

# POLYU at TREC 2020 Conversational Assistant Track: Query Reformulation with Heuristic Topic Phrases Discovery

Kaishuai Xu  
The Hong Kong Polytechnic  
University  
Hong Kong, China  
xu.kaishuai@gmail.com

Wenjie Li  
The Hong Kong Polytechnic  
University  
Hong Kong, China  
cswjli@comp.polyu.edu.hk

Yongli Li  
The Hong Kong Polytechnic  
University  
Hong Kong, China  
liyongqi0@gmail.com

## ABSTRACT

This paper demonstrates a 2-stage conversational search architecture for the Conversational Assistant Track in TREC 2020, including the initial rule-based retrieval and BERT-based re-ranking. We propose a straightforward query reformulation method with topic phrases discovery and inheritance. The method can efficiently extract the key phrase as a topic and inherit phrases based on specific rules. Experimental results show that our method performs as well as top-2 teams in CAsT 2019 evaluation datasets (NDCG@3: 0.433) with a simpler query expansion and smaller BERT model.

## KEYWORDS

conversational search, query reformulation, information retrieval

### ACM Reference Format:

Kaishuai Xu, Wenjie Li, and Yongli Li. 2020. POLYU at TREC 2020 Conversational Assistant Track: Query Reformulation with Heuristic Topic Phrases Discovery. In *TREC '20: Text REtrieval Conference, November 18–20, 2020, Gaithersburg, Maryland USA*. ACM, New York, NY, USA, 2 pages. <https://doi.org/10.1145/nnnnnnn.nnnnnnn>

## 1 INTRODUCTION

Conversational Information Seeking (CIS) or Conversational Search (CS) has received burgeoning attention in information retrieval and natural language processing researches. To create a reusable benchmark for open-domain CIS systems, TREC organizes the Conversational Assistance Track (CAsT) based on several decades of multiple turns of dialogues and definitive collection. The primary aim of this year is to find relevant passages using contextual information.

Our work concentrates on solving correlation and omission problems in a straightforward and low-computing method without training on deep learning models. We construct a pipeline architecture, including the initial retrieval and the BERT-based re-ranking. In the first step, we develop a heuristic query expansion algorithm to reformulate the queries with topic phrases discovery and inheritance based on BM25 scores. The ranking model in this step is also BM25. In the second step, a simple BERT-based classifier is trained

on another dataset for transfer learning. The detailed methodology is described in Section 2, followed by the experiments (Section 3) and results (Section 4).

## 2 METHODOLOGY

Following previous reports and recent advances [2], we construct the 2-stage retrieval pipeline. In the initial retrieval, we use BM25 to retrieve top-1000 relevant passages for each query and record the corresponding scores. In the re-ranking, a BERT-based classifier is built on pairs of the query and passage to obtain the classification value as a relevance score. Finally, we sum up scores from two stages with equal weight and get the passages' new ranking list.

### 2.1 Initial retrieval

The initial retrieval is based on a probabilistic retrieval model BM25. For the application of such models, the preprocessing procedures such as stop word removal and query reformulation are crucial to increasing the recall rate. The relevance is calculated through the word occurrence probability in documents and word frequency, so it is required to eliminate the irrelevant words and make the query self-contained.

**2.1.1 Stop word removal.** The stop words in our work are selected in two ways: 1. Most words are summarized by [1] according to its experiment; 2. The others are chosen with relatively low BM25 scores. We supplement another 17 words into the stop words list and get a total of 95 stop words.

**2.1.2 Query expansion.** Inspired by the history query expansion in [4], we develop a heuristic algorithm for topic phrases discovery and inheritance, simpler and easier to use. The proposed method's core idea is similar to [4]. However, instead of distinguishing query keywords from session ones, we exclusively obtain or inherit one key phrase as a topic phrase from the current query or last key phrase. The two reasons for this revision are that BM25 scores alone cannot tell from different types of keywords, and topic words generally appear in phrases. Our method focuses on finding phrases and selecting the key phrase in a heuristic way. Based on observations on the datasets, we assume that the current query is only related to the previous query. Accordingly, if there are coreference, omission, or no key phrase in the current query, the method will inherit the last key phrase and supplement the current query with the preceding query phrases. The exact method is as follows.

