Query and Answer Expansion from Conversation History

Jheng-Hong Yang[★] Sheng-Chieh Lin[★] Jimmy Lin[†]
Ming-Feng Tsai[‡] Chuan-Ju Wang[★]
[★] Research Center for Information Technology Innovation, Academia Sinica, Taiwan
[†] David R. Cheriton School of Computer Science, University of Waterloo, Canada
[‡] Department of Computer Science, National Chengchi University, Taiwan

ABSTRACT

In this paper, we present our methods, experimental analysis, and final submissions for the Conversational Assistance Track (CAsT) at TREC 2019. In addition to language understanding, extracting knowledge from historical dialogues (e.g., previous queries, searching results) is a key to the conversational IR task. However, limited annotated data in the CAsT task makes machine learning or other data-driven approaches infeasible. Along this line, we propose two ad hoc and intuitive approaches: Historical Query Expansion and Historical Answer Expansion, to improve the performance of the conversational IR system with limited training data. Our empirical result on the CAsT training set shows that the proposed methods significantly improve the quality of conversational search in terms of retrieval (recall@1000: 0.774 → 0.844) and ranking (mAP: 0.187 → 0.197) compared to our strong baseline. As a result, our submitted entries outperform the median performance of all the 21 teams.

ACM Reference Format:
Jheng-Hong Yang[★] Sheng-Chieh Lin[★] Jimmy Lin[†] and Ming-Feng Tsai[‡] Chuan-Ju Wang[★]. 2019. Query and Answer Expansion from Conversation History. In *TREC '19: Text REtrieval Conference, Nov 13–15, 2019, Gaithersburg, Maryland.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 METHODOLOGY

Following [1], we build a two-stage QA system, document retrieval and reranking. Document retrieval is to narrow down the search space by retrieving top-k candidates, while reranking focuses on computing the relevance scores for all the given k query-passage pairs. To further improve the quality of conversational search, we propose two ad hoc approaches.

1.1 Historical Query Expansion

Unlike other conversational QA tasks (e.g. CoQA [9], QuAC [2]), in which simply adding historical queries and answers to current query can solve co-reference by model training, CAsT, however, does not have enough training data, making these methods ineffective. Thus, we propose to extract key words from historical queries to expand current query for subsequent search tasks.

```
Session 1
Title: Career choice for Nursing and Physician's
Assistant
Turn1: What is a physician's assistant?
Turn2: What are the educational requirements required
to become one?
Turn3: What does it cost?
…
```

Figure 1: CAsT example. Bold and underlying words denote the keywords specifying the topic of the session and query respectively.

Suppose we have a set of documents $d_j \in \mathcal{D}$ and a session $\mathcal{S}$ with $\mathcal{N}$ turns of conversational queries, $\mathcal{S} = \{Q_i\}_{i=1}^{\mathcal{N}}$. Each query Q_i has $n(i)$ tokens, represented as a tuple $(q_0^i, q_1^i, \ldots, q_{n(i)}^i)$. Algorithm 1 details our historical query expansion method: keyword extraction (line 3 − 8) and expansion (line 9 − 13). Here $\mathcal{F}$ represents the BM25 score function between the document d_i and the query Q_i; r_Q ($r_\mathcal{S}$) is the hyperparameter to judge whether a word is a key word related to current query (session, respectively); θ is the hyperparameter to judge whether a query is specific enough; W_Q ($W_\mathcal{S}$) is the query (session, respectively) keyword list extracted from historical queries.

Intuitively, we assume that a precise query includes keywords specifying the topic of the session and the query itself. For example, as shown in Figure 1, the topic of the session is related to "Nursing and Physician's Assistant," which is specified by Turn1. Turn2 and Turn3 only have keywords related to each query and are still ambiguous. Specifically, Turn2 is related to "educational requirements" but not specific enough without the session keywords while Turn3 is more ambiguous and needs to be clarified by adding both "Physician's Assistant" and "education requirement", the session and historical query keywords respectively. This reflects our design (line 9-13 in Algorithm 1) that adding session keywords to all the queries except for the first one and further clarifying the ambiguous queries by adding query keywords from their last three queries.

