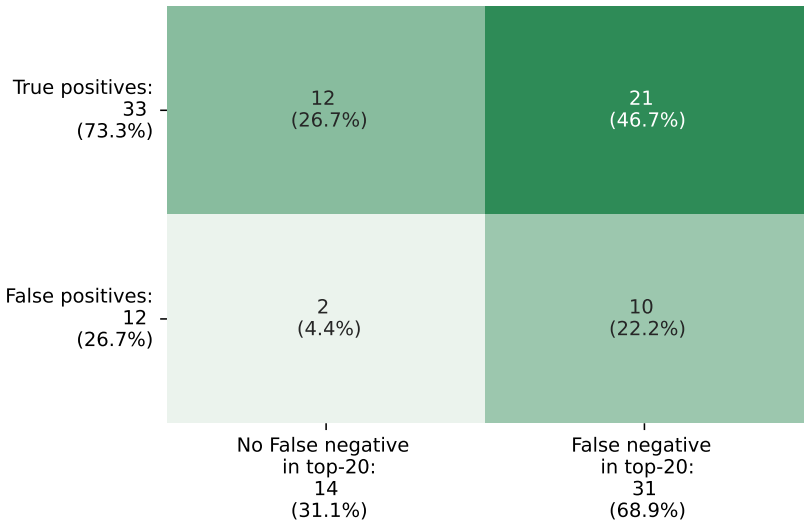

Figure 1: Manual Assessment of 45 queries MS MARCO v2 and their labels: out of the 45 analysed queries, more than a quarter of the positives are actually false positives.

4 TREC DL 2021 - initial analysis

4.1 Overall analysis

Passage track Considering the results on MS MARCO v2 dev1, we were expecting models to have similar performance on TREC 2021, i.e. *V2+V1* > *V1* > *Quick*, with lower performance in absolute terms when compared to what we were used to in the previous TREC competitions (around 0.7 nDCG@10) would be a very large downgrade. However, results turned out to be different, as we can see in Table 2. Compared to the mean of best results per query and the mean of median, the results seem good enough and we are eager to know how it compares to other solutions. Also, our best model (V1) is inline with what we saw in previous years (absolute nDCG@10 around 0.7), but it was not the one we expected to perform best. Because our best model is the one trained on MS MARCO v1, it might indicate that the new dataset has some issues that need to be fixed before training.

We note that [Arabzadeh et al., 2021] recently argued that the noisy MS MARCO labels are problematic, possibly leading to wrong conclusions about the model ranking and true performance. It could be the case here again: before the assessment, we thought that the *V1+V2* model was the

best according to its dev performance. However, we see that after manual assessment, it is quite the opposite conclusion.

Table 2: TREC DL 21 Results on the passage track. nDCG@10 values have been multiplied by 100 for ease of presentation.

Run	First Stage			Full Ranking			Baselines		
Metric	*Quick*	*V1*	*V1+V2*	*Quick*	*V1*	*V1+V2*	BM25	Median	Best
nDCG@10	60.9	**65.3**	60.2	60.9	**73.5**	67.2	44.6	60.0	**83.7**
R@10	13.4%	**16.8%**	16.6%	13.4%	**18.9%**	17.4%	9.60%	-	-
R@100	46.8%	**52.7%**	49.2%	46.8%	**60.4%**	57.0%	32.6%	-	-

Document Track For this track we submitted runs from the same models as the passage ones, using max pooling over passages: the document score is the maximum score over its passages. Therefore we get results that are very inline with what we saw on the passage track (see Table 3). As expected, when compared to the mean of best and mean of medians the document results are inferior to the passage ones.

Table 3: TREC DL 21 Results on the document track. nDCG@10 values have been multiplied by 100 for ease of presentation.

