Question Answering-Based Query Expansion for Conversational Search: IIIA@UNIPD at TREC CAsT 2022

Guglielmo Faggioli[1][0000−0002−5070−2049], Nicola Ferro[1][0000−0001−9219−6239], and Mattia Romanello[1]

University of Padova, Via Gradenigo 6/b, 35138 Italy

Abstract. Conversational Search task is becoming ubiquitous in our daily interaction with information systems. Nevertheless, it remains an extremely complex task due to its profoundly different nature from ad-hoc retrieval, and the challenges presented by the way a user interacts with the system. CAsT TREC Track explicitly aims at fostering the development of conversational systems and the discussion of the linked challenges within the research community. In this work, we describe the approach that we propose to CAsT 2022 to tackle the conversational task. In particular, our approach is based on expanding and rewriting the utterance at hand with the response to the previous utterance, using a QA model to extract it from the first passage retrieved in response to the previous query.

1 Introduction

The conversational search task is constantly drawing more and more interest from both the academy and industry. Conversational agents that rely on conversational search modules are increasingly more popular, both in the form of vocal assistants but also as textual chatbots meant to help users to deal with simple tasks. In this context, the CAsT TREC track is meant to foster the development of new conversational search systems, while also stimulating the discussion within the community on the rising challenges linked to this specific task. CAsT is currently at is fourth iteration [1–3] with increasingly growing interest from the community. Differently from previous years, two tasks were considered in this year's CAsT Track:

- Primary task: retrieve the candidate response.
- Mixed initiative task: produce a response to the user's utterance, including clarifying questions, requests for feedback or task elicitation.

In our experiments, we focus only on the primary task. In the following sections, we describe our approach to the CAsT 2022 TREC track which relies on the following steps.

- Query expansion based on the previous utterance and the response to it, obtained via a QA approach;

– Query rewriting based on POS tagging and coreference resolution via pre-trained models;
– First stage retrieval based on classical approaches that exploit the Lucene implementation of BM25 and LMD;
– Re-ranking based on BERT [4].

The work is organized as follows: Section 2 reports the methodology adopted to address the task, Section 3 contains the description of the runs submitted, Section 4 describes the achieved results and, Section 5 draws our conclusion.

2 Methodology

In this section, we describe the methodology adopted to tackle the CAsT 2022 TREC Track. Notice that, our focus is only on the primary task. Figure 1 illustrates the complete pipeline.

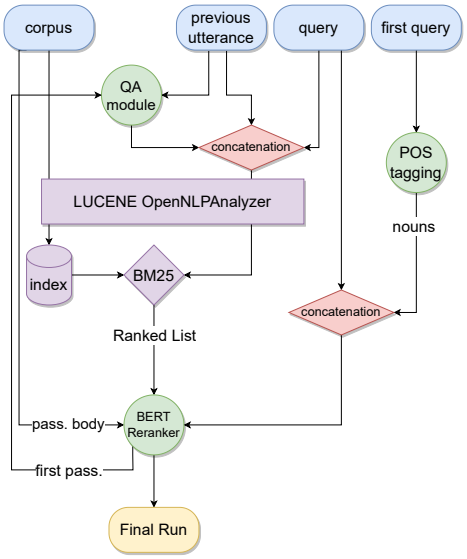

Fig. 1: The complete pipeline

2.1 Basic pipeline

Our pipeline is divided into four main steps:

– query expansion;
– utterance rewriting;
– first stage retrieval;
– reranking.

We describe each of these steps in the remainder of this section.

Query Expansion Being usually self-explanatory, we did not carry out any expansion on the first utterance of the conversation. Therefore, our expansion procedure is applied only starting from the second turn of the conversation. Our pipeline relies on two different languages: Python and Java. More in detail, Java and the Lucene library are used to compute the first stage run, using BM25 [5]. On the other hand, python is used for more machine-learning-oriented passages in our pipeline, such as the BERT-reranking, Question Answering and carrying out coreference resolution with the NeuralCoref Library.

