Adaptive Utterance Rewriting for Conversational Search

Ida Mele[a], Cristina Ioana Muntean[b], Franco Maria Nardini[b],
Raffaele Perego[b], Nicola Tonellotto[c], Ophir Frieder[d]

[a]*IASI-CNR, Rome, Italy*
[b]*ISTI-CNR, Pisa, Italy*
[c]*University of Pisa, Pisa, Italy*
[d]*Georgetown University, Washington, DC, USA*

Abstract

In a conversational context, a user converses with a system through a sequence of natural-language questions, i.e., utterances. Starting from a given subject, the conversation evolves through sequences of user utterances and system replies. The retrieval of documents relevant to an utterance is difficult due to informal use of natural language in speech and the complexity of understanding the semantic context coming from previous utterances. We adopt the 2019 TREC Conversational Assistant Track (CAsT) framework to experiment with a modular architecture performing in order: (i) automatic utterance understanding and rewriting, (ii) first-stage retrieval of candidate passages for the rewritten utterances, and (iii) neural re-ranking of candidate passages. By understanding the conversational context, we propose adaptive utterance rewriting strategies based on the current utterance and the dialogue evolution of the user with the system. A classifier identifies those utterances lacking context information as well as the dependencies on the previous utterances. Experimentally, we evaluate the proposed architecture in terms of traditional information retrieval metrics at small cutoffs. Results demonstrate the effectiveness of our techniques, achieving an improvement up to 0.6512 (+201%) for P@1 and 0.4484 (+214%) for nDCG@3 w.r.t. the CAsT baseline.

Keywords: Conversational IR, Neural re-ranking, Query rewriting.

1. Introduction

Conversational Information Retrieval (IR) has recently gained interest due to the widespread popularity of conversational assistant systems. Thanks to the recent advances in automatic speech recognition and understanding, conversational assistant systems are widely used in chatbots and smart home devices, e.g., Google Home and Amazon Alexa, as well as in wearable devices and smartphones, e.g., Apple Siri, Google Now, and Microsoft Cortana. These systems help users in activities ranging from simple to complex tasks. Simple tasks include checking the weather forecast and managing music streaming services, while complex

Email addresses: `ida.mele@iasi.cnr.it` (Ida Mele), `cristina.muntean@isti.cnr.it` (Cristina Ioana Muntean), `francomaria.nardini@isti.cnr.it` (Franco Maria Nardini), `raffaele.perego@isti.cnr.it` (Raffaele Perego), `nicola.tonellotto@unipi.it` (Nicola Tonellotto), `ophir@ir.cs.georgetown.edu` (Ophir Frieder)

include performing e-commerce transactions. Users, including the elderly who are typically unfamiliar with technology, find comfort as they converse with the system in a multi-turn conversation of questions (i.e., utterances)[1] and replies.

Contrasting their ease of use, technologies enabling conversational assistant systems are complex and heterogeneous. We focus on a key conversational assistant component, the conversational IR system. Given a user utterance, the conversational IR system is tasked to retrieve one to a few documents, from within a document collection, that contain the answer to the information need expressed in the utterance. Our efforts enhance utterance context, improving conversational IR system accuracy, and thus, conversational flow.

The automatic management of user utterances is difficult as the conversational IR system must: (i) understand the utterance and its relation with the context expressed in the conversation, (ii) find relevant information in a knowledge base or document collection, and (iii) select a single or a very narrow list of results preferably including the specific answer. Question understanding must deal with vague or misleading formulations of the utterances. Indeed, natural language utterances are prone to the ambiguity and polysemy of words, the presence of acronyms, mistakes, and grammar misuses. More importantly, a complex information need cannot be resolved with a single question; rather, the user formulates multiple subsequent utterances that can be related to each other.

Consider the following questions in a conversation: (1) “Is Red Bull bad for you?”, (2) “Can it kill you?”, (3) “How much can you drink in a day?”, (4) “What is taurine?”, (5) “What are its health effects?” While the first and fourth questions are relatively easy to process by IR systems, in the second and third questions, there are explicit and implicit references to the first subject of the conversation, i.e., “Red Bull”. The fifth question refers instead to a specific substance, i.e., “taurine”, introduced later during the conversation.

Even in this short conversation, it is possible to identify *self-explanatory* utterances that do not require rewriting, i.e., the first and fourth questions. However, the second question is not self-explanatory, and there is an explicit reference to a previous topic, i.e., “it” refers to “Red Bull” mentioned in the first question. The third question is even more tricky as there is no explicit reference to a previous topic, “How much can you drink in a day?” one may ask “drink what?”, so the question itself lacks context. In the conversation, we can also observe a *topic drift*, i.e., “Red Bull” → “Taurine”, that makes the fifth question refer to the newly introduced topic, i.e., “taurine” and not to the initial subject of the conversation, i.e., “Red Bull”.

