

NOVA at TREC 2020 Conversational Assistance Track

Rafael Ferreira, David Semedo, and Joao Magalhaes

NOVA University of Lisbon, Portugal
{rah.ferreira, df.semedo}@campus.fct.unl.pt, jmag@fct.unl.pt

Abstract. The use of conversational assistants to search for information is becoming more popular among the general public. In particular, in the last few years, the interest in conversational search is increasing, being this a step forward in allowing a more natural interaction with search systems. In this paper, we describe our work and submitted runs to TREC Conversational Assistance Track (CAST) 2020 [4]. This track is mainly focused on Passage Conversational Information Seeking, being the context of the conversation key to retrieve relevant information. Our approach leverages a three-stage architecture composed of: (a) context tracking via query rewriting, (b) retrieval, and (c) re-ranking using a transformer model. The results obtained with this architecture achieved state-of-the-art results when compared to TREC CAST 2019 baselines [5].

Keywords: Conversational Search · Multi-turn Question Answering · Query Rewriting · Information Retrieval · Passage Ranking.

1 Introduction

In CAST 2019 [5], the conversational search task is formally defined by a sequence of natural language conversational turns for a topic T , with queries q . For each conversation turn $T = q_1, \dots, q_i, \dots, q_n$, the task is to find relevant passages p_k for each query q_i , satisfying the user’s information need for that turn according to the conversational context.

This task presents challenges that typical information retrieval systems do not have to address. One of the primary examples is that the current query may not include all the information needed to retrieve the answer that the user is searching for. This is evidenced in table 1 through the use of “one” (explicit coreference) in turn 2, which refers to physician’s assistant, and in turn 4, where the “starting salary” is for the physician’s assistant position, although there is no direct evidence (implicit coreference). Another challenge is to re-rank the passages according to the conversational context, pushing to the top the passages that are more relevant according to the context of the conversation.

In this work, we describe our submission to TREC CAST 2020 [4]. In particular, we use a three-stage architecture composed of (a) context tracking by

Table 1: Example of a conversation retrieved from TREC CAsT 2019 training set.

Turn	Type of Query	Conversational Query
1	Context-Independent	What is a <u>physician’s assistant</u> ?
2	Conversational	What are the educational requirements required to become <u>one</u> ?
	Context-Independent	What are the educational requirements required to become a <u>physician’s assistant</u> ?
3	Conversational	What does <u>it</u> cost?
	Context-Independent	What does <u>becoming a physician’s assistant</u> cost?
4	Conversational	What’s the average starting salary in the UK?
	Context-Independent	What’s the average starting salary in the UK <u>for a physician’s assistant</u> ?

means of a query rewriting transformer model, (b) retrieval using typical information retrieval techniques, and (c) re-ranking via another transformer model. This architecture achieved state-of-the-art results in TREC CAsT 2019 when compared to the submitted runs [5].

The rest of the paper is organized as follows: next we discuss the developed architecture, in section 3, we discuss the evaluation on the 2019 dataset. Section 4 specifies the results of the delivered runs to TREC CAsT 2020, and concluding remarks are presented in section 5.

2 Method

Our complete architecture is composed by three major components that we will introduce in this section: (a) query rewriting, to track the conversational context and retrieve relevant information, (b) retrieval, to efficiently retrieve passages from the millions present in the index, and (c) re-ranking, to obtain a better rank than the one given by the retrieval stage, by applying a more complex model, trained on large amounts of data which has a better language representation of the text in the query and passages.

2.1 Query-Rewriting

To perform query rewriting, the source of the context are the previous (historical) queries and answers. These will allow us to rewrite the conversational queries to form context-independent queries.

Incorporation of previous queries. Concatenating previous queries is a simple way of adding context but adds irrelevant terms if the current query is not conversational, or if a topic shift occurred. We propose two simple methods:

- **Prefixing (Pref)** - Prefixes the first query of the conversation to the current query. We use the first query since it is not conversational, and it usually sets the topic for the rest of the conversation.

- **Full-Union** - This method performs the union of the current query with all of the previous queries, creating a longer query, based on the number of turns in the conversation. This is a baseline approach that aims to show that choosing the right context is more important than having all of the context in the query.

