MLIA-DAC@TREC CAsT 2022: Sparse Contextualized Query Embedding

Le Hai Nam[1], Thomas Gerald[2], Thibault Formal[1,3], Jian-Yun Nie[4], Benjamin Piwowarski[1], and Laure Soulier[1]

[1] Sorbonne Université, CNRS, ISIR, F-75005 Paris, France
{hai.le@etu.sorbonne-universite.com,
thibault.formal@corbonne-universite.fr, benjamin@piwarski.fr,
laure.soulier@sorbonne-universite.fr}

[2] Université Paris Saclay, CNRS, SATT, LISN, 91400 Orsay, France
thomas.gerald@lisn.upsaclay.fr

[3] Naver Labs Europe, Meylan, France
thibault.formal@naverlabs.com

[4] University of Montreal, Montreal, Canada
nie@iro.umontreal.ca

Abstract. We extend SPLADE, a sparse information retrieval model, as our first stage ranker for the conversational task. This end-to-end approach achieves a high recall (as measure on TREC CAsT 2021). To further increase the effectiveness of our approach, we train a T5-based re-ranker. This working note fully describes our model and the four runs submitted to TREC CAsT 2022.

1 Introduction

Most Conversational Information Retrieval models are a two-step pipeline: Contextual Query Rewriting and ad-hoc Information Retrieval (IR). Most prior works rely on a Historical Query Expansion step [11], i.e. a query expansion mechanism that takes into account all past queries and their associated answers. Such query expansion model is learned on the CANARD dataset [2], which is composed of a series of questions and their associated answers, together with a disambiguated query, referred to as *gold query* in this paper. However, relying on a reformulation step is computationally costly and might be sub-optimal as underlined in [7,8]. Krasakis et al. [7] proposed to use ColBERT [6] in a zero-shot manner, replacing the query by the sequence of queries, without any training of the model. Lin et al. [8] proposed to learn a dense *contextualized* representation of the query history, optimizing a learning-to-rank loss over a dataset composed of weak labels. This makes the training process complex (labels are not reliable) and long.

We propose a much lighter training process for the first-stage ranker, where we focus on queries and do not make use of any passage – and thus of a learning-to-rank training. It moreover sidesteps the problem of having to derive weak labels from the CANARD dataset. Shortly, we require that the representation of the query matches that of the disambiguated query (i.e. the *gold query*). We then train a second-stage ranker (i.e. re-ranker). Leveraging the fact that our first-stage ranker outputs weights over the (BERT) vocabulary, we propose a simple mechanism that provides a conversational context to the re-ranker in the form of keywords selected by SPLADE.

1.1 First stage

The original SPLADE model [3] scores a document using the dot product between the sparse representation of a document ($\hat{d}$) and of a query ($\hat{q}$):

$$s(\hat{q}, \hat{d}) = \hat{q} \cdot \hat{d} \tag{1}$$

The document embedding $\hat{d}$ is obtained using the pre-trained SPLADE model, i.e. $\hat{d} = SPLADE(\texttt{[CLS]}\ d;\ \theta_{SPLADE})$ where θ_{SPLADE} are the original SPLADE parameters obtained from HuggingFace[5]. We can use standard indices built from

[5] The weights can be found at https://huggingface.co/naver/splade-cocondenser-ensembledistil

the original SPLADE document representations to retrieve efficiently the top-k documents.

Query representation We use a simple model where the contextual query representation at turn n, denoted by $\hat{q}_{n,k}$, is the combination of two representations, $\hat{q}_n^{queries}$ which encodes the current query in the context of all the previous queries, and $\hat{q}_{n,k}^{answers}$ which encodes the current query in the context of the past k answers. Formally, the contextualized query representation $\hat{q}_{n,k}$ is:

$$\hat{q}_{n,k} = \hat{q}_n^{queries} + \hat{q}_{n,k}^{answers} \tag{2}$$

where we use two versions of SPLADE parameterized by $\theta_{queries}$ for the full query history and $\theta_{answers,k}$ for the answers. These parameters are learned by optimizing the loss defined in Eq. (8).