Traditional IR systems are not designed to answer utterances in conversational dialogues as the one shown above. They focus on retrieving relevant documents in response to self-explanatory user queries, while a conversational search system should also deal with the enrichment of utterance with contextual information

[1] Although the user requests are usually formulated as questions, there are cases in which they are not (e.g., “Tell me about...” or “Show me...”). Thus, we will preferably use the term utterance unless we explicitly refer to a request formulated as a question.

permitting to retrieve the best documents answering the user information need. Our work focuses precisely on utterance understanding to enrich those utterances that lack context. Previously, we investigated topic propagation in multi-turn conversations and experimented with some heuristic techniques for utterance rewriting [1]. Furthering those results, we now propose a generalized architecture for conversational search implementing advanced strategies for utterance understanding and enrichment.

Research Objectives. The research presented improves conversational information retrieval, a domain where the user expresses her information needs verbally, and the system automatically understands the user utterances to retrieve a good set of candidate answers. This is challenging as some utterances are not self-explanatory and some are ambiguous. Moreover, since the search involves a multi-turn conversation, some of the requests lack context as they refer to a topic previously mentioned in the conversation. Traditional information retrieval does not support such kind of interactions; so we must improve conversational search systems. Our objective is to prove that adding missing context chosen in a proper way improves the retrieval task in conversational systems. We also prove that utterance classification can be used to find the best rewriting technique for transforming a raw utterance into an enriched and self-explanatory one that is easy to process for an automatic information retrieval system.

Novel Contributions. We present strategies for utterance enrichment driven by an in-depth analysis of the current utterance and the evolution of the previous conversational turns. We rely on effective classifiers to promptly detect the utterance context and to choose the best utterance rewriting strategy to adopt for improving retrieval effectiveness. We experiment with different utterance classification and rewriting techniques, showing that their adoption, together with linguistic analysis, allows us to understand how to enrich utterances with keywords enclosing the right conversational topic and yielding to utterances that can be effectively answered by the IR system. We integrate our novel techniques into a modular pipeline architecture performing: (i) automatic utterance rewriting based on cascading or multi-class utterance classification, (ii) first-stage retrieval of candidate documents for the rewritten utterances, and (iii) neural re-ranking of candidate documents.

Utterance rewriting exploits linguistic analysis performed on the utterance and the output of its context classification to devise additional content that explicitly refers to the implicit subject of the conversation or to the specific facet of the question. The classification allows us to understand if the utterance is self-explanatory or it misses explicit references to its context. In this latter case, the classification also helps to identify which context is more suitable for enriching it. The second step, i.e., *first-stage retrieval*, narrows down the search space by using the rewritten utterance to retrieve a limited set of relevant results. The last step, i.e., *neural re-ranking*, exploits a contextualized language model based on BERT to re-rank the retrieved documents [2].

A comprehensive experimental evaluation of our modular architecture in terms of popular IR metrics, e.g, P@1, P@3, and nDCG@3, is performed by adopting the TREC Conversational Assistant Track 2019 (CAsT) framework [3]. Our results show that adding context to the utterances based on the choice of our classifiers is crucial for a more effective retrieval of the relevant documents when an explicit reference to the conversational topic is missing. We also note that the conversation subject is typically enclosed in the first utterance. However, context changes are also common in multi-turn conversations; so it is important to promptly identify them to specialize the rewriting technique and to improve the quality of the results. Our techniques for utterance rewriting take into account all these variations, surpassing, in performance, other approaches based on co-referencing [4].

Regarding first-stage retrieval and neural re-ranking, experimental results show that pseudo-relevance feedback is beneficial as it further expands the query with keywords extracted from the top K results allowing an improvement of recall. Neural re-ranking provides, on the other hand, an important boost in precision at very short cutoffs. Experiments show that our best approach achieves an improvement of 0.4855 (+215%) for P@1 and 0.4566 (+188%) for P@3 at the first-stage retrieval and of 0.6512 (+213%) for P@1 and 0.5988 (+200%) for P@3 at the neural-based re-ranking stage w.r.t. the CAsT provided baseline. The improvement for nDCG@3 is instead of 0.3072 (+208%) at first-stage retrieval and 0.4484 (+215%) at neural re-ranking stage, respectively.

Summarizing, the novel and unpublished contributions presented are:

- An approach for addressing the problem of utterance understanding based on a pipeline of three components: (1) a context identification component, to detect a subset of the previous utterances considered relevant for identifying the context of the current utterance, (2) a context generator component, responsible to extract from the previous utterances the keywords needed to enrich the current utterance with the missing information, and (3) an utterance rewriting component, to generate the query that will be processed by an automatic IR system. Our approach allows to detect the context of an utterance and leverage it to generate a query for an IR system.

- Utterance classifiers for context identification trained thanks to our utterance taxonomy based on 3 classes: *self-explanatory* utterance, utterance depending on the *first* or *previous* topic. Our proposed classifiers are based on gradient boosting decision trees and on bidirectional transformer neural networks, and we provide an evaluation of the accuracy of different combinations of classifiers.

