NAVER LABS EUROPE (SPLADE) @ TREC DEEP LEARNING 2022

Carlos Lassance, Stephane Clinchant
Naver Labs Europe
France
carlos.lassance@naverlabs.com, stephane.clinchant@naverlabs.com

ABSTRACT

This paper describes our participation to the 2022 TREC Deep Learning challenge. We submitted runs to all four tasks, with a focus on the full retrieval passage task. The strategy is almost the same as 2021, with first stage retrieval being based around SPLADE, with some added ensembling with ColBERTv2 and DocT5. We also use the same strategy of last year for the second stage, with an ensemble of re-rankers trained using hard negatives selected by SPLADE. Initial result analysis show that the strategy is still strong, but is still unclear to us what next steps should we take.

1 Introduction

In this paper, we detail our TREC 2022 Deep Learning track submission, based on the latest improvements of the SPLADE model [Lassance and Clinchant, 2022, Formal et al., 2022]. Compared to last year, there were only three changes: i) The use of Rocchio [Joachims, 1996] with SPLADE to improve first stage ranking, ii) A first stage ensemble of different SPLADE models, ColBERTv2 [Santhanam et al., 2021] and DocT5 [Nogueira and Lin, 2019], and iii) Inclusion of a MonoT5 based reranker using T0pp. In total, we submitted more than 30 runs, most of them baselines using the models from [Lassance and Clinchant, 2022, Formal et al., 2022] that are available at https://huggingface.co/naver. As per last year, we focus on passage retrieval and for the document task, we simply score a document by taking the maximum score over its passages.

This year we decided to make this notebook more streamlined compared to last year. For a more thorough introduction of the models used here, we invite the reader to check the following articles: SPLADE training [Formal et al., 2022, Lassance and Clinchant, 2022], ColBERTv2[Santhanam et al., 2021], DocT5 [Nogueira and Lin, 2019], Training style we used for our rerankers [Gao et al., 2021] and T5 based reranking [Nogueira et al., 2020].

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

For the first stage this year, we only have one addition which is the use of Rocchio on SPLADE via Anserini. Notably: we use the following methods i) SPLADE++ EnsembleDistil [Formal et al., 2022], ii) SPLADE++ SelfDistil [Formal et al., 2022], iii) ColBERTv2[Santhanam et al., 2021], and iv) DocT5 [Nogueira and Lin, 2019]

2.2 Second Stage

As per last year competition we use a mix of different PLMs as rerankers for which training is inspired by [1] [Gao et al., 2021] and using negatives from SPLADE. Namely we use the following PLMs: Deberta-v2 xxlarge, Deberta-v3 large, Electra-large, T0pp, Albert-v2-xxlarge, Roberta-Large. We make the rerankers available at `https://huggingface.co/naver/\{name\}` with the models being named: trecdl22-crossencoder-{debertav2,debertav3,electra,albert,roberta}.

One main difference from last year is that we added to the mix a reranker based on MonoT5 [Nogueira et al., 2020]. For that reranker we start from the T0pp model which is an 11B PLM. We train it both with the MonoT5 loss and the InfoLCE from [Gao et al., 2021]. Unfortunately, this model did not worked as well as it would be expected from its size, and in hindsight would be better served to go fully on the InfoLCE loss, as demonstrated by a post-TREC paper [Zhuang et al., 2022].

2.3 Ensembling

We also have applied ensembling in order to improve our results. This year we used ranx [Bassani and Romelli, 2022] to generate all our ensembles, using average normalized score over the ensembles, unless explicitly noted. The normalized score uses the min and max values of the query so that, for each model, the best score is 1 and the lowest one 0.

2.4 Runs submitted to TREC

We submitted a total of 32 runs, 16 for the passage task and 16 for the document task. All of the document runs are the same as the passage ones, just with the added max pooling over the passages over the same document. For the 16 runs, we have the 3 official full ranking runs, 3 official rerank runs and 10 baselines runs (5 with rocchio, 5 without).