Neural Coreference Resolution Model. Many conversational queries have mentions to other entities referenced in previous turns (coreferences). To perform the search in the index, we need to resolve these coreferences to retrieve relevant information. To do this, we used AllenNLP’s coreference resolution model [9]. We use as input the sequence of queries in the conversation and use the output of the model for the last query as the resolved query. With this model, we developed two approaches:

- **Coref** - All identified mentions are replaced by the first one. After analyzing some of the outputs, we saw that the model mistakenly replaces mentions that are not coreferent.
- **Coref-Pronoun** - To mitigate the problem identified in the previous approach, we only replace the mention if it is needed, i.e., when pronouns are present in the mention.

Conversational Query-Rewriting Transformer Model. Another approach that we developed uses the pre-trained, text-to-text transformer model T5 [14]. This model requires an input sequence and a target given as strings. We followed [10] and fine-tuned a T5 BASE model using the CANARD dataset [8], using as input the concatenation of the current query, a separator token, and the previous turns (query-answer pairs), and as target the query rewritten. The model was fine-tuned according to [10] for 4000 steps, using a maximum input sequence length of 512 tokens, a maximum output sequence length of 64 tokens, a learning rate of 0.0001, and batches of 256 sequences.

During evaluation on TREC CAsT 2019, since historical utterances don’t depend on the responses of the system [5], the input structure uses only the concatenation of historical queries separated by the same separator token. From the analysis of model’s output, we saw that it was capable of performing both implicit and explicit coreference resolution like presented in table 1.

2.2 Indexing and Retrieval

To index and search, we used Anserini [17], and in particular, the Python implementation Pyserini¹. We indexed the dataset removing stop words using Lucene’s default list, and applied the stemming algorithm Kstem². In our preliminary experiments using TREC CAsT’s 2019 dataset, the use of stemming improved retrieval performance by a large margin. Our initial experiments also showed

¹ <https://github.com/castorini/pyserini>

² <http://lexicalresearch.com/kstem-doc.txt>

that LMD (Language Model Dirichlet) was the best performing method when compared to LMJM (Language Model Jelinek-Mercer) and BM25, confirming previous knowledge [18] that stated that LMD performs best for shorter queries, which are common in a conversational search setting. So the developed methods use LMD as the retrieval model.

2.3 Passage Re-ranking

Transformer Re-ranking As re-ranking model, we used the pre-trained neural language model BERT [6]. This model is capable of generating contextual embeddings for a sentence and each of its tokens. This approach allows us to go beyond simple term matching thanks to the model’s understanding of the interactions between the terms in the query and passage, being able to judge more thoroughly if a passage is relevant to a query.

The model used to perform passage re-ranking, was a BERT BASE model with a linear layer on top, fine-tuned on the query-passage binary relevance estimation task on the MS MARCO dataset following [13]. Our implementation used Huggingface’s Transformer Library [15] and the fine-tuned model *nboost*³. This model is used to calculate the probability of a passage being relevant given a query. We do this for the top-100 passages retrieved and order them by the resulting probabilities.

Rank Fusion. Another way of producing a new rank is by combining (fusing) different ranks, where each one can capture different aspects. As fusion algorithm, we used the Reciprocal Rank Fusion (RRF) [3] defined as:

$$RRFscore(d) = \sum_i \frac{1}{k + r_i(p)}, \quad (1)$$

where k is a hyperparameter, for which we used the default value of 60, and r_i is the rank of passage p in list i .

3 Evaluation

3.1 Experimental Setup and Protocol

CANARD dataset [8] - This dataset was created by manually rewriting the queries in QuAC [1] to form non-conversation queries. This amounts to 31.538, 3.418, and 5.571 query-rewrites for the training, validation, and test sets, respectively. We used this dataset to train and evaluate the T5 query-rewriting model.

TREC CAsT 2019 dataset [4] - This dataset was used to evaluate the performance of the conversational search system. In the evaluation set, each passage’s relevance to each query was graded using a value that ranges from 0

³ <https://huggingface.co/nboost/pt-bert-base-uncased-msmarco>

(not relevant) to 4 (highly relevant). The passage collection is composed by MS MARCO [11], TREC CAR [7], and WaPo [12] datasets, which creates a complete pool of close to 47 million passages (although in the final assessment WaPo was removed from the judgments due to a deduplication problem).

