

Query Resolution for Conversational Search with Limited Supervision

Nikos Voskarides¹ Dan Li¹ Pengjie Ren¹ Evangelos Kanoulas¹ Maarten de Rijke^{1,2}

¹University of Amsterdam, Amsterdam, The Netherlands ²Ahold Delhaize, Zaandam, The Netherlands
nickvosk@gmail.com, d.li@uva.nl, p.ren@uva.nl, e.kanoulas@uva.nl, m.derijke@uva.nl

ABSTRACT

In this work we focus on multi-turn passage retrieval as a crucial component of conversational search. One of the key challenges in multi-turn passage retrieval comes from the fact that the current turn query is often underspecified due to zero anaphora, topic change, or topic return. Context from the conversational history can be used to arrive at a better expression of the current turn query, defined as the task of query resolution. In this paper, we model the query resolution task as a binary term classification problem: for each term appearing in the previous turns of the conversation decide whether to add it to the current turn query or not. We propose QuReTeC (**Q**uery **R**esolution by **T**erm **C**lassification), a neural query resolution model based on bidirectional transformers. We propose a distant supervision method to automatically generate training data by using query-passage relevance labels. Such labels are often readily available in a collection either as human annotations or inferred from user interactions. We show that QuReTeC outperforms state-of-the-art models, and furthermore, that our distant supervision method can be used to substantially reduce the amount of human-curated data required to train QuReTeC. We incorporate QuReTeC in a multi-turn, multi-stage passage retrieval architecture and demonstrate its effectiveness on the TREC CAsT dataset.

KEYWORDS

Conversational search; Query resolution

ACM Reference Format:

Nikos Voskarides, Dan Li, Pengjie Ren, Evangelos Kanoulas, and Maarten de Rijke. 2020. Query Resolution for Conversational Search with Limited Supervision. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '20)*, July 25–30, 2020, Virtual Event, China. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3397271.3401130>

1 INTRODUCTION

Conversational AI deals with developing dialogue systems that enable interactive knowledge gathering [17]. A large portion of work in this area has focused on building dialogue systems that are capable of engaging with the user through chit-chat [23] or helping the

Table 1: Excerpt from an example conversational dialog. Co-occurring terms in the conversation history and the relevant passage to the current turn (#4) are shown in bold-face.

Turn	Query
1	who formed saosin ?
2	when was the band founded?
3	what was their first album?
4	when was the album released? <i>resolved: when was saosin 's first album released?</i>

*Relevant passage to turn #4: The original lineup for **Saosin**, consisting of Burchell, Shekoski, Kennedy and Green, was formed in the summer of 2003. On June 17, the **band** released their **first** commercial production, the EP Translating the Name.*

user complete small well-specified tasks [32]. In order to improve the capability of such systems to engage in complex information seeking conversations [34], researchers have proposed information seeking tasks such as conversational question answering (QA) over simple contexts, such as a single-paragraph text [7, 37]. In contrast to conversational QA over simple contexts, in conversational search, a user aims to interactively find information stored in a large document collection [10].

In this paper, we study multi-turn passage retrieval as an instance of conversational search: given the conversation history (the previous turns) and the current turn query, we aim to retrieve passage-length texts that satisfy the user's underlying information need [11]. Here, the current turn query may be under-specified and thus, we need to take into account context from the conversation history to arrive at a better expression of the current turn query. Thus, we need to perform *query resolution*, that is, add missing context from the conversation history to the current turn query, if needed. An example of an under-specified query can be seen in Table 1, turn #4, for which the gold standard query resolution is: “when was saosin 's first album released?”. In this example, context from all turns #1 (“saosin”), #2 (“band”) and #3 (“first”) have to be taken into account to arrive to the query resolution.

Designing automatic query resolution systems is challenging because of phenomena such as zero anaphora, topic change and topic return, which are prominent in information seeking conversations [50]. These phenomena are not easy to capture with standard NLP tools (e.g., coreference resolution). Also, heuristics such as appending (part of) the conversation history to the current turn query are likely to lead to query drift [27]. Recent work has modeled query resolution as a sequence generation task [15, 21, 36]. Another way of implicitly solving query resolution is by query modeling [18, 42, 47], which has been studied and developed under the setup of session-based search [5, 6].

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 the author(s) 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.

SIGIR '20, July 25–30, 2020, Virtual Event, China

© 2020 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 978-1-4503-8016-4/20/07...\$15.00

<https://doi.org/10.1145/3397271.3401130>

In this paper, we propose to model query resolution for conversational search as a binary term classification task: for each term in the previous turns of the conversation decide whether to add it to the current turn query or not. We propose QuReTeC (**Q**uery **R**esolution by **T**erm **C**lassification), a query resolution model based on bidirectional transformers [43] – more specifically BERT [13]. The model encodes the conversation history and the current turn query and uses a term classification layer to predict a binary label for each term in the conversation history. We integrate QuReTeC in a standard two-step cascade architecture that consists of an initial retrieval step and a reranking step. This is done by using the set of terms predicted as relevant by QuReTeC as query expansion terms.