1.2 Historical Answer Expansion

This work proposes to facilitate the pretrained passage re-ranking BERT model to solve the issues regarding data sparsity and transfer the gap between conversational QA tasks and conversational-information-seeking (CIS) problem. Incorporating historical answers in conversational QA tasks had been proposed to expand the

Text REtrieval Conference '19, Nov 13–15, 2019, Gaithersburg, Maryland

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

Algorithm 1: Historical Query Expansion

Input: $\mathcal{S} = \{Q_i\}_{i=1}^{\mathcal{N}}$, $\mathcal{D}$
Output: $\mathcal{S}$

$W_{\mathcal{S}} \leftarrow ()$; $W_Q \leftarrow ()$
for $i = 1$ *to* $\mathcal{N}$ **do**
  for $k = 1$ *to* $n(i)$ **do**
    $r_k^i = \max_{d_j \in \mathcal{D}} \mathcal{F}(d_j, (q_k^i))$
    if $r_k^i > r_{\mathcal{S}}$ **then**
      $W_{\mathcal{S}}$.insert(q_k^i)
    if $r_k^i > r_Q$ **then**
      W_Q.insert(q_k^i)
  if $i > 1$ **then**
    $R_i = \max_{d_j \in \mathcal{D}} \mathcal{F}(d_j, Q_i) = \max_{d_j \in \mathcal{D}} \mathcal{F}(d_j, (q_0^i, q_1^i, \cdots, q_{n(i)}^i))$
    Q_i.insert(q_k^n) for all $q_k^n \in W_{\mathcal{S}}$
    if $R_i < \theta$ **then**
      Q_i.insert(q_k^n) for all $q_k^n \in W_Q \wedge n \geq i-3$
return $\mathcal{S}$

answers in each turn [8]. In the previous work, inserting historical answers with additional tokens into query-answer pairs makes the proposed method fuse with the state-of-the-art BERT model and its variants easily [3–5, 11]. However, the data sparsity and unique CIS setting in the CAsT make training complicated models infeasible, neither directly training a conversational QA model (e.g., BERT with historical answer embedding [8]) on CAsT dataset nor jointly training with other conversational QA tasks that focus on finding fine-grained answer span within passages. Hence, the historical answer expansion (HAE) is proposed to solve the issues mentioned above. Specifically, we use a pretrained BERT model, which is trained for passage re-ranking on MS MARCO dataset [6], to estimate query-passage log-likelihood scores and directly mix the scores from the current i-th turn and the previous $(i-l)$-th turns. The negative log-likelihood scores from previous turns are multiplied by a constant factor λ, which serves as a decay factor to lower the weight of historical answers. In this work, we only consider the pairs from previous one turn ($l = 1$): $(Q_{i-1}, \mathcal{D}_{i-1})$, where $\mathcal{D}_i = \{d_j\}_{j=1}^{k}$ stands for the top-k passages of each i-th turn. The final candidate list for each conversational turn is the re-ranked list cut off at top-k ($k = 1000$ in our experiments) passages according to the mixed negative log-likelihood scores of query-passages pairs: $((Q_i, \mathcal{D}_i), (Q_{i-1}, \mathcal{D}_{i-1}))$.