2.4.1 Official full ranking runs

For our three full-ranking runs, we always use the same 6 rerankers for the second stage, while we vary the first stage by adding a new model for each new run:

- *NLE_SPLADE_RR*: First stage ensemble of naver/splade-cocondenser-ensembledistil and naver/splade-cocondenser-selfdistil, both models using Rocchio, followed by the ensemble reranking of 6 models.
- *NLE_SPLADE_CBERT_RR*: Same as before, but adding ColBERTv2 [Santhanam et al., 2021] [2] to the first stage ensemble.
- *NLE_SPLADE_CBERT_DT5_RR*: Same as the previous one, but adding DocT5 [Nogueira and Lin, 2019] to the first stage ensemble.

2.4.2 Official rerank runs

For our three official reranking runs, we submit the ensemble of the 6 rerankers using either the average normalized score or average normalized Condorcet score and a run based on T0pp.

2.4.3 Baseline runs

For our baseline runs, the idea was to simply run the four models from our SIGIR 2022 contributions [Formal et al., 2022, Lassance and Clinchant, 2022] that have been made available in huggingface and an ensemble of the two best Splade++ models. To those 5 runs, we also add the Rocchio versions of each run, totaling 10 baselines.

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

[2] `https://downloads.cs.stanford.edu/nlp/data/colbert/colbertv2/colbertv2.0.tar.gz`

3 Analysis on MS MARCO v1 and v2

In order to generate our final runs, we used as development scores the dev set of MSMARCO v1 and the last three TREC competitions. Results for our final runs are available in Table 3. From it we drew some conclusions (passage track):

1. Rocchio [Joachims, 1996] improves the performance of SPLADE when labels are not sparse (i.e. outside of MSMARCO v1 dev)
2. As expected, TREC19, 20, and 21 are biased to techniques that participated in the competition, we notice that the ensemble of SPLADE models is way more competitive in 21 where it was first applied, while the effectiveness of DT5 greatly diminishes over time (as more techniques are added to the fold)
3. Moreover, we are able to "beat" the best nDCG@10 results for TREC2019 and 21, but not for 2020, while we are able to increase mAP in all years. The TREC2020 reranker that got the best results seems very impressive, especially because it gets those results reranking BM25 directly.
4. There is not so much correlation between results on the MSMARCO dev and TREC sets. For example, SPLADE_EFFICIENT_V performs the best out of all baselines on MSMARCO dev but is not on the TOP5 when we consider the average of TREC results.
5. More-so, while our efficient models get very good results on MSMARCO, they struggle on TREC (something we already had seen). We are still not sure of the reason for this decrease in performance
6. Statistically significant results are hard to get on TREC. We do not report on the Table, but most first-stage models are considered "the same" when we apply statistical significance testing, maybe there would be a way of joining all years in order to get more queries and thus more significant results? The average as we use here does not seem like a good idea (see point 1)
7. Over the rerankers it seems that Electra-large took the cake as the best model we trained, while T0pp struggled on MSMARCO, but was surprisingly good on TREC21. However, adding the models on an ensemble almost always helped.
8. While the rerankers improve over the first stage models, it is slightly deceiving as they add a lot more cost for inference. However, this is something that seems to be changing (the gap is larger on TREC21)

Table 1: Results for the developmental process of our runs. *:SPLADE models considered without rocchio for ensembles evaluated on MSMARCO dev set.