Metric	*Quick*	*V1*	*V1+V2*	*Quick*	*V1*	*V1+V2*	BM25	Median	Best
nDCG@10	60.2	**63.2**	60.6	60.2	**72.2**	68.7	51.2	66.0	**85.7**
Recall@10	8.5%	**9.2%**	8.4%	8.50%	**10.7%**	10.4%	7.8%	-	-
Recall@100	31.3%	**35.6%**	32.5%	31.3%	**41.3%**	39.6%	31.9%	-	-

In-depth analysis In Tables 4 (passage) and Table 5 (document) of the appendix section, we present the results of our best run (*V1*) in all queries and compare it to the mean and median nDCG@10 results. For the passage track, we have 12 queries out of 53 (23%) for which we have the best nDCG@10. On the other hand, we also have 8 queries for which we have a result that is at most 1 nDCG point (0.01) better than the median, with 4 of those 8 where we are worse than the median. This is better than what we expected during the competition / after seeing some results on the dev1 dataset. To better understand the positives and negatives results of our passage submission, we have selected 5 queries: the 2 "worst" performing queries and the 3 "best" performing queries, that we analyze in detail in the following. For more information on the queries and the retrieved documents please see Section A in the appendix.

4.2 Worst performing queries

Query 838273 - what is the methylmalon a. c test: This is the worst performing query of our *V1* method, with a 0.13 nDCG@10 loss compared to the median (and 0.31 points loss compared to the best). When analyzing the top-5 results, what we see is that while we always retrieve a relevant passage, it is not necessarily neither considered very relevant nor perfect. This mostly happens because we are retrieving passages that explain what the test is, but not always for what it is used. This might be linked to the training strategy, that only considers binary relevance (and not a graded one), making it difficult for models with this type of query.

Query 190623 - for what is david w. taylor known: The other worst performing query fails for a completely different reason. While the loss of performance is not as dramatic as in the previous case, we are still 0.03 points worse than the median (and a whooping 0.52 points worse than the best). When analyzing our top-5 results, what we actually see is that the model did not understand that the entity "David W. Taylor" was, and actually retrieved results for e.g. "David Taylor" and "Taylor", or it failed in the same was as before and found things that are related to the entity, but that are not the person itself (e.g. a research center dedicated to the person). This entity problem has actually being studied recently in [Sciavolino et al., 2021].

4.3 Best performing queries

Query 629937 - what does a popped blood vessel in hand feel like: This is a rather hard query as there are some important – but not obvious – words ("in hand"); the *V1* model is able to find perfectly relevant passages for the entire top-5. We do not gain much knowledge from analysing this result other that being able to consider "in finger" as something that is "in hand" to be very helpful. We are able to achieve an nDCG@10 of 0.88, which is 0.3 points better than the median.

Query 1104447 - which kind of continental boundary is formed where two plates move horizontally past one another: This is a very long query with a lot of specific sequences ("continental boundary", "two plates", "move horizontally") for which retrieving correct results without proper context could be very hard; this type of query clearly benefits from contextualization. We are able to get a perfect score nDCG@10 of 1.0, while the median is almost 0.4 points lower.

Query 661905 - what foods should you stay away from if you have asthma: This query has rather weird results. None of the top-10 passages is considered more than relevant (R@10 = 0) but we get a reasonably good nDCG@5 and nDCG@10 (0.5). All of the top-5 are scored only as relevant, but to us should be considered as highly relevant, especially when compared to the answers that were considered highly relevant. Nonetheless our model gets the best result at 0.5 nDCG@10, while the median is of only 0.28 which is on the top-5 of worst median scores.

5 Conclusion

For the TREC DL 21 competition, we submitted runs based on several variants of SPLADE for first-stage ranking, followed by an ensemble of BERT re-rankers trained with hard negatives selected by SPLADE. Much to our surprise, the run trained on MS MARCO v1 obtained the best results on the track, even though evaluation was done on the new dataset. More surprising to us is that the results on the dev set of MS MARCO v2 do not seem to correlate with the TREC results, *contrary to previous years*. We qualitatively inspect the best and worse performing queries and find some advantages and issues with our models, that we aim to improve. Finally, the results seem to indicate that our document approach (i.e. performing everything with passages and mapping the *ids* to documents) was not as successful as our passage runs.