We further illustrate the idea of HAE in Algorithm 2. In HAE, we define a collection of our query-passage pairs: $\mathcal{A} = \{(Q_i, \mathcal{D}_i)\}_{i=1}^{\mathcal{N}}$. Our goal is to insert the passage candidates $\mathcal{D}_i$ for each query Q_i from its previous previous passage candidates $\mathcal{D}_{i-1}$. A hyperparameter λ is introduced to modify the log-likelihood $\mathcal{L}_{ij}$ from the pretrained BERT model for each pair of (Q_i, d_{ij}), where $d_{ij} \in \mathcal{D}_i; j = 1, 2, \ldots, k$. Besides, we keep d_{ij} for the corresponding i-th turn but drop $d_{(i-1)j}$, if $d_{(i-1)j}$ is a duplicated passage from the previous $(i-1)$-th turn. Note that the HAE method only involves λ tuning, which is not only a "training"-free method but can be easily integrated with HQE or other query expansion techniques.

Algorithm 2: Historical Answer Expansion

Input: $\mathcal{A} = \{(Q_i, \mathcal{D}_i)\}_{i=1}^{\mathcal{N}}, \lambda$
Output: $\mathcal{A}$

Initialize $\mathcal{P}[\mathcal{N}][k]$; $\mathcal{P}_{temp}[k]$
for $i = 1$ *to* $\mathcal{N}$ **do**
  for $j = 1$ *to* k **do**
    Estimate log-likelihood: $\mathcal{L}_{ij}(Q_i, d_{ij})$
    $\mathcal{P}[i][j] \leftarrow (d_{ij}, \mathcal{L}_{ij})$
for $i = 1$ *to* $\mathcal{N}$ **do**
  Append $\mathcal{P}_{temp}$ with $\mathcal{P}[i]$
  Sort $\mathcal{P}_{temp}$ by $\mathcal{L}$
  for $j = 1$ *to* k **do**
    $n_i j = d_i j \in \mathcal{P}_{temp}[j]$
  Update $\mathcal{D}_i$ with $\{n_j\}_{j=1}^{k}$
  for $j = 1$ *to* k **do**
    $\mathcal{P}_{temp}[j] \leftarrow (d_{ij}, \lambda \cdot \mathcal{L}_{ij}) \in \mathcal{P}[i][j]$
return $\mathcal{A}$

2 EMPIRICAL ANALYSIS

Table 1: Performance comparison on CAsT training set

Condition	1	2	3	4	5	6
Retrieval	Title	Title	Title	HQexp	HQexp	HQexp
Re-ranking	Title	HQExp	Coref	Title	HQExp	Coref
R@1000	0.774	0.774		**0.818**	**0.818**	
mAP	0.187	**0.194**	-	0.189	0.192	-
mRR@10	0.273	**0.282**		0.257	0.264	
			+HAExp			
R@1000	0.790	0.790		**0.844**	**0.844**	
mAP	0.187	0.192	-	0.193	**0.197**	-
mRR@10	0.273	**0.279**		0.268	0.277	

Table 2: Co-reference effect on CAsT training subset

Condition	1	2	3	4	5	6
Retrieval	Title	Title	Title	HQexp	HQexp	HQexp
Re-ranking	Title	HQExp	Coref	Title	HQExp	Coref
R@1000	0.897	0.897	0.897	0.859	0.859	0.859
mAP	0.258	0.291	**0.392**	0.261	0.274	0.374
mRR@10	0.358	0.442	**0.525**	0.377	0.433	0.544
			+HAExp			
R@1000	0.910	0.910	0.910	0.863	0.863	0.863
mAP	0.257	0.285	0.388	0.261	0.272	0.371
mRR@10	0.358	0.440	0.524	0.377	0.431	0.520

2.1 Datasets and Preprocessing

Figure 2 shows the statistics of CAsT training set, including 30 sessions with 269 turns in total. Passages with graded judgements