To expand the query, we considered the following signals:

- previous utterance;
- previous response;
- current utterance

The rationale behind our approach is that the previous utterance itself can provide the required context (entities) to answer the current utterance. Similarly, the utterance issued by the user in the current turn might be related to the response provided by the system to the user. Notice that, the concept of response can be either a specific entity that represents the answer to the question of the user, or the entire paragraph returned by the system to satisfy the user's information need. We experimented with both solutions. Our empirical findings highlight that using a short answer (often a single entity) produces better results than using the entire paragraph. We, therefore, apply the `deepset/roberta-base-squad2`[1] pre-trained question-answering model to compute the response to the previous utterance, using the first retrieved paragraph as context.

Query Rewriting Once the response to the previous utterance has been computed, we rewrite the current utterance with the following approach:

1. first of all, the strings are concatenated into a single string: previous utterance + previous response + current utterance.
2. We compute the Part Of Speech (POS) tagging of the string computed previously. Such tagging is used to resolve complex scenarios where the coreferences in the current utterances cannot be resolved automatically.
3. We apply a coreference resolution algorithm to obtain a rewritten string that does not contain anaphoras, coreferences or ellipses.
4. The result is further refined, by taking only the last part of the string, which corresponds to the current utterance.

To carry out the coreference resolution we apply a well-known approach using the pre-trained `NeuralCoref` model[2]. Notice that, NeuralCoref may fail in solving the coreferences, leaving some pronouns in the sentence untouched. In that case, we apply a greedy algorithm to force the resolution of the coreference. In particular, we consider all the tokens annotated as `nouns` by the POS tagging

[1] https://huggingface.co/deepset/roberta-base-squad2
[2] https://github.com/huggingface/neuralcoref

procedure (step 2), and replace the remaining pronouns using the token having the highest weight provided by `NeuraCoref`.

Similarly, we also noticed that, if the query doesn't end with "?", it is often a clarification question related to the previous utterance. Therefore, in the case that the query was not terminated with a question mark, we append the previous utterance to the query.

First Stage Retireval To carry out first stage retrieval, we used the Lucene retrieval library[3] with the default parameters for the similarity functions used. In particular, we considered two similarity functions: BM25 and Language Model with Dirichlet (LMD) smoothing. The settings for BM25 are:

- k2 = 1.2
- b = 0.75

While, in the case of LMD [6], we applied the following hyperparameters:

- $\mu = 200$

Reranking Once documents have been retrieved using the first-stage retrieval function, we re-rank the documents by using the BERT classifier. Notice that we use a pre-trained version of BERT, without additional finetuning. In particular, we use `cross-encoder/ms-marco-TinyBERT-L-2-v2` [4]. Besides the documents retrieved by the first-stage approach, we provide to the classification model a second expanded version of the current utterance. In particular, we add to it the list of nouns identified using the POS tagging approach from the first utterance, which is expected to provide the general context of the conversation. Finally, we re-rank the first 75 documents obtained after the first stage retrieval pass using the BERT classifier. We experimented with different cutoffs (30, 50, 75, 100), achieving the best empirical performance for 75 documents.

2.2 Improved Coreference Resolution

To allow NeuralCoref to solve pronouns coreferences properly the query must contain all the necessary information and it must be written in a logical sense. The same utterance written in two different ways causes the results to change dramatically. Let's consider the topic 106 of TREC CAsT 2021 as reported in Table 1. Considering utterance 3, the pronoun "it" can refer to the response of the previous utterance or the main argument of the previous utterance. Suppose that NeuralCoref replaces the pronoun "it" with "breast cancer" (the topic of the previous utterance). If the pronoun "it" is replaced with "Lobular Carcinoma" (response to the previous utterance) the result will be different, as it refers to two quite different contexts. To avoid this problem (if the algorithm fails the substitution: the resulting query is wrong) and increase the performance, we used the `automatic_resolved_utterance` provided

[3] https://lucene.apache.org/
[4] https://huggingface.co/cross-encoder/ms-marco-TinyBERT-L-2-v2

Table 1: topic 106 from CAsT 2022

Turn	Utterance
1	I just had a breast biopsy for cancer. What are the most common types?
2	Once it breaks out, how likely is it to spread?
3	How deadly is it?
	...

by TREC CaST organizers. We use a combination between NeuralCoref and the `automatic_resolved_utterance` Therefore, if the utterance and its rewritten version were the same, instead of applying the POS-based approach, we resort to using directly the `automatic_resolved_utterance`.