Protocol. To analyze the performance of the retrieval system, we used the official TREC CAsT 2019 metrics, nDCG@3 (normalized Discounted Cumulative Gain at 3), MAP (Mean Average Precision), and MRR (Mean Reciprocal Rank), as well as, Recall and P@3 (Precision at 3).

3.2 Results

Retrieval. In table 2, we show the results of the various query rewriting techniques developed using LMD as the retrieval model.

The first thing that becomes evident is the need for query rewriting methods evidenced by the low scores in the original conversational queries. All the query rewriting methods had the desired effect, increasing all metrics by a considerable margin, approximating the results to the ones obtained using the manually rewritten queries (non-conversational queries). Pref is a simple approach but proved to be useful, even getting better results than Coref and Coref-Pronoun because the model in these last two is not able to detect all of the coreferences. The combination Pref+Coref and Pref+Coref-Pronoun, further improved the results by combining both coreference resolution and concatenation of previous queries. The technique that uses the union of all previous queries and Coref-Pronoun was one of the worst-performing methods because of the long queries with irrelevant terms, showing the importance of choosing the correct context for each turn. With these models, the best query rewriting method was the T5 model achieving the best results in the metrics that evaluate the earlier positions of the rank, which are the most useful for this task.

After obtaining these results in the re-ranking step, we opted to use only the best-performing methods, so we used the Pref+Coref-Pronoun and T5 methods.

Re-ranking. In table 3, we show the results on the TREC CAsT 2019 evaluation dataset with the best query rewriting methods discovered in the previous experiment, in conjunction with the BERT BASE model re-ranker on the top-100 passages retrieved. Adding to this, we also present some baselines from TREC CAsT 2019 [5]. In particular, *clacBase* [2] is a method that uses AllenNLP coreference resolution [9] and a fine-tuned BM25 model with pseudo-relevance feedback, and *HistoricalQE* [16] is a method that uses a query expansion algorithm based on session and query words together with a BERT LARGE model for re-ranking. The latter was the best performing method in terms of nDCG@3 in TREC CAsT 2019 [5].

The results show the effectiveness of the re-ranker achieving an improvement over the simple retrieval method in all metrics. This is due to the better understanding that the fine-tuned BERT model has of the interactions between the query and passage terms. With the architecture that uses T5 for query-rewriting

Table 2: Results of retrieval on the TREC CAsT 2019 evaluation set. All implemented methods use LMD as the retrieval model.

Queries	Recall	P@3	MAP	MRR	nDCG@3
Original	0.454	0.262	0.141	0.336	0.167
Pref	0.667	0.432	0.227	0.547	0.284
Coref	0.573	0.360	0.179	0.445	0.238
Coref-Pronoun	0.619	0.380	0.203	0.486	0.258
Pref+Coref	0.670	0.420	0.218	0.540	0.281
Pref+Coref-Pronoun	0.715	0.462	0.246	0.571	0.304
Coref-Pronoun+Full-Union	0.623	0.389	0.178	0.528	0.255
T5	0.697	0.474	0.251	0.597	0.322
Manual Baselines					
Manual	0.820	0.590	0.327	0.694	0.406

Table 3: Results of retrieval on the TREC CAsT 2019 evaluation set. All implemented methods use LMD as the retrieval model.

Queries	Re-ranker	Recall	P@3	MAP	MRR	nDCG@3
Original	-	0.454	0.262	0.141	0.336	0.167
Original	BERT	0.454	0.382	0.167	0.463	0.276
Pref+Coref-Pronoun	-	0.715	0.462	0.246	0.571	0.304
Pref+Coref-Pronoun	BERT	0.715	0.565	0.282	0.692	0.418
T5	-	0.697	0.474	0.251	0.597	0.322
T5	BERT	0.697	0.611	0.297	0.724	0.461
TREC CAsT 2019 baselines						
clacBase [2]	-	-	-	0.246	0.640	0.360
HistoricalQE [16]	BERT	-	-	0.267	0.715	0.436
Manual Baselines						
Manual	-	0.820	0.590	0.327	0.694	0.406
Manual	BERT	0.820	0.726	0.372	0.874	0.569

and BERT for re-ranking, we were able to achieve state-of-the-art results. We attribute this to the better query-rewriting method that allows the retrieval model to retrieve passages given the conversational context, providing the re-ranker with more relevant passages.