Table 3: Overall performance of submitted runs on CAsT evaluation set

Team Run Entry	median	CFDA_CLIP 1	h2oloo 2	h2oloo 3	h2oloo 4	h2oloo 5	CFDA_CLIP 6	CFDA_CLIP 7	CFDA_CLIP 8
Indexed corpus		MARCO	CAsT	CAsT	CAsT	CAsT	CAsT	CAsT+D2Q	CAsT+D2Q
Retrieval		Title	Title	Title	Title+RM3	HQExp	Coref+RM3	Title	HQExp
Re-ranking		Coref	HQExp	Coref	Coref	Coref	Coref	HQExp	Coref
+HAExp						✓			✓
R@1000	-	0.412	0.632	0.632	0.639	0.689	**0.812**	0.611	0.695
mAP	0.174	0.226	0.274	0.324	0.321	0.354	**0.395**	0.269	0.363
mAP@5	0.042	0.071	0.066	0.082	0.081	0.096	**0.101**	0.068	0.099
NDCG@5	0.296	0.459	0.427	0.530	0.532	0.564	**0.576**	0.427	0.568

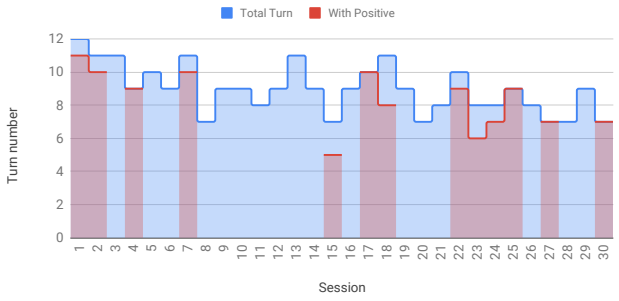

Figure 2: CAsT training data statistics

of relevance to queries are provided for 120 turns among the 269 turns. The judgements are graded with three levels (2 for very relevant, 1 for relevant, and 0 for not relevant). For simplicity, in our experiments, we consider the query-passage pairs with the grade higher than 0 as positive and remove all the pairs graded 0, resulting in 108 turns with 640 positive pairs in total.

2.2 Evaluation and Settings

We use the CAsT training set to evaluate model performance in terms of recall (R@1000), mean average precision (mAP) and mean reciprocal rank (mRR@10). At the stage of document retrieval, we use the Anserini toolkit [10] to index and retrieve the top-1000 relevant passages for each query with BM25 plus fine-tuned RM3. As for reranking, we use the BERT-Large model fine tuned on the MS MARCO dataset [6], the queries of which are similar to the ones in CAsT, as our reranker.

As shown in Table 1, in order to find the optimal condition for CAsT, we try different methods to rewrite input queries at both the stages of document retrieval and reranking. The three methods are described as follows:

- Title: adding the title of the current session to the input query
- Historical Query Expansion (**HQExp**): an automatic keyword expansion method proposed in Section 1.1.
- Coreference: the manually annotated queries with coreference resolution provided by the organizer.

In addition, for each condition, we also perform our proposed Historical Answer Expansion **(HAExp)** described in Section 1.2. Note that since the annotated queries are provided only for the first two sessions in the training set, we conduct another experiment on the training subset to see the effect of coreference resolution, the results of which are shown in Table 2.

2.3 Quantitative Analysis

2.3.1 Results on CAsT training set. The results using queries' raw texts with different kinds of query expansion techniques are shown in Table 1. An observation from the table, the ad-hoc methods perform well both in recall and ranking metrics. The proposed HQExp method boosts mAP and mRR@10 by 3.7% and 3.3%, by comparing with conditions: 1 and 2, respectively. In addition, with the HQExp method involved in the retrieval stage, the proposed method further gains 5.7% in R@1000 by comparing conditions: 1 and 4. The combination that involves the HQExp method both in retrieval and re-ranking stage — condition 5 — does not win over the method that HQExp only involved in re-ranking stage — condition 2. However, the ranking metrics: mAP and mRR@10, still follows an increasing trend, comparing the pair of condition 1 and 2 (and the other one, condition 4 and 5, respectively).

As for the proposed post-processing HAExp method, it also performs well. HAExp method boosts the recall metric from all conditions (1, 2, 4, and 5) and both ranking metrics by 2.1% and 2.6% in mAP (4.3% and 4.9% in mRR@10), comparing condition 4 and 5. Albeit the HAExp method does not seems to make a positive impact on the ranking metrics with only title query expansion scenario (conditions: 1 and 2), the combination of HQExp plus HAExp has the best entry on CAsT training set in terms of R@1000 and mAP.