References

[Arabzadeh et al., 2021] Arabzadeh, N., Vtyurina, A., Yan, X., and Clarke, C. L. A. (2021). Shallow pooling for sparse labels.

[Clark et al., 2020] Clark, K., Luong, M.-T., Le, Q. V., and Manning, C. D. (2020). Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*.

[Craswell et al., 2021] Craswell, N., Mitra, B., Yilmaz, E., and Campos, D. (2021). Overview of the trec 2020 deep learning track.

[Craswell et al., 2020] Craswell, N., Mitra, B., Yilmaz, E., Campos, D., and Voorhees, E. M. (2020). Overview of the trec 2019 deep learning track. *arXiv preprint arXiv:2003.07820*.

[Devlin et al., 2018] Devlin, J., Chang, M., Lee, K., and Toutanova, K. (2018). BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

[Formal et al., 2021a] Formal, T., Lassance, C., Piwowarski, B., and Clinchant, S. (2021a). Splade v2: Sparse lexical and expansion model for information retrieval.

[Formal et al., 2021b] Formal, T., Piwowarski, B., and Clinchant, S. (2021b). 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 '21, page 2288–2292, New York, NY, USA. Association for Computing Machinery.

[Gao and Callan, 2021] Gao, L. and Callan, J. (2021). Unsupervised corpus aware language model pre-training for dense passage retrieval. *arXiv preprint arXiv:2108.05540*.

[Gao et al., 2021] Gao, L., Dai, Z., and Callan, J. (2021). Rethink training of bert rerankers in multi-stage retrieval pipeline.

[Hofstätter et al., 2021] Hofstätter, S., Lin, S.-C., Yang, J.-H., Lin, J., and Hanbury, A. (2021). Efficiently Teaching an Effective Dense Retriever with Balanced Topic Aware Sampling. In *Proc. of SIGIR*.

[Hofstätter et al., 2020] Hofstätter, S., Althammer, S., Schröder, M., Sertkan, M., and Hanbury, A. (2020). Improving efficient neural ranking models with cross-architecture knowledge distillation.

[Khattab and Zaharia, 2020] Khattab, O. and Zaharia, M. (2020). Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, page 39–48, New York, NY, USA. Association for Computing Machinery.

[Lin et al., 2021] Lin, S.-C., Yang, J.-H., and Lin, J. (2021). In-batch negatives for knowledge distillation with tightly-coupled teachers for dense retrieval. In *Proceedings of the 6th Workshop on Representation Learning for NLP (RepL4NLP-2021)*, pages 163–173.

[Liu et al., 2019] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

[Nogueira and Cho, 2019] Nogueira, R. and Cho, K. (2019). Passage re-ranking with bert.

[Ren et al., 2021] Ren, R., Qu, Y., Liu, J., Zhao, W. X., She, Q., Wu, H., Wang, H., and Wen, J.-R. (2021). Rocketqav2: A joint training method for dense passage retrieval and passage re-ranking.

[Sciavolino et al., 2021] Sciavolino, C., Zhong, Z., Lee, J., and Chen, D. (2021). Simple entity-centric questions challenge dense retrievers.

[Wang et al., 2020] Wang, W., Wei, F., Dong, L., Bao, H., Yang, N., and Zhou, M. (2020). Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *arXiv preprint arXiv:2002.10957*.

[Xiong et al., 2021] Xiong, L., Xiong, C., Li, Y., Tang, K.-F., Liu, J., Bennett, P. N., Ahmed, J., and Overwikj, A. (2021). Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *International Conference on Learning Representations*.

[Zhang et al., 2021] Zhang, H., Gong, Y., Shen, Y., Lv, J., Duan, N., and Chen, W. (2021). Adversarial retriever-ranker for dense text retrieval.