In figure 1, we show the results of the Original, Manual, and T5 queries with BERT re-ranking in the top-100 in each turn until turn depth 8. As expected, the performance in the original queries drops significantly after the first turn because of the lack of conversational context. In the Manual queries, although we see differences in performance in each turn, these are not directly affected by turn depth. With respect to the T5 queries, we observe a decrease in performance in turn 2 because of the inclusion of conversational elements (coreferences and mentions to previous context), after this, performance increases in turn 3, and in most metric proceeds to decrease the deeper we go into the conversation. This is

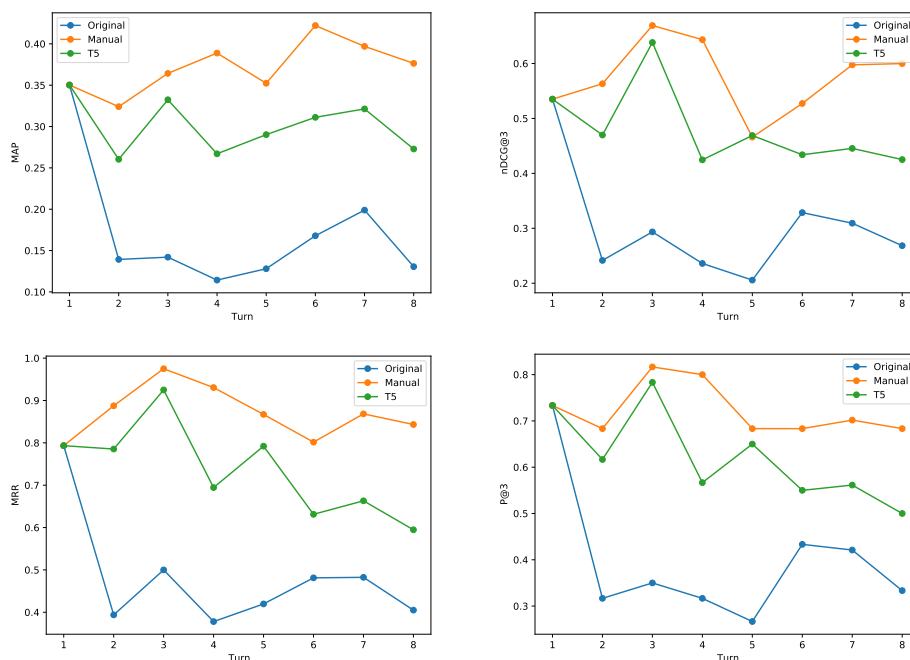


Fig. 1: Results in TREC CAsT 2019 evaluation set by turn depth of Original, Manual, and T5 queries, using LMD as the retrieval model and BERT BASE re-ranking in the top-100.

the expected behaviour and one of the main challenges of conversational search, keeping track of the context in long conversations.

4 Submitted Runs

In TREC CAsT 2020 [4], the task is similar to our experimental setup but with the added challenge that each query in a conversation can be about any of the previous queries or answers (in 2019 only previous queries were considered). Also provided in this year’s edition besides the raw and manually rewritten queries are automatic queries (AUTO). These automatic queries were developed by the organizers [4] using a query rewriting method comparable to ours.

With all these elements and the results obtained in section 3, the submitted runs were the following:

- **AUTO BERT-100** - This run uses the automatic rewritten queries provided by the organizers in conjunction with the fine-tuned BERT BASE model to re-rank the top-100 passages retrieved by LMD.
- **T5 BERT-100** - Uses the queries generated by our fine-tuned T5 model in conjunction with the fine-tuned BERT BASE model to re-rank the top-100 passages retrieved by LMD.

Table 4: Performance of submitted runs on TREC CAsT 2020 evaluation set.

Run	nDCG@3	nDCG@5	nDCG@1000	MAP@1000
Median	0.280	0.274	0.375	0.180
AUTO BERT-100	0.304	0.296	0.364	0.182
T5 BERT-100	0.301	0.298	0.353	0.177
AUTO-T5 BERT-100	0.302	0.296	0.377	0.185

- **AUTO-T5 BERT-100** - Performs retrieval with both the automatic rewritten queries and T5 generated queries and re-ranks the results of both lists with the fine-tuned BERT BASE model on the top-100 passages. We then join these lists using the Reciprocal Rank Fusion (RRF) algorithm.