2.3 Improved Query Expansion

One of the biggest problems of our approach, when applied to the CAsT 2021 dataset, is related to the high number of utterances for which no relevant document was retrieved. We noticed that the problem tends to be highly related to the fact that the retrieved documents tend to be relevant to the previous query, and not to the current one. It is therefore likely that our query expansion procedure, based on concatenating the current utterance with the previous one, while increasing the recall, reduced the precision – which is the main focus of the conversational task.

We, therefore, replace the expansion of the query based on the concatenation of the previous utterance, previous response and current utterance, by considering only the current utterance and the response to the previous one, without considering the previous utterance.

3 Submitted Runs

We submitted 4 runs to CAsT 2022 TREC track:

DEI-run1 : this run is the result of the pipeline as described in 2.1. It first expands the query considering the previous utterance and the response to it, using the QA-based approach. It rewrites the run, using the `NeuralCoref` based approach. It uses BM25 as a first-stage retrieval model. It re-ranks the result using the BERT approach.

DEI-run2 : this run replaces the rewriting of the query described in 2.1, with the one proposed in 2.2. Similarly to the previous case, it expands the query in the same way (considering the previous utterance and the answer to it), it uses BM25 to carry out first-stage retrieval and reranks the result with the BERT approach.

DEI-run4 : this run further aims at improving the previous approaches using the improved query expansion approach described in subsection 2.3. It also

adopts the same strategy as described in subsection 2.2 to resolve coreferences. We use BM25 as a first-stage retriever.

DEI-run5 : The final run is equivalent to DEI-run4, but it employs LMD as a first-stage retriever.

4 Results

Table 2 reports the performance achieved by the submitted runs. It is possible to observe, while they have generally uniform performance, DEI-run4 is the best performing one.

Table 2: Results achieved on CAsT 2022 collection by the proposed approaches

	median	**DEI-run1**	**DEI-run2**	**DEI-run4**	**DEI-run5**
ap (lenient)	0.1749	0.1483	0.1505	0.1505	0.1355
ndcg@20 (lenient)	0.3172	0.2758	0.2711	0.2759	0.2505
ap (strict)	0.1479	0.1294	0.1286	0.1281	0.1172
ndcg@20 (strict)	0.3204	0.2758	0.2711	0.2759	0.2505

Figures 2 to 9 report the difference in performance with respect to the median. Notice that, while in general the approach tend to attain low performance, we can observe that almost all approaches deal correctly with topic `137`, while topic `146` is the one on which all versions of the approach fail the most.

5 Conclusions

In this work, we presented an approach to tackle the conversational search task. In particular, we detail our participation in the CAsT 2022 TREC track. The method, while being unable to achieve high performances, can deal with the task with low computational cost and a low degree of complexity. Notice that, given the limited availability of computational resources, we exploited pretrained models for all the tasks based on machine learning approaches. In the future, we plan to continue our work, by fine-tuning the different components of the pipeline.

References

1. Dalton, J., Xiong, C., Callan, J.: Trec cast 2019: The conversational assistance track overview. arXiv preprint arXiv:2003.13624 (2020)
2. Dalton, J., Xiong, C., Callan, J.: Cast 2020: The conversational assistance track overview. Tech. rep., Technical report (2021)
3. Dalton, J., Xiong, C., Callan, J.: Cast 2021: The conversational assistance track overview. Tech. rep., Technical report (2022)