A Analysed Queries

A.1 Worst performing queries

A.1.1 Query 838273

Text: what is the methylmalon a. c test

nDCG@10 comparison: *V1*: 46.3; Median: 59.2; Best: 77.2

Top-5 passages retrieved by *V1*:

1. *Relevance score: 1-Relevant.* The methylmalonic acid test is performed on blood plasma or blood serum taken from a standard blood draw. Typically, a doctor or nurse will take a blood sample from your arm in a clinical setting. The blood will be collected in a tube and sent to a lab for analysis. Once the lab reports the results, your doctor will be able to provide you with more information about the results and what they mean.
2. *Relevance score: 1-Relevant.* The methylmalonic acid test is performed on blood plasma or blood serum taken from a standard blood draw. Typically, a doctor or nurse will take a blood sample from your arm in a clinical setting. The blood will be collected in a tube and sent to a lab for analysis. Once the lab reports the results, your doctor will be able to provide you with more information about the results and what they mean.
3. *Relevance score: 3-Perfect.* Methylmalonic acid test. The methylmalonic acid blood test measures the amount of methylmalonic acid in the blood. The methylmalonic acid test may be used to help diagnose an early or mild vitamin B12 deficiency. It may be ordered by itself or along with a homocysteine test as a follow-up to a vitamin B12 test result that is in the lower end of the normal range.

4. *Relevance score: 3-Perfect.* Methylmalonic acid test. The methylmalonic acid blood test measures the amount of methylmalonic acid in the blood. The methylmalonic acid test may be used to help diagnose an early or mild vitamin B12 deficiency. It may be ordered by itself or along with a homocysteine test as a follow-up to a vitamin B12 test result that is in the lower end of the normal range.
5. *Relevance score: 1-Relevant.* Methylmalonate Test - More Information. The methylmalonic acid test, also known as a methylmalonic acid blood test, methylmalonate lab test and an MMA level test, measures the methylmalonic acid blood level.

A.1.2 Query 190623

Text: for what is david w. taylor known

nDCG@10 comparison: *V1*: 40.5; Median: 44.4; Best: 92.6

Top-5 documents from *V1*:

1. *Relevance score: 3-Perfect.* Rear Adm. David W. Taylor. Rear Admiral David Watson Taylor, USN (March 4, 1864 - July 28, 1940) was a naval architect and engineer of the United States Navy. He served during World War I as Chief Constructor of the Navy, and Chief of the Bureau of Construction and Repair. Taylor is best known as the man who constructed the first experimental towing tank ever built in the United States.
2. *Relevance score: 0-Irrelevant.* World Champ David taylor the magic man. World Champ. David taylor the magic man. David Taylor, widely known as The Magic Man, is a 4x NCAA All-American, 4x BIG 10 Champion, and a 2x NCAA Champion – and he's just getting started. Having wrapped up his NCAA career in March of 2014, David is just getting started on his international career and ultimately, his quest for Gold in Tokyo, 2020.
3. *Relevance score: 1-Relevant.* History. The facility was previously known as the David W. Taylor Naval Ship Research and Development Center; it was renamed "David Taylor Research Center (DTRC)" in 1987 and later became the " Carderock Division of the Naval Surface Warfare Center " in 1992.
4. *Relevance score: 0-Irrelevant.* Taylor is best known for being the former lead singer of the music group Kool & the Gang. Taylor worked as an amateur night club singer and joined his first band at 13. He joined Kool & The Gang in 1979 and became their lead singer in 1979.
5. *Relevance score: 0-Irrelevant.* Taylor is predominately known for his roles as Romeo in Student Bodies, and Kwest in Instant Star. He played the role of Lewis 'Lou' Young in the Canadian police drama television series Flashpoint until his character was killed in the 23rd episode (part of the second season).