With the first two runs, we want to make a comparison between our query rewriting model, T5, and the query rewriting model developed by the track organizers. With the last run, we want to evaluate if it is possible to use a combination of both queries to achieve better results.

4.1 Submitted Runs Results

Overall performance. Table 4 shows the results of the official metrics obtained in TREC CAsT 2020 with the submitted runs, as well as the median. Our submitted runs are superior to the median except nDCG@1000 in AUTO BERT-100 and T5 BERT-100, and in MAP@1000 in T5 BERT-100. As we can see, comparing both AUTO BERT-100, which uses query-rewrites made available by the organizers using another model, and T5-BERT 100, that uses our developed method, we obtain similar results. This shows the competitiveness of the query rewriting component developed. Our last run, which combines the two query-rewriting models using RRF, also achieved similar results to previous runs but achieved a higher value for metrics that evaluate the entire list of returned passages (@1000 metrics). This is due to the algorithm’s ability to combine the scores of a passage in both lists in an effective manner.

“Loosely” comparing (the queries are not the same) the results in 2019 (table 3) and 2020 (table 4) editions of TREC CAsT in terms of nDCG@3, we see a large decrease in metrics. This demonstrates that the 2020 dataset is more complex and challenging, mainly due to the inclusion of queries that depend on previous system responses (retrieved passages), which our model didn’t specifically handle.

Performance per turn. Figure 2 shows the performance until turn depth 8 of the submitted runs and median of all runs. As it happened in last year’s data, performance drops significantly in the second turn due to the conversational aspect. Performance suffers an unexpected increase in our submitted runs in the sixth turn and proceeds to decrease until the last turn. Our results are superior to

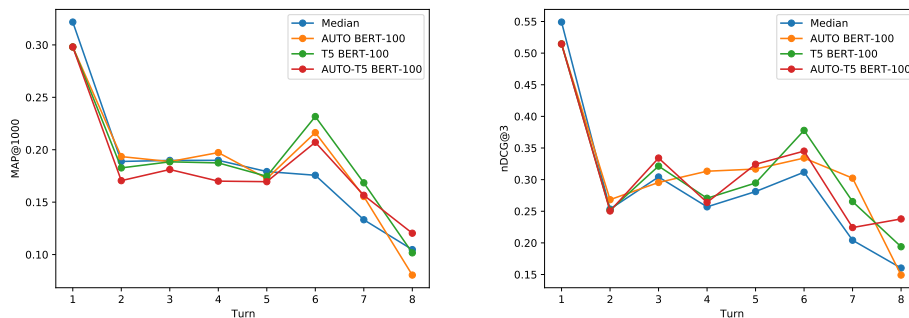


Fig. 2: Results by turn depth of the submitted runs to TREC CAsT 2020.

the median in most turns, with the exception of the first turn, where there is no conversational context. By analyzing these results, we are able to see that having correct rewritten queries is imperative to achieve good performance, being this even more evident in the 2020 edition.

5 Conclusions

In this work, we presented a conversational search architecture composed by a three-stage pipeline. We demonstrated the need for a conversational query rewriting component and achieved good results using a T5 model fine-tuned on this task. This model showed good capabilities for coreference resolution while also having the ability to provide context to a query when this is not explicit. We also showed that a BERT-based re-ranking model can be used to further improve the results of conversational information retrieval systems. To summarize, the full architecture composed by the T5 query re-writing model, LMD retrieval model, and BERT re-ranker achieved state-of-the-art results in the TREC CAsT 2019 dataset [4].

The delivered runs for TREC CAsT 2020 showed that the 2020 dataset is more challenging than the previous year and that our query rewriting model is on par with the model developed by the track organizers. The results also showed the need for a model capable of using previous system answers to rewrite the current query.

As future work, we aim to improve the query rewriting component by adding the previous answers in a way that does not compromise the integrity of the output, such as selecting only previous relevant terms. Concerning the passage ranking model, we want to explore models that can attend to the full conversational context instead of only considering the current query and list of retrieved passages for that turn.