Following [8], we define $\hat{q}_n^{queries}$ to be the query representation produced by encoding the concatenation of the current query and all the previous ones:

$$\hat{q}_n^{queries} = SPLADE(\texttt{[CLS]}\ q_n\ \texttt{[SEP]}\ q_1\ \texttt{[SEP]}\ \ldots\ \texttt{[SEP]}\ q_{n-1}; \theta_{queries}) \tag{3}$$

using a set of specific parameters $\theta_{queries}$.

Following prior work [1], we can consider a various number of answers k, and in particular, we choose $k = 1$ (the last answer). Formally, the representation $\hat{q}_{n,k}^{answers}$ is computed as:

$$\hat{q}_{n,k}^{answers} = \frac{1}{k} \sum_{i=n-k}^{n-1} SPLADE(q_n\ \texttt{[SEP]}\ a_i; \theta_{queries,k}) \tag{4}$$

Training Based on the above, training aims at obtaining a good representation $\hat{q}_n$ for the last issued query q_n, i.e. to contextualize q_n using the previous queries and answers. To do so, we can leverage the gold query q_n^*, that is, a (hopefully) contextualized and unambiguous query. We can compute the representation $\hat{q}_n^*$

of this query by using the original SPLADE model, i.e.

$$\hat{q}_n^* = SPLADE(q_n^*; \theta_{SPLADE}) \tag{5}$$

We propose a modified MSE loss, whose first component is the standard MSE loss:

$$Loss_{MSE}(\hat{q}_{n,k}, \hat{q}_n^*) = MSE(\hat{q}_{n,k}, \hat{q}_n^*) \tag{6}$$

In our experiments, we observed that models trained with the direct MSE do not capture well words from the context, especially for words from the answers. We thus added an asymmetric MSE, designed to encourage term expansion from past answers, but avoid introducing noise by restricting the terms to those present in the gold query q_n^*. Formally, our asymmetric loss is:

$$Loss_{asym}(\hat{q}_{n,k}^{answers}, \hat{q}_n^*) = \left(\max(\hat{q}_n^* - \hat{q}_{n,k}^{answers}, 0)\right)^2 \tag{7}$$

where the maximum is component-wise. This loss thus pushes the answer-biased representation $\hat{q}_{n,k}^{answers}$ to include tokens from the gold answer. Contrarily to MSE, it does not impose (directly) an upper bound on the components of the $\hat{q}_{n,k}^{answers}$ representation – this is done indirectly through the final loss function described below.

The final loss we optimize is a simple linear combination of the losses defined above, and only relies on computing two query representations:

$$Loss(\hat{q}_{n,k}, \hat{q}_n^*) = Loss_{MSE}(\hat{q}_{n,k}, \hat{q}_n^*) + Loss_{asym}(\hat{q}_{n,k}^{answers}, \hat{q}_n^*) \tag{8}$$

Implementation details. For the first-stage, we initialize both encoders (one encoding the queries, and the other encoding the previous answer) with pre-trained weights from SPLADE model for adhoc retrieval. We use the ADAM optimizer with train batch size 16, learning rate 2e-5 for the first encoder and 3e-5 for the second. We fine-tune for only 1 epoch over the CANARD dataset.

1.2 Reranking

We perform reranking using a T5Mono [9] approach, where we enrich the raw query q_n with keywords identified by the first-stage ranker. The enriched query q_n^+ for conversational turn n is as follows:

$$q_n^+ = q_n.\ Context : q_1\ q_2\ \ldots\ q_{n-1}.\ Keywords : w_1, w_2, ..., w_K \tag{9}$$

where the w_i are the top-K most important words that we select by leveraging the first-stage ranker as follows. First, to reduce noise, we only consider words that appear either in any query q_i or in the associated answers a_i (for $i \leq n-1$). Second, we order words by using the maximum SPLADE weight over tokens that compose the word.[6] In this work, we choose $K = 10$.