Training QuReTeC requires binary labels for each term in the conversation history. One way to obtain such labels is to use human-curated gold standard query resolutions [15]. However, these labels might be cumbersome to obtain in practice. On the other hand, researchers and practitioners have been collecting general-purpose passage relevance labels, either by the means of human annotations or by the means of weak signals, e.g., clicks or mouse movements [19]. We propose a distant supervision method to automatically generate training data, on the basis of such passage relevance labels. The key assumption is that passages that are relevant to the current turn share context with the conversation history that is missing from the current turn query. Table 1 illustrates this assumption: the relevant passage to turn #4 shares terms with the conversation history. Thus, we label the terms that co-occur in the relevant passages¹ and the conversation history as relevant for the current turn.

Our main contributions can be summarized as follows:

- (1) We model the task of query resolution as a binary term classification task and propose to address it with a neural model based on bidirectional transformers, QuReTeC.
- (2) We propose a distant supervision approach that can use general-purpose passage relevance data to substantially reduce the amount of human-curated data required to train QuReTeC.
- (3) We experimentally show that when integrating the QuReTeC model in a multi-stage ranking architecture we significantly outperform baseline models. Also, we conduct extensive ablation studies and analyses to shed light into the workings of our query resolution model and its impact on retrieval performance.

2 RELATED WORK

Conversational search. Early studies on conversational search have focused on characterizing information seeking strategies and building interactive IR systems [3, 4, 9, 30]. Vtyurina et al. [45] investigated human behaviour in conversational systems through a user study and find that existing conversational assistants cannot be effectively used for conversational search with complex information needs. Radlinski and Craswell [35] present a theoretical framework for conversational search, which highlights the need for multi-turn interactions. Dalton et al. [11] organize the Conversational Assistance Track (CAST) at TREC 2019. The goal of the track is to establish a concrete and standard collection of data with information needs to make systems directly comparable. They

release a multi-turn passage retrieval dataset annotated by experts, which we use to compare our method to the baseline methods.

Query resolution. Query resolution has been studied in the context of dialogue systems. Raghu et al. [36] develop a pipeline model for query resolution in dialogues as text generation. Kumar and Joshi [21] follow up on that work by using a sequence to sequence model combined with a retrieval model. However, both these works rely on templates that are not available in our setting. More related to our work, Elgohary et al. [15] studied query resolution in the context of conversational QA over a single paragraph text. They use a sequence to sequence model augmented with a copy and an attention mechanism and a coverage loss. They annotate part of the QuAC dataset [7] with gold standard query resolutions on which they apply their model and obtain competitive performance. In contrast to all the aforementioned works that model query resolution as text generation, we model query resolution as binary term classification in the conversation history.

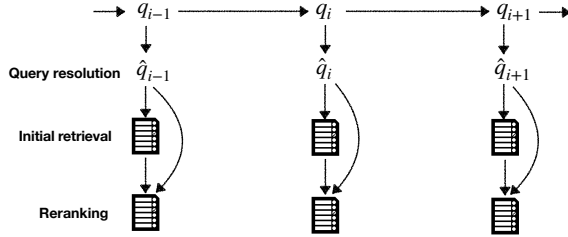
Query modeling. Query modeling has been used in session search, where the task is to retrieve documents for a given query by utilizing previous queries and user interactions with the retrieval system [6]. Guan et al. [18] extract substrings from the current and previous turn queries to construct a new query for the current turn. Yang et al. [47] propose a query change model that models both edits between consecutive queries and the ranked list returned by the previous turn query. Van Gysel et al. [42] compare the lexical matching session search approaches and find that naive methods based on term frequency weighing perform on par with specialized session search models. The methods described above are informed by studies of how users reformulate their queries and why [41], which, in principle, is different in nature from conversational search. For instance, in session search users tend to add query terms more than removing query terms, which is not the case in (spoken) conversational search. Another form of query modeling is query expansion. Pseudo-relevance feedback is a query expansion technique that first retrieves a set of documents that are assumed to be relevant to the query, and then selects terms from the retrieved documents that are used to expand the query [1, 22, 29]. Note that pseudo-relevance feedback is fundamentally different from query resolution: in order to revise the query, the former relies on the top-ranked documents, while the latter only relies on the conversation history.

Distant supervision. Distant supervision can be used to obtain large amounts of noisy training data. One of its most successful applications is relation extraction, first proposed by Mintz et al. [26]. They take as input two entities and a relation between them, gather sentences where the two entities co-occur from a large text corpus, and treat those as positive examples for training a relation extraction system. Beyond relation extraction, distant supervision has also been used to automatically generate noisy training data for other tasks such as named entity recognition [49], sentiment classification [39], knowledge graph fact contextualization [44] and dialogue response generation [38]. In our work, we follow the distant supervision paradigm to automatically generate training data for query resolution in conversational search by using query-passage relevance labels.

¹A relevance passage contains not only the answer to the question but also context and supporting facts that allow the algorithm or the human to reach to this answer.

Table 2: Notation used in the paper.