Reranker	Dev MSMARCO v1	TREC 2019		TREC 2020		TREC 2021		Average over TREC	
	MRR@10	nDCG@10	mAP@1000	nDCG@10	mAP@1000	nDCG@10	mAP@100	nDCG@10	mAP
BASELINE - Best@year	46.3	76.5	50.49	80.3	56.43	74.9	39.23	77.2	48.7
Runs sent as baselines									
SPLADE_PP_SDISTIL	37.8	73.6	50	72.8	51.4	68	32.4	71.5	44.6
SPLADE_PP_EDISTIL	38.3	73.2	50.5	72.0	50	68.5	33.3	71.2	44.6
SPLADE_EFFICIENT_V	**38.8**	71.5	47.4	71.5	48.4	67.9	31.9	70.3	42.6
SPLADE_EFFICIENT_VI-BT	38.0	70.3	46.2	69.8	47.4	65.7	29.3	68.6	41.0
SPLADE_PP_SDISTIL_ROCCHIO	32.9	71.2	51.4	72.8	50.8	70.0	34.6	71.3	45.6
SPLADE_PP_EDISTIL_ROCCHIO	32.8	71.5	50.7	73.7	53.0	69.7	34.4	71.6	46.0
SPLADE_EFFICIENT_V_ROCCHIO	32.2	70.1	49.1	72.6	49.8	67.7	33.0	70.1	44.0
SPLADE_EFFICIENT_VI-BT_ROCCHIO	30.2	71.4	49.5	67.2	46.1	64.1	29.0	67.6	41.5
SPLADE_ENSEMBLE_PP	38.4	72.7	50.4	73.1	51.4	69.2	33.9	71.7	45.2
SPLADE_ENSEMBLE_PP_ROCCHIO	33.7	71.8	51.2	73.7	52.5	70.8	35.3	72.1	46.3
First Stage ensembles (not sent. just a base of comparison)									
Run 1 - NLE_SPLADE	38.4*	71.8	51.2	73.7	52.5	70.8	35.3	72.1	46.3
Run 2 - NLE_SPLADE_CBERT	**39.5***	72.8	52	75.7	54.2	71.5	36.9	73.3	47.7
Run 3 - NLE_SPLADE_CBERT_DT5	39.3*	**76.3**	**54.3**	75.3	54.3	70.7	37.1	74.1	48.6
Rerankers over Run 3 (not sent. just a base of comparison)									
Deberta-v2 xxlarge	41.5	75.7	56.0	76.6	57	73.6	39.9	75.3	51.0
Deberta-v3 large	40.7	**77.3**	**57.2**	75.7	57	73.8	39.5	75.6	51.2
Electra-large	**41.9**	76.9	**57.3**	**77.1**	**57.6**	**74.2**	**39.9**	**76.1**	**51.6**
T0pp	38.8	73.5	55.1	74.5	55.4	73.4	**39.9**	73.8	50.1
Albert-v2-xxlarge	41.7	77.2	57.0	76.9	57.3	73.4	38.5	75.8	50.9
Roberta-Large	41.5	75.3	55.6	75.8	56.2	69.8	36.5	73.6	49.4
Final runs									
Run 1 - NLE_SPLADE_RR	**43.1**	78.6	58.8	**79.2**	**59.9**	75.5	42.5	77.8	53.7
Run 2 - NLE_SPLADE_COLBERT_RR	43.0	78.4	59.0	79	59.3	75.4	42.6	77.6	53.6
Run 3 - NLE_SPLADE_COLBERT_DT5_RR	42.9	**78.8**	**59.9**	78.7	59.4	75.5	42.7	77.7	54.0

4 TREC DL 2022 - initial analysis

As per the previous year, we mostly focused on the passage track, and thus report only results and analysis for it. Results are made available on Table 4 . We drew some initial conclusions and are still analyzing results:

1. The gap between the first stage and reranked runs increased compared to previous years.
2. The gap between our best run and the best possible run was stable compared to last year (0.102 difference last year, 0.106 this year), however our best run increased the gap to the “median” run[3] (0.135 last year, and 0.177 this year). While the former shows that we maintained the goodness of our runs, the latter is probably a consequence of point 1 (larger gap between first stage and reranking).
3. Our three runs performed almost the same, which is expected given the results we had seen on the development set, but is kinda frustrating as improving the first stage ranking seems to plateau after reranking.
4. As it had happened on the development set, Rocchio improved the results from SPLADE.
5. There seems to be a lack of consistency on our ensembling process, making that various of our runs got “the best” result for different queries. What is more worrying is the case of query 2002533, where our first stage models got the best nDCG@10 over all submitted runs, but our rerankers got a more average score. More intelligent ensembling seems to be needed. More details on Table 3
6. Over all our trained networks we only used the training set of MSMARCO v1.
7. Complementing the last point, there were not so many new things we added this year, and currently, it seems that we would do the same for next year as well.

Table 2: Results for the TREC 2022 passage track.