We experiment to check the effectiveness of coreference in passage re-ranking as records shown in Table 2. Observed from Table 2, not surprisingly, the result of the coreference resolved queries achieves the best performance. Among other query expansion techniques, the coreference resolved query has the best ranking metrics: 0.392 in terms of mAP and 0.525 in terms of mRR@10. It seems that the positive impact of the proposed methods in full training set disappears in the annotated subset, albeit the HAExp method still boosts R@1000. However, the amount of data may be too few for us to judge the effectiveness of the combination of HQExp, HAExp, and the coreference resolved queries.

2.3.2 Systems submitted to TREC. We submitted a total of eight runs for CAsT this year with the techniques we mentioned in the

previous section. In addition, we further consider two baselines described as follows:

- MARCO: a baseline that only conducts inverted indexing and retrieves paragraphs from the collections of the MS MARCO dataset only.
- Document2Query (D2Q): a query expansion method that expands a paragraph with its relevant queries [7]. Which model is trained on the MS MARCO dataset, and we only use the model to expand the paragraphs in the MS MARCO collections before inverted indexing.

The results of the evaluation are demonstrated in Table 3. The columns indicated the conditions of our final submissions regarding the proposed ad-hoc methods and the two baselines mentioned above. Comparing to the statistics of total the 21 teams' submission provided by TREC, it appears that the simplest baseline, which only uses inverted indexing and BERT re-ranker with MS MARCO corpus, outperforms 50% of submissions. We observed an interesting phenomenon among our submissions; the most straightforward method takes all. With coreference resolution involved in both document retrieval and re-ranking stages, the best entry is CFDA_CLIP_Run6, which scores 0.812 in R@1000, 0.395 in mAP, 0.101 in mAP@5, and 0.576 in NDCG@5. Without involving coreference resolution in the retrieval stage, the full combination of HQExp and HAExp with coreference resolved queries performs worse than the best run. However, which combination still deliver the best performance among other baselines. The effectiveness of the proposed methods needs a detail examination since the data distribution could be different from the training set. To be more specific, the difference comes from the issue that the WAPO collection is removed from the evaluation set due to its problem in removing duplicated paragraph.

ACKNOWLEDGMENTS

Especially thanks to Jimmy Lin for the instructions and ideas.

REFERENCES

[1] Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading Wikipedia to Answer Open-Domain Questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics.

[2] Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wen-tau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. Quac: Question answering in context. *arXiv preprint arXiv:1808.07036* (2018).

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv preprint arXiv:1810.04805* (2018).

[4] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. *arXiv preprint arXiv:1909.11942* (2019).

[5] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692* (2019).

[6] Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage Re-ranking with BERT. *arXiv preprint arXiv:1901.04085* (2019).

[7] Rodrigo Nogueira, Wei Yang, Jimmy Lin, and Kyunghyun Cho. 2019. Document Expansion by Query Prediction. *arXiv preprint arXiv:1904.08375* (2019).

[8] C. Qu, L. Yang, M. Qiu, W. B. Croft, Y. Zhang, and M. Iyyer. 2019. BERT with History Answer Embedding for Conversational Question Answering. In *SIGIR '19*.

[9] Siva Reddy, Danqi Chen, and Christopher D Manning. 2019. Coqa: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics* 7 (2019), 249–266.

[10] Peilin Yang, Hui Fang, and Jimmy Lin. 2018. Anserini: Reproducible ranking baselines using Lucene. *Journal of Data and Information Quality (JDIQ)* 10, 4 (2018), 16.

[11] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. 2019. XLNet: Generalized Autoregressive Pretraining for Language Understanding. *arXiv preprint arXiv:1906.08237* (2019).