Name	Description
$terms(x)$	set of terms in term sequence x
D	Passage collection
q_i	Query at the current turn i
$q_{1:i-1}$	Sequence of previous turn queries
q_i^*	Gold standard resolution of q_i
$E_{q_i}^*$	Gold standard resolution terms for q_i , see Eq. (2)
\hat{q}_i	Predicted resolution of q_i
$p_{q_i}^*$	A relevant passage for q_i

**Figure 1: Illustration of our multi-turn passage retrieval pipeline for three turns.**

3 MULTI-TURN PASSAGE RETRIEVAL PIPELINE

In this section we provide formal definitions and describe our multi-turn passage retrieval pipeline. Table 2 lists notation used in this paper.

3.1 Definitions

Multi-turn passage ranking. Let $[q_1, \dots, q_{i-1}, q_i]$ be a sequence of conversational queries that share a common topic T . Let q_i be the current turn query and $q_{1:i-1}$ be the conversation history. Given q_i and $q_{1:i-1}$, the task is to retrieve a ranked list of passages L from a passage collection D that satisfy the user’s information need.²

In the multi-turn passage ranking task, the current turn query q_i is often underspecified due to phenomena such as zero anaphora, topic change, and topic return. Thus, context from the conversation history $q_{1:i-1}$ must be taken into account to arrive at a better expression of the current turn query q_i . This challenge can be addressed by query resolution.

Query resolution. Given the conversation history $q_{1:i-1}$ and the current turn query q_i , output a query \hat{q}_i that includes both the existing information in q_i and the missing context of q_i that exists in the conversation history $q_{1:i-1}$.

3.2 Multi-turn passage retrieval pipeline

Figure 1 illustrates our multi-turn passage retrieval pipeline. We use a two-step cascade ranking architecture [46], which we augment with a query resolution module (Section 4). First, the unsupervised initial retrieval step outputs the initial ranked list L_1 (Section 3.2.1).

²We follow the TREC CAsT setup and only take into account $q_{1:i-1}$ but not the passages retrieved for $q_{1:i-1}$.

Second, the re-ranking step outputs the final ranked list L (Section 3.2.2). Below we describe the two steps of the cascade ranking architecture.

3.2.1 Initial retrieval step. In this step we obtain the initial ranked list L_1 by scoring each passage p in the passage collection D with respect to the resolved query \hat{q}_i using a lexical matching ranking function f_1 . We use query likelihood (QL) with Dirichlet smoothing [51] as f_1 , since it outperformed other ranking functions such as BM25 in preliminary experiments over the TREC CAsT dataset.

3.2.2 Reranking step. In this step, we re-rank the list L_1 by scoring each passage $p \in L_1$ with a ranking function f_2 to obtain the final ranked list L . To construct f_2 , we use rank fusion and combine the scores obtained by f_1 (used in initial retrieval step) and a supervised neural ranker f_n . Next, we describe the neural ranker f_n .

Supervised neural ranker. We use BERT [13] as the neural ranker f_n , as it has been shown to achieve state-of-the-art performance in ad-hoc retrieval [25, 33, 48]. Also, BERT has been shown to prefer semantic matches [33], and thereby can be complementary to f_1 , which is a lexical matching method. As is standard when using BERT for pairs of sequences, the input to the model is formatted as $[\langle \text{CLS} \rangle, \hat{q}_i \langle \text{SEP} \rangle, p]$, where $\langle \text{CLS} \rangle$ is a special token, \hat{q}_i is the resolved current turn query, p is the passage. We add a dropout layer and a linear layer l_a on top of the representation of the $\langle \text{CLS} \rangle$ token in the last layer, followed by a tanh function to obtain f_n [25]. We score each passage $p \in L_1$ using f_n to obtain L_n . We fine-tune the pretrained BERT model using pairwise ranking loss on a large-scale single-turn passage ranking dataset [48]. During training we sample as many negative as positive passages per query.

Rank fusion. We design f_2 such that it combines lexical matching and semantic matching [31]. We use Reciprocal Rank Fusion (RRF) [8] to combine the score obtained by the lexical matching ranking function f_1 , and the semantic matching supervised neural ranker f_n . We choose RRF because of its effectiveness in combining individual rankers in ad-hoc retrieval and because of its simplicity (it has only one hyper-parameter). We define f_2 as the RRF of L_1 and L_n [8]:

$$f_2(p) = \sum_{L' \in \{L_1, L_n\}} \frac{1}{k + \text{rank}(p, L')}, \quad (1)$$

where $\text{rank}(p, L')$ is the rank of passage p in a ranked list L' , and k is a hyperparameter.³ We score each passage p in the initial ranked list L_1 with f_2 to obtain the final ranked list L .

Since developing specialized re-rankers for the task at hand is not the focus of this paper, we leave more sophisticated methods for choosing the neural ranker f_n and for combining multiple rankers as future work. In the next section, we describe our query resolution model, QuReTeC, which is the focus of this paper.

4 QUERY RESOLUTION

In this section we first describe how we model query resolution as term classification (Section 4.1), then present our query resolution model, QuReTeC, (Section 4.2), and finally describe how we generate distant supervision labels for the model (Section 4.3).

³We set $k = 60$ and do not tune it.