References

1. Choi, E., He, H., Iyyer, M., Yatskar, M., Yih, W., Choi, Y., Liang, P., Zettlemoyer, L.: Quac : Question answering in context. CoRR **abs/1808.07036** (2018), <http://arxiv.org/abs/1808.07036>
2. Clarke, C.L.A.: Waterlooclarke at the TREC 2019 conversational assistant track. In: 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/papers/WaterlooClarke.C.pdf>
3. Cormack, G.V., Clarke, C.L.A., Büttcher, S.: Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In: Allan, J., Aslam, J.A., Sanderson, M., Zhai, C., Zobel, J. (eds.) Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2009, Boston, MA, USA, July 19-23, 2009. pp. 758–759. ACM (2009). <https://doi.org/10.1145/1571941.1572114>, <https://doi.org/10.1145/1571941.1572114>
4. Dalton, J., Xiong, C., Callan, J.: The trec conversational assistance track CAsT (1 2020), <http://www.treccast.ai/>
5. Dalton, J., Xiong, C., Callan, J.: TREC cast 2019: The conversational assistance track overview. CoRR **abs/2003.13624** (2020), <https://arxiv.org/abs/2003.13624>
6. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. CoRR **abs/1810.04805** (2018), <http://arxiv.org/abs/1810.04805>
7. Dietz, L., Gamari, B., Dalton, J.: Trec car 2.1: A data set for complex answer retrieval (7 2018), <http://trec-car.cs.unh.edu>
8. Elgohary, A., Peskov, D., Boyd-Graber, J.L.: Can you unpack that? learning to rewrite questions-in-context. In: Inui, K., Jiang, J., Ng, V., Wan, X. (eds.) Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019. pp. 5917–5923. Association for Computational Linguistics (2019). <https://doi.org/10.18653/v1/D19-1605>, <https://doi.org/10.18653/v1/D19-1605>
9. Lee, K., He, L., Lewis, M., Zettlemoyer, L.: End-to-end neural coreference resolution. CoRR **abs/1707.07045** (2017), <http://arxiv.org/abs/1707.07045>
10. Lin, S., Yang, J., Nogueira, R., Tsai, M., Wang, C., Lin, J.: Conversational question reformulation via sequence-to-sequence architectures and pretrained language models. CoRR **abs/2004.01909** (2020), <https://arxiv.org/abs/2004.01909>
11. Nguyen, T., Rosenberg, M., Song, X., Gao, J., Tiwary, S., Majumder, R., Deng, L.: MS MARCO: A human generated machine reading comprehension dataset. CoRR **abs/1611.09268** (2016), <http://arxiv.org/abs/1611.09268>
12. NIST: Trec washington post corpus (12 2019), <https://trec.nist.gov/data/wapost/>
13. Nogueira, R., Cho, K.: Passage re-ranking with BERT. CoRR **abs/1901.04085** (2019), <http://arxiv.org/abs/1901.04085>
14. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the limits of transfer learning with a unified text-to-text transformer. CoRR **abs/1910.10683** (2019), <http://arxiv.org/abs/1910.10683>
15. Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Brew, J.: Huggingface’s transformers: State-of-

- the-art natural language processing. CoRR **abs/1910.03771** (2019), <http://arxiv.org/abs/1910.03771>
16. Yang, J., Lin, S., Wang, C., Lin, J., Tsai, M.: Query and answer expansion from conversation history. In: 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/papers/CFDA_CLIP.C.pdf
 17. Yang, P., Fang, H., Lin, J.: Anserini: Enabling the use of lucene for information retrieval research. In: Kando, N., Sakai, T., Joho, H., Li, H., de Vries, A.P., White, R.W. (eds.) Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Shinjuku, Tokyo, Japan, August 7-11, 2017. pp. 1253–1256. ACM (2017). <https://doi.org/10.1145/3077136.3080721>, <https://doi.org/10.1145/3077136.3080721>
 18. Zhai, C., Lafferty, J.: A study of smoothing methods for language models applied to ad hoc information retrieval. In: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. p. 334–342. SIGIR '01, Association for Computing Machinery, New York, NY, USA (2001). <https://doi.org/10.1145/383952.384019>, <https://doi.org/10.1145/383952.384019>