A.2 Best performing queries

A.2.1 Query 629937

Text: what does a popped blood vessel in hand feel like

nDCG@10 comparison: *V1*: 88.2; Median: 59.2; Best: 88.2

Top-5 passages retrieved by *V1*:

1. *Relevance score: 3-Perfect.* Popped Blood Vessel In Hand Symptoms. The condition will show under the layer of transparent skin and is characterized by: Bright red or dark appearance in the outermost layer of the skin. A feeling of minor pain upon contact.
2. *Relevance score: 3-Perfect.* Popped Blood Vessel In Hand Symptoms. The condition will show under the layer of transparent skin and is characterized by: Bright red or dark appearance in the outermost layer of the skin. A feeling of minor pain upon contact.

3. *Relevance score: 3-Perfect.* Symptoms of popped blood vessel in hand: Bursting of blood vessels or popped blood vessel in hand and subsequent bleeding under the skin of your hands has the following symptoms: Swelling in the affected area of the skin. In some cases, an associated fracture in the bone may take place. Some pain (minor) may be felt upon touching the affected area.
4. *Relevance score: 3-Perfect.* Symptoms of a Popped Blood Vessel in the Finger. Onset of this condition is sudden or may follow after a minor injury. Sudden onset of intense burning pain felt in the hand or finger. Sudden localized swelling.
5. *Relevance score: 3-Perfect.* Symptoms of a Popped Blood Vessel in the Finger. Onset of this condition is sudden or may follow after a minor injury. Sudden onset of intense burning pain felt in the hand or finger. Sudden localized swelling.

A.2.2 Query 1104447

Text: which kind of continental boundary is formed where two plates move horizontally past one another

nDCG@10 comparison: *V1*: 100.0; Median: 62.2; Top: 100.0

Top-5 passages retrieved by *V1*:

1. *Relevance score: 3-Perfect.* Transform boundaries. Most boundaries are either convergent or divergent, but transform boundaries occur in a few places to accommodate lateral motion, where plates move horizontally past one another. This type of boundary is very rare on continents, but they are dramatic where they do occur.
2. *Relevance score: 3-Perfect.* Transform boundaries. Most boundaries are either convergent or divergent, but transform boundaries occur in a few places to accommodate lateral motion, where plates move horizontally past one another. This type of boundary is very rare on continents, but they are dramatic where they do occur.
3. *Relevance score: 3-Perfect.* In some places, two plates move apart from each other; this is called a diverging plate boundary. Elsewhere two plate move together; this is a converging plate boundary. Finally plates can also slide past each other horizontally. This is called a transform plate boundary.
4. *Relevance score: 3-Perfect.* The way one plate moves relative to another determines the type of boundary: spreading, where the two plates move away from each other; subduction, where the two plates move toward each other, with one sliding beneath the other; and transform, where the two plates slide horizontally past each other.
5. *Relevance score: 3-Perfect.* The way one plate moves relative to another determines the type of boundary: spreading, where the two plates move away from each other; subduction, where the two plates move toward each other, with one sliding beneath the other; and transform, where the two plates slide horizontally past each other.

A.2.3 Query 661905

Text: what foods should you stay away from if you have asthma

nDCG@10 comparison: *V1*: 50.0; Median: 28.2; Top: 50.0.

Top-5 passages retrieved by *V1*:

1. *Relevance score: 1-Relevant.* Beans, cabbage, fried foods, carbonated drinks, onion and garlic are foods to avoid if you have asthma. World Asthma Day 2018: Asthmatic patients should avoid processed foods. 2. People who have asthma should avoid processed foods since they come with added preservatives and flavours.
2. *Relevance score: 1-Relevant.* Foods to Avoid if You Have Asthma. 1. Cheese. Cheese is a troublesome dairy product that has been linked to the development of asthma in several studies, and has also been shown to exacerbate symptoms (2).

3. *Relevance score: 1-Relevant.* Cut out foods that aggravate your asthma. People with asthma may have certain food triggers that are typically unique to each person. In general, individuals with asthma should avoid the common triggers such as eggs, fish, peanuts, soy, yeast, cheese, wheat and rice.
4. *Relevance score: 1-Relevant.* Cut out foods that aggravate your asthma. People with asthma may have certain food triggers that are typically unique to each person. In general, individuals with asthma should avoid the common triggers such as eggs, fish, peanuts, soy, yeast, cheese, wheat and rice.
5. *Relevance score: 1-Relevant.* Citrus fruits and tomatoes. People with asthma should avoid citrus fruits and tomatoes. Both contain a lot of nutrients and fiber, which benefit your health, but they also have some components that can worsen asthma symptoms. You don't have to stop eating them overnight, but at least make an effort to lower your consumption.