We denote the T5 model fine-tuned for this input as $T5^+$. As in the original paper [9], the relevance score of a document d for the query q_n is the probability of generating the token "`true`" given a prompt $\mathrm{pt}(q_n^+, d)$ = "`Query:` q_n^+`. Document:` d`. Relevant:`":

$$score(q_n^+, d; \theta) = \frac{p_{T5}(\texttt{true}|\mathrm{pt}(q_n^+, d); \theta)}{p_{T5}(\texttt{true}|\mathrm{pt}(q_n^+, d); \theta) + p_{T5}(\texttt{false}|\mathrm{pt}(q_n^+, d); \theta)} \tag{10}$$

where θ are the parameters of the T5Mono model.

Differently to the first stage training, we fine-tune the ranker by aligning the scores of the documents, and not the weight of a query (which is obviously not possible with the T5 model). Here the "gold" score of a document is computed using the original T5Mono with the gold query q_n^*. The T5 model is initialized with weights made public by the original authors[7], denoted as θ_{T5}.

More precisely, we finetune the pre-trained T5Mono model using the MSE-Margin loss [4]. The loss function for the re-ranker (at conversation turn n, given

[6] To improve coherence, we chose to make keywords follow their order of appearance in the context, but did not vary this experimental setting.

[7] We used the Huggingface checkpoint https://huggingface.co/castorini/monot5-base-msmarco

documents d_1 and d_2) is calculated as follows:

$$\mathcal{L}_R = \left[\left(s(q_n^+, d_1; \theta_{T5+}) - s(q_n^+, d_2; \theta_{T5+})\right) - \left(s(q_n^*, d_1; \theta_{T5}) - s(q_n^*, d_2; \theta_{T5})\right)\right]^2 \tag{11}$$

We optimize the θ_{T5+} parameters by keeping the original θ_{T5} to evaluate the score of gold queries.

We also experimented with a simple MSE Loss.

$$\mathcal{L}_R = \left[s(q_n^+, d; \theta_{T5+}) - s(q_n^*, d; \theta_{T5})\right]^2 \tag{12}$$

Implementation details We initialize θ_{T5+} as θ_{T5}, and fine-tune for 3 epochs, with a batch size of 8 and a learning rate 1e-4. We sample pairs (d_1, d_2) using the first-stage top-1000 documents: d_1 is sampled among the top-3, and d_2 among the remaining 997 to push the model to focus on important differences in scores.

2 Data

To train our model, we used the TREC CAsT 2020 and TREC CAsT 2021 dataset, and the CANARD conversational dataset.

As answers from CANARD are short (short sentences extracted from Wikipedia – contrarily to CAsT ones), we expand them to reduce the discrepancy between training and inference. For each sentence, we find the Wikipedia passage it appears in (if it exists in ORConvQA [10]), and sample a short snippet of 3 adjacent sentences from it.

3 Submissions

We submitted 4 runs, listed below. The first one focuses on the first stage-ranker, to evaluate its full ranking performance, while the three next ones use a second-stage ranker for optimal performance (starting with a MSE loss up to an ensemble of re-rankers).

MLIA_DAC_splade First stage ranker only: we rank passages by using our conversational SPLADE model, i.e. after training the query representation is given by Eq. (2) with the asymmetric loss (Eq. 8). This allows to evaluate our main component, i.e. the first stage ranker.

splade_t5mse As MLIA_DAC_splade, but using a re-ranker (as described in section 1.2) trained with the MSE Loss (Eq. 12).

splade_t5mm As MLIA_DAC_splade, but using a re-ranker (as described in section 1.2) trained with the MSE-Margin loss (Eq. 11). This loss is supposed to increase the robustness of the re-ranker [5] (but was designed for first-stage rankers).

splade_t5mm_ens As *splade_t5mm* but with an ensemble of 3 T5-Reranker for the second stage (the difference in training is due to the different sampled triplets).