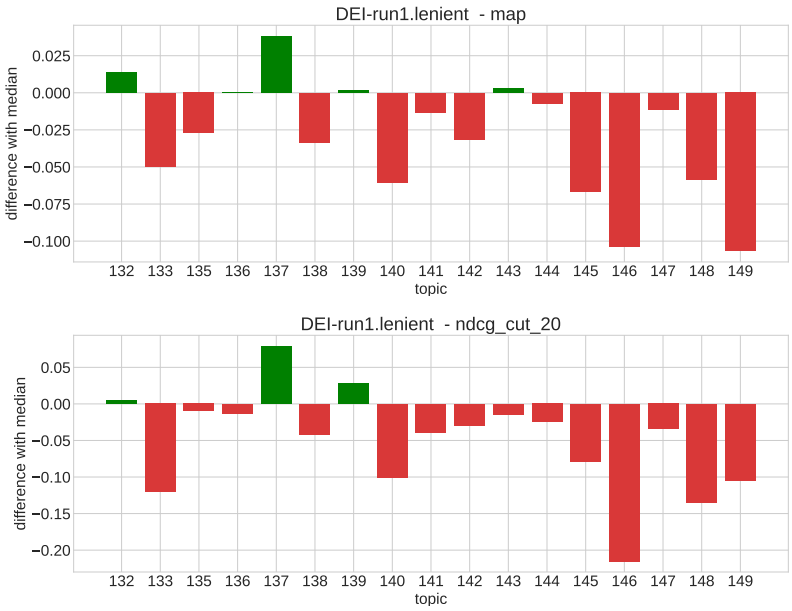

Fig. 2: DEI-run1 performance with lenient evaluation

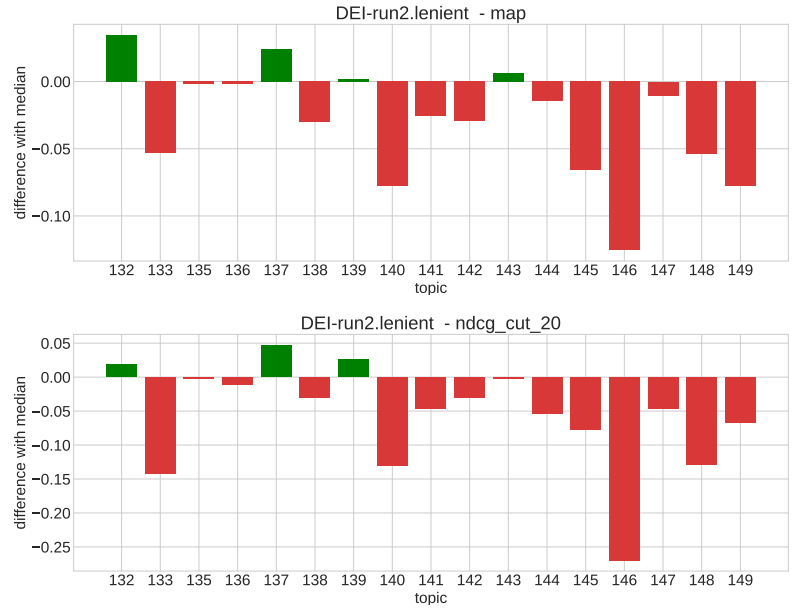

Fig. 3: DEI-run2 performance with lenient evaluation

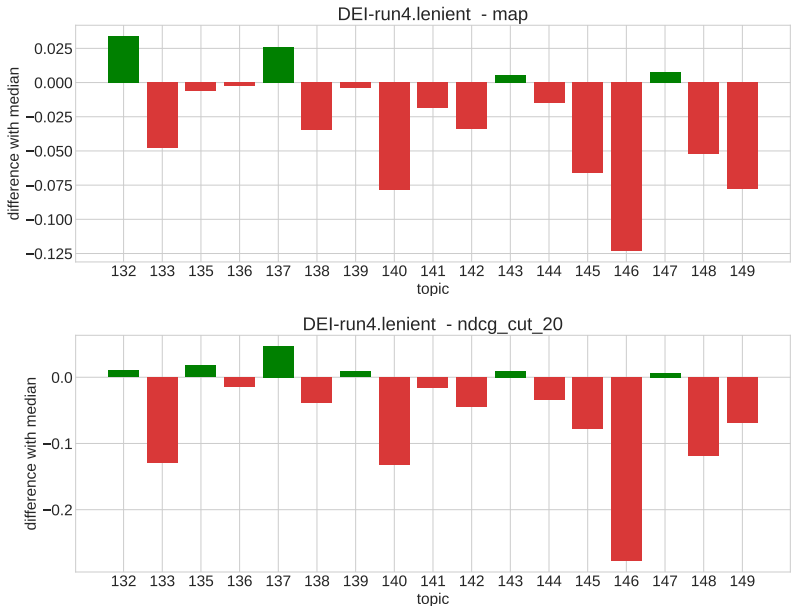

Fig. 4: DEI-run4 performance with lenient evaluation

Fig. 5: DEI-run5 performance with lenient evaluation

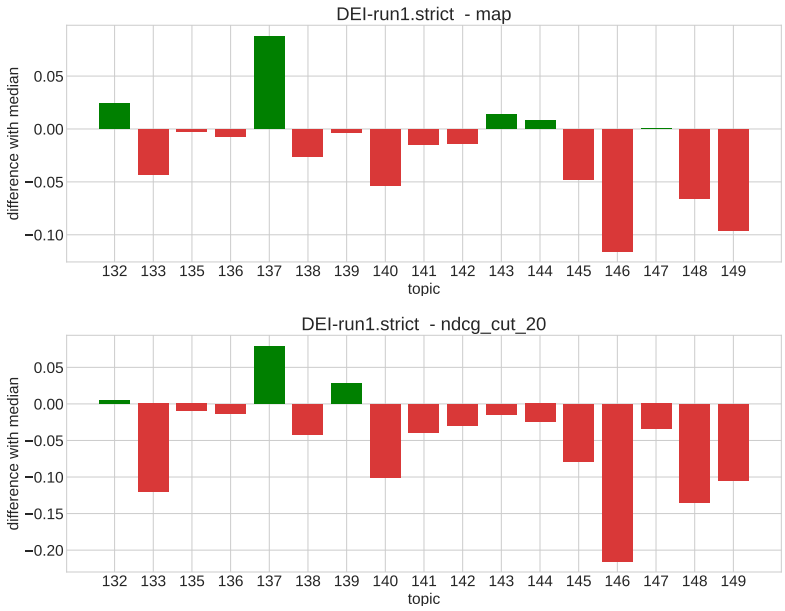

Fig. 6: DEI-run1 performance with strict evaluation