Method	ndcg@10	mAP (100)
Runs sent as baselines		
SPLADE_PP_SDISTIL	57.05	18.46
SPLADE_PP_EDISTIL	57.86	18.01
SPLADE_EFFICIENT_V	55.09	16.31
SPLADE_EFFICIENT_VI-BT	52.71	14.52
SPLADE_PP_SDISTIL_ROCCHIO	58.97	19.68
SPLADE_PP_EDISTIL_ROCCHIO	59.17	19.23
SPLADE_EFFICIENT_V_ROCCHIO	54.52	17.25
SPLADE_EFFICIENT_VI-BT_ROCCHIO	50.84	14.52
SPLADE_ENSEMBLE_PP	57.89	18.62
SPLADE_ENSEMBLE_PP_ROCCHIO	59.91	20.05
Runs		
Run 1 - NLE_SPLADE_RR	70.92	29.77
Run 2 - NLE_SPLADE_COLBERT_RR	71.41	29.63
Run 3 - NLE_SPLADE_COLBERT_DT5_RR	71.45	29.50

5 Conclusion

For the TREC DL 22 competition, we submitted runs based on our recent SPLADE advancements for first-stage ranking, followed by an ensemble of re-rankers trained with hard negatives selected by SPLADE. While this is a very rough draft, and we are still analyzing the results, it seems that this year there is a larger gap between the first stage runs and the reranked ones. Also, compared to the best results possible per query, our submitted runs get results that are inline with last year.

[3] the run that got the median score on all queries, which is different from the “median” submission

Table 3: Some query examples and comments

query id	query	overall results	our results
2053884	when a house goes into foreclosure what happens to items on the premises	Has two distinct wh propositions (when and what). It was the query with the highest distance between median nDCG@10 (0.07) and best nDCG@10 (1.0)	SPLADE and BM25 are completely lost in this query (0 nDCG@10), but rerankers saved the day.
2002533	how much average cost to plan a 8' tree?	model has to understand that 8' is 8 feet. Has the worst best nDCG@10 at 0.42 and median is pretty low as well (0.07)	Differently from the previous one, here our first stage models shine (best nDCG@10 at 0.42) and the rerankers suffered (0.24)
2003157	how to cook frozen ham steak on nuwave oven	not so sure why it was so hard, maybe nuwave got badly tokenized? Had the worst PREC@10 (best model got only 3 positives in the top10).	Our models perform well, very close to the best nDCG@10 and we get the best PREC@10.
2028378	when is trial by jury used	also not sure why it is so hard, but had the worst best mAP (only 0.06). It has too many positives and many duplicates	Average results for our models, but the number of duplicates makes our mAP suffer

References

[Bassani and Romelli, 2022] Bassani, E. and Romelli, L. (2022). ranx. fuse: A python library for metasearch. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 4808–4812.

[Formal et al., 2022] Formal, T., Lassance, C., Piwowarski, B., and Clinchant, S. (2022). From distillation to hard negative sampling: Making sparse neural ir models more effective. *arXiv preprint arXiv:2205.04733*.

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

[Joachims, 1996] Joachims, T. (1996). A probabilistic analysis of the rocchio algorithm with tfidf for text categorization. Technical report, Carnegie-mellon univ pittsburgh pa dept of computer science.

[Lassance and Clinchant, 2022] Lassance, C. and Clinchant, S. (2022). An efficiency study for splade models. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2220–2226.

[Nogueira et al., 2020] Nogueira, R., Jiang, Z., and Lin, J. (2020). Document ranking with a pretrained sequence-to-sequence model. *arXiv preprint arXiv:2003.06713*.

[Nogueira and Lin, 2019] Nogueira, R. and Lin, J. (2019). From doc2query to docttttquery.

[Santhanam et al., 2021] Santhanam, K., Khattab, O., Saad-Falcon, J., Potts, C., and Zaharia, M. (2021). Colbertv2: Effective and efficient retrieval via lightweight late interaction. *arXiv preprint arXiv:2112.01488*.

[Zhuang et al., 2022] Zhuang, H., Qin, Z., Jagerman, R., Hui, K., Ma, J., Lu, J., Ni, J., Wang, X., and Bendersky, M. (2022). Rankt5: Fine-tuning t5 for text ranking with ranking losses.