B Query by query result tables

Table 4: Query level results for *V1* on the passage full ranking task. nDCG@10 results are given and have been multiplied by 100 for ease of presentation. Numbers are underscored if *V1* is not able to outperform the median by at least 1 point and are bolded if it gets the best nDCG@10 for a query.

Query Id	*V1*	Best	*V1* vs Best	Median	*V1* vs Median
2082	88.7	100	-11.3	82.9	5.8
23287	54.3	65.3	-11.0	26.5	27.8
30611	31.3	75.7	-44.4	31.0	0.2
112700	47.8	68.9	-21.1	47.8	0.0
168329	69.4	74.0	-4.6	68.5	0.9
190623	40.5	92.6	-52.1	44.4	-3.9
226975	44.6	66.8	-22.2	28.9	15.7
237669	75.7	91.9	-16.3	54.0	21.7
253263	69.6	89.1	-19.5	62.1	7.5
300025	78.3	83.9	-5.6	53.9	24.4
300986	69.9	79.4	-9.5	62.3	7.6
337656	65.2	71.9	-6.7	26.9	38.2
364210	**84.8**	**84.8**	**0.0**	**78.7**	**6.1**
395948	68.4	72.7	-4.3	60.9	7.5
421946	77.2	92.6	-15.4	65.7	11.5
493490	94.3	100	-5.7	85.5	8.8
505390	70.9	86.8	-15.9	66.1	4.9
508292	50.0	53.9	-3.9	46.8	3.2
540006	86.5	88.0	-1.6	63.0	23.4
596569	92.6	100	-7.4	56.5	36.1
629937	**88.2**	**88.2**	**0.0**	**57.2**	**31.0**
646091	74.4	84.0	-9.7	65.4	8.9
647362	**95.3**	**95.3**	**0.0**	**62.7**	**32.6**
661905	**50**	**50**	**0.0**	**28.2**	**21.8**
681645	**71.8**	**71.8**	**0.0**	**52.1**	**19.7**
688007	51.1	74.6	-23.4	53.2	-2.1
707882	**100**	**100**	**0.0**	**95.2**	**4.9**
764738	62.2	86.6	-24.3	60.4	1.9
806694	95.4	96.6	-1.3	81.2	14.2
818583	61.1	76.2	-15.1	47.3	13.8
832573	46.3	77.2	-30.9	59.2	-12.9
835760	77.7	86.0	-8.4	71.9	5.7
845121	57.9	61.2	-3.2	35.1	22.8
935353	91.8	94.2	-2.4	44.4	47.4
935964	**97.8**	**97.8**	**0.0**	**75.8**	**22.0**
952262	**100**	**100**	**0.0**	**93.6**	**6.4**
952284	60.9	77.8	-16.9	43.3	17.6
975079	67.9	76.1	-8.2	58.1	9.8
1006728	51.7	64.4	-12.7	34.9	16.7
1040198	68.1	88.7	-20.7	59.3	8.7
1104300	**100**	**100**	**0.0**	**92.2**	**7.8**
1104447	**100**	**100**	**0.0**	**62.2**	**37.8**
1107704	59.5	64.9	-5.3	24.5	35.1
1107821	71.7	91.0	-19.4	71.6	0.1
1109840	80.3	82.9	-2.6	67.4	12.9
1110996	77.9	94.6	-16.7	73.5	4.4
1111577	71.5	75.8	-4.2	51.5	20.1
1113361	**100**	**100**	0.0	**95.5**	**4.5**
1117243	**84.8**	**84.8**	0.0	**75.4**	**9.4**
1118716	80.9	84.5	-3.6	63.2	17.6
1121909	85.2	95.8	-10.6	66.0	19.2
1128632	80.9	90.1	-9.2	82.2	-1.2
1129560	80.7	86.1	-5.4	62.5	18.2

Table 5: Query level results for *V1* on the document full ranking task. nDCG@10 results are given and have been multiplied by 100 for ease of presentation. Numbers are underscored if *V1* is not able to outperform the median by at least 1 point and are bolded if it gets the best nDCG@10 for a query.