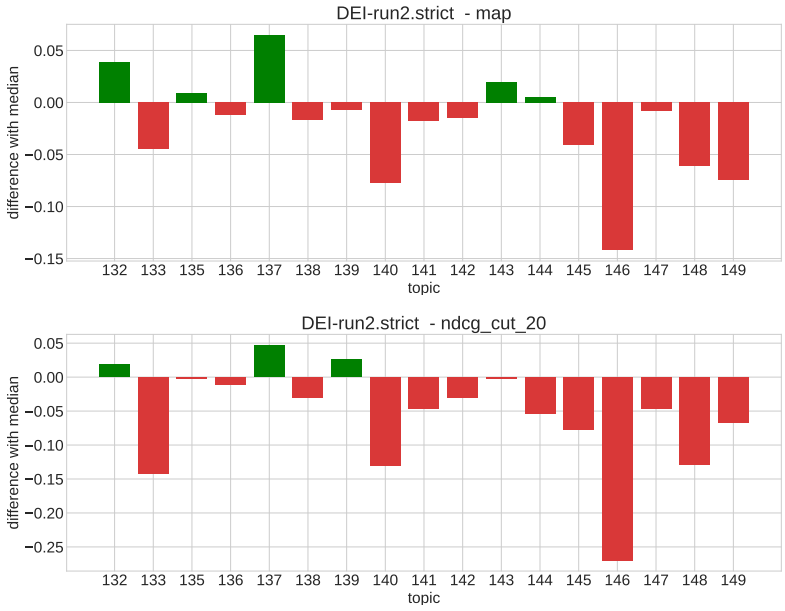

Fig. 7: DEI-run2 performance with strict evaluation

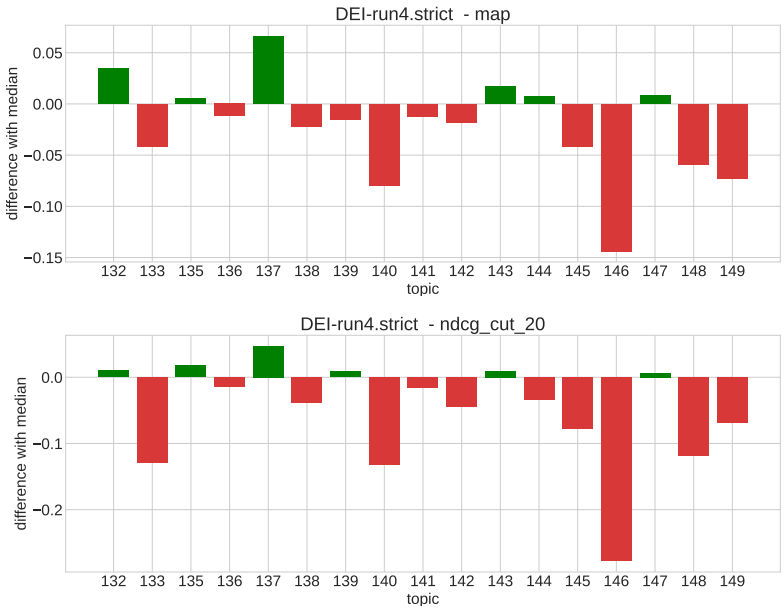

Fig. 8: DEI-run4 performance with strict evaluation

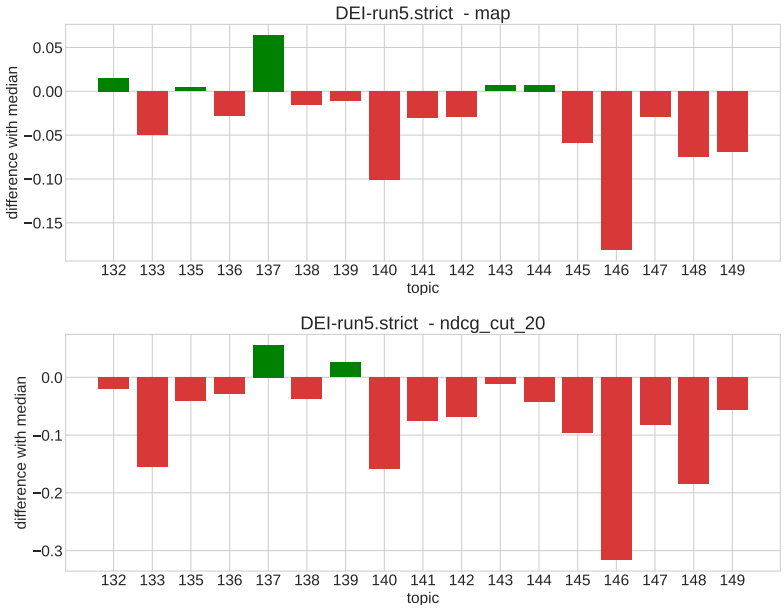

Fig. 9: DEI-run5 performance with strict evaluation

4. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers). pp. 4171–4186 (2019)
5. Robertson, S.E., Zaragoza, H.: The probabilistic relevance framework: BM25 and beyond. Found. Trends Inf. Retr. **3**(4), 333–389 (2009). https://doi.org/10.1561/1500000019, https://doi.org/10.1561/1500000019
6. Zhai, C., Lafferty, J.: A study of smoothing methods for language models applied to ad hoc information retrieval. SIGIR Forum **51**(2), 268–276 (aug 2017). https://doi.org/10.1145/3130348.3130377, https://doi.org/10.1145/3130348.3130377