The method input is a set of passages  $d_i \in D$ , and a topic  $T$  with  $N$  conversational queries,  $T = \{Q_i\}_{i=1}^N$ . Algorithm 1 describes

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).  
*TREC '20, November 18–20, 2020, Gaithersburg, Maryland USA*

© 2020 Association for Computing Machinery.  
ACM ISBN XXX-X-XXXX-XXXX-X/XX/XX...\$15.00  
<https://doi.org/10.1145/nnnnnnn.nnnnnnn>

the details of topic phrases discovery and inheritance. The function *FindPhrase* is a constituency parsing algorithm that extracts phrases from a query.  $F(\cdot)$  calculates the BM25 scores between phrases and passages.  $Ind(\cdot)$  is a rule-based query independence judge. We judge the existence of the coreference or omission in queries based on three rules: 1. Phrases like “what about”, “how about” are in a query; 2. Determiners are in a query, although the coreference resolution has been carried out; 3. The superlative words appear in the form of “the best”.  $len(\cdot)$  computes the number of tokens in a phrase.  $\lambda$  is the threshold controlling whether the phrase is specific enough to be considered the topic.  $W_{KP}$  is the key phrase list.

---

**Algorithm 1: Topic Phrases Discovery and Inheritance**


---

**Data:**  $T = \{Q_i\}_{i=1}^N, D$   
**Result:**  $T$   
**begin**  
 $W_{KP} \leftarrow \emptyset$   
**for**  $i = 1$  **to**  $N$  **do**  
 $is\_KP = 0$   
 $P_i = FindPhrase(Q_i) = \{p_k^i\}_{k=1}^{n(i)}$   
**for**  $j = 1$  **to**  $n(i)$  **do**  
 $S_j^i = \max_{d_t \in D} F(d_t, p_j^i)$   
**if**  $\frac{S_j^i}{len(p_j^i)} > \lambda$  **then**  
 $is\_KP = 1$   
 $W_{KP} \leftarrow W_{KP} + \{p_j^i\}$   
**if**  $is\_KP = 0$  **or**  $Ind(Q_i) = 0$  **and**  $i > 1$  **then**  
 $Q_i \leftarrow Q_i + p_k^i$  **for all**  $p_k^{i-1} \in P_{i-1} \wedge S_k^{i-1} > \lambda$   
**return**  $T$

---

## 2.2 Re-ranking

We construct a BERT-based classifier with the same architecture in [3]. The [CLS] vector is extracted and put into a fully connected layer to classify the relevance. The classifier is trained on other open-domain question answering datasets.

## 3 EXPERIMENTAL SETUP

The proposed pipeline is carried out in two stages. The first stage is the initial retrieval using the BM25 algorithm. According to previous reports, we choose the parameter range and obtain the best parameters ( $k = 1.2, \beta = 0.25$ ) through further experiments. In pre-processing, coreference resolution and constituency parsing are both implemented using the AllenNLP toolkit<sup>1</sup>. All phrases and queries are tokenized through the spaCy toolkit<sup>2</sup>. We retrieve 1000 passages at the first stage.

In the second stage, we build a classifier based on the BERT-base model using the Keras-bert package. The model is trained on MS MARCO Question Answering datasets. Most parameters are initial default values, but the maximum token number of BERT is

<sup>1</sup><https://allennlp.org/>

<sup>2</sup><https://spacy.io/>

**Table 1: Results on the CASt evaluation**

Models	Datasets	Type	NDCG@3
Our method (defective)*	2019	automatic	0.324
Our method			0.433
h2oloo_RUN2			0.434
CFDA_CLIP_RUN7			0.436
Our method (defective)	2020	automatic	0.265
		manual	0.398

\* the input number of tokens in BERT is mistakenly set to 15

set 192 according to all samples’ length. The queries without the expansion and initial-retrieved passages are put into the model to get the relevance scores. The final ranking is concluded from the weighted sum of initial retrieval scores and classifier scores. We set equal weight for two stages.

## 4 RESULTS AND DISCUSSION

The result is represented in Table 1. Since some bugs are not modified before submission, the CASt 2020 evaluation results are not the proposed method’s best performance. The main issue is that we mistakenly set the number of tokens in the BERT input to 15. After correction to 192, our method has proved effective compared to the last year’s best results.

## ACKNOWLEDGMENTS

The work described in this paper is supported by Research Grants Council of Hong Kong (5210919), National Natural Science Foundation of China (62076212) and PolyU Internal Grant (ZVRH, ZG7H).

## REFERENCES

- [1] Charles LA Clarke. 2019. WaterlooClarke at the TREC 2019 Conversational Assistant Track. In *TREC*.
- [2] Sheng-Chieh Lin, Jheng-Hong Yang, Rodrigo Nogueira, Ming-Feng Tsai, Chuan-Ju Wang, and Jimmy Lin. 2020. Query Reformulation using Query History for Passage Retrieval in Conversational Search. *arXiv preprint arXiv:2005.02230* (2020).
- [3] Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage Re-ranking with BERT. *arXiv preprint arXiv:1901.04085* (2019).
- [4] Jheng-Hong Yang, Sheng-Chieh Lin, Chuan-Ju Wang, Jimmy Lin, and Ming-Feng Tsai. 2019. Query and Answer Expansion from Conversation History. In *TREC*.