Query Id	*V1*	Max	*V1* vs Max	Median	*V1* vs Median
2082	76.0	100	-23.9	92.3	-16.3
23287	67.6	71.5	-3.9	37.9	29.6
30611	78.3	88.4	-10.0	67.6	10.7
112700	53.5	72.4	-18.7	48.1	5.5
168329	84.2	93.7	-9.4	80.8	3.5
190623	38.7	77.1	-38.4	58.9	-20.2
226975	48.1	57.0	-8.9	30.2	17.9
237669	78.8	89.8	-10.9	74.9	3.9
253263	65.3	83.9	-18.6	46.2	19.1
300025	**90.9**	**90.9**	**0**	**60.2**	**30.6**
300986	64.7	81.8	-17.1	66.5	-1.8
337656	49.5	90.1	-40.6	45.2	4.3
364210	89.9	97.8	-7.9	91.7	-1.8
395948	73.7	95.5	-21.8	80.9	-7.2
421946	53.8	78.4	-24.6	66.2	-12.3
493490	87.8	100	-12.2	95.1	-7.3
505390	85.8	93.3	-7.5	81.7	4.1
508292	54.7	71.4	-16.7	56.3	-1.6
540006	87.3	89.7	-2.4	79.3	8.0
596569	76.9	97.7	-20.8	63.5	13.4
615176	54.1	78.2	-24.1	49.6	4.5
629937	75.3	80.9	-5.7	57.8	17.4
632075	28.3	53.8	-25.5	22.1	6.3
646091	68.6	88.3	-19.7	73.4	-4.8
647362	**100**	**100**	**0**	**72.7**	**27.3**
661905	56.4	80.3	-23.9	39.1	17.3
681645	70.7	81.6	-10.9	61.5	9.1
688007	88.4	90.9	-2.6	79.7	8.7
707882	80.4	93.4	-12.9	78.3	2.2
764738	72.9	84.8	-11.9	73.1	-0.2
806694	85.0	95.7	-10.6	74.9	10.1
818583	31.1	73.3	-42.2	49.4	-18.3
832573	79.8	93.9	-14.1	75.7	4.1
835760	86.6	95.7	-9.1	86.6	0.0
845121	72.3	94.6	-22.2	63.3	9.1
935353	68.2	79.9	-11.8	49.4	18.8
935964	97.8	97.9	-0.1	82.6	15.2
952262	**100**	**100**	**0**	**100**	**0.0**
952284	52.5	64.5	-12.0	44.9	7.6
975079	57.1	82.6	-25.4	53.3	3.9
1006728	34.1	56.9	-22.8	40.2	-6.1
1040198	61.2	76.4	-15.2	47.4	13.8
1103547	89.0	100	-11.0	100	-11.0
1104300	92.8	100	-7.2	92.0	0.8
1104447	97.9	100	-2.1	75.8	22.1
1107704	82.4	88.2	-5.7	61.4	21.0
1107821	75.2	76.7	-1.6	57.8	17.4
1109840	53.5	67.8	-14.3	49.3	4.2
1110996	93.7	100	-6.3	93.2	0.5
1111577	78.7	86.9	-8.3	71.3	7.4
1113361	96.8	100	-3.2	96.8	0.0
1117243	75.5	91.5	-15.9	56.2	19.3
1117298	70.9	82.9	-12.1	66.9	4.0
1118716	65.5	87.1	-21.6	39.7	25.8
1128632	86.8	100	-13.2	92.9	-6.2