4 Results

We present our results at cut off 1000, in comparison with the organizer's baseline "BM25_T5_BART_automatic".

Run	Recall	MAP	MRR	nDCG	nDCG@3
BM25_T5_BART_automatic	0.3244	0.1498	0.5272	0.2987	0.3619
MLIA_DAC_splade	0.6384	0.1619	0.5143	0.4327	0.3482
splade_t5mse	0.6384	0.1270	0.4101	0.3933	0.2711
splade_t5mm	0.6384	0.2018	0.5742	0.4704	0.4005
splade_t5mm_ens	0.6384	0.2193	0.5923	0.4832	0.4159

Table 1. Results

We see that the first stage ranker performed well, approaching the rerankers' performances. The simple MSE Loss actually deteriorates performance compared to the first stage ranker, justifying the usage of MSEMargin Loss.

5 Acknowledgements

We would like to thank ANR for supporting this work under the following grants: ANR JCJC SESAMS (ANR-18-CE23-0001) and ANR COST (ANR-18-CE23-0016-003).

References

1. Arabzadeh, N., Clarke, C.L.A.: Waterlooclarke at the trec 2020 conversational assistant track (2020)
2. Elgohary, A., Peskov, D., Boyd-Graber, J.: Can You Unpack That? Learning to Rewrite Questions-in-Context. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). pp. 5918–5924. Association for Computational Linguistics, Hong Kong, China (Nov 2019). https://doi.org/10.18653/v1/D19-1605, https://aclanthology.org/D19-1605
3. Formal, T., Lassance, C., Piwowarski, B., Clinchant, S.: From Distillation to Hard Negative Sampling: Making Sparse Neural IR Models More Effective. In: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 2353–2359. SIGIR '22, Association for Computing Machinery, New York, NY, USA (Jul 2022). https://doi.org/10.1145/3477495.3531857, http://doi.org/10.1145/3477495.3531857
4. Hofstätter, S., Althammer, S., Schröder, M., Sertkan, M., Hanbury, A.: Improving efficient neural ranking models with cross-architecture knowledge distillation. ArXiv **abs/2010.02666** (2020)
5. Hofstätter, S., Althammer, S., Schröder, M., Sertkan, M., Hanbury, A.: Improving efficient neural ranking models with cross-architecture knowledge distillation http://arxiv.org/abs/2010.02666
6. Khattab, O., Zaharia, M.: ColBERT: Efficient and effective passage search via contextualized late interaction over BERT http://arxiv.org/abs/2004.12832
7. Krasakis, A.M., Yates, A., Kanoulas, E.: Zero-shot Query Contextualization for Conversational Search. In: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 1880–1884. SIGIR '22, Association for Computing Machinery, New York, NY, USA (Jul

2022). https://doi.org/10.1145/3477495.3531769, https://doi.org/10.1145/3477495.3531769

8. Lin, S.C., Yang, J.H., Lin, J.: Contextualized query embeddings for conversational search http://arxiv.org/abs/2104.08707
9. Nogueira, R., Jiang, Z., Pradeep, R., Lin, J.: Document ranking with a pretrained sequence-to-sequence model. In: Findings of the Association for Computational Linguistics: EMNLP 2020. pp. 708–718. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.findings-emnlp.63, https://www.aclweb.org/anthology/2020.findings-emnlp.63
10. Qu, C., Yang, L., Chen, C., Qiu, M., Croft, W.B., Iyyer, M.: Open-retrieval conversational question answering pp. 539–548. https://doi.org/10.1145/3397271.3401110, http://arxiv.org/abs/2005.11364
11. Zamani, H., Trippas, J.R., Dalton, J., Radlinski, F.: Conversational Information Seeking (Jan 2022). https://doi.org/10.48550/arXiv.2201.08808, http://arxiv.org/abs/2201.08808, arXiv:2201.08808 [cs]