- A detailed experimental evaluation of the impact of our utterance understanding pipeline on the effectiveness of a multi-stage retrieval system, composed by a first-stage retrieval system and a second-stage neural re-ranking system. We report different effectiveness metrics for different baselines, competitors, and for the newly proposed methods for both the first-stage and the end-to-end systems. The strategies proposed in this paper are statistically better than the baseline and the rewriting techniques proposed in [1].

Paper Structure. Section 2 discusses related work, while Section 3 introduces our conversational IR architecture and provides a detailed functional description of its utterance understanding components. Section 4.2 reports on the training of the utterance classifiers for context identification and discusses their performance. The end-to-end performance of the proposed conversational IR system is carefully assessed in Section 5, where experimental settings and baselines are also introduced. Finally, Section 6 presents the concluding remarks.

2. Related Work

Early studies on conversational search date back to the 1980s, when efforts focused on investigating information seeking strategies for interactive IR systems [5, 6]. More recently, Vtyurina *et al.* study the human interaction with conversational systems [7] to conclude that existing conversational assistants do not deal effectively with complex information needs. The difficulty in handling conversations with complex and context-dependent information needs is also highlighted by Radlinski and Craswell [8]. They highlight the need of a multi-turn interaction model for conversational search and propose a theoretical framework to better cover contextual information needs. Previously this area was investigated by using conversation proxies coming from services such as social media and microblogging services, where conversations are the posts with comments from the users in the network. These tasks do not focus on retrieving responses rather it is focused on how the discussion unravels, either in detecting social conversational features [9] or in predicting the response rate [10].

In recent years, conversational IR gained attention in the research community in part due to advancements in the human interaction experience introduced by emerging conversational assistants, e.g., Apple Siri, Amazon Alexa, Google Now, and Microsoft Cortana. There are several faceted tasks related to this topic, with research ranging from the analysis of spoken conversational search [11] to multi-modal or mixed-initiative interaction systems [12]. Since 2019, TREC runs the Conversational Assistance Track (CAsT)[2], with the goal of establishing a robust and standard assessment process. Systems are directly compared to a common gold standard by providing multi-turn conversations and passages relevant for the utterances extracted from publicly available collections [3]. The primary focus of the new CAsT initiative is (i) system understanding of information needs in a conversational format, and (ii) finding relevant passages leveraging conversational context. The long-term vision of CAsT is focusing on progressively more complex tasks requiring multiple interactions of the user with the system.

[2] `http://www.treccast.ai/`

Current approaches to conversational IR include techniques belonging to different research fields. In the following, we discuss some recent advances in two research fields, namely query/utterance rewriting and query classification.

Query/Utterance Rewriting. Query rewriting is central in modern search as it better models the information need of the user and enhances retrieval effectiveness. Research on query rewriting techniques for IR has a long history, pioneered by the techniques for automatic query expansion based on pseudo-relevance feedback, first deployed by the SMART system in TREC-3 [13]. In conversational IR, similar challenges arise, since utterances, like queries, may be ambiguous or not-well formulated. Conversational utterance rewriting aims to reformulate a concise turn in a conversational context to a fully specified, context-independent query that can be effectively handled by information retrieval systems.

In conversational IR, utterance rewriting is investigated by focusing on effective strategies for handling linguistic characteristics of human dialogues, such as anaphora (words that explicitly reference previous conversation turns) and ellipsis (one or more words that are omitted from the conversation and nevertheless understood in the context of the remaining elements of the conversation). Conversational IR systems require mechanisms capable of resolving contextual dependencies to correctly interpret the evolution of the conversation, and utterance rewriting techniques are key in solving this task. While existing co-reference resolution tools are designed to handle anaphoras, they do not provide any support for elliptical constructions. Yang *et al.* propose the use of neural matching models for the next question prediction task in conversations [14]. Experiments on public data show that neural matching models perform well on this kind of task. Mele *et al.* show that the addition of topic-related information when an explicit reference to the conversational topic is missing is crucial for the effectiveness of conversational IR systems. Moreover, they propose some utterance rewriting strategies leveraging topic information extracted from the first utterance or from previous utterances containing cues or noun chunks. Their experiments show that such rewriting strategies obtain higher effectiveness than rewriting strategies based on co-referencing [1].

Ren *et al.* propose the exploitation of a sequence-to-sequence model for context-aware rewriting of conversational queries [15], while Aliannejadi *et al.* illustrate a dataset and a methodology to identify the utterances relevant to a given turn by introducing a neural utterance relevance model based on BERT [16]. Yu *et al.* presents a few-shot generative approach to conversational query rewriting [17]. The authors develop two methods, based on rules and self-supervised learning, to generate weak supervision data using large amounts of ad-hoc search sessions. These data are used to fine-tune GPT-2 to rewrite conversational queries. They show that on the TREC Conversational Assistance Track, the weakly supervised GPT-2 rewriter improves the state-of-the-art ranking accuracy by 12%, by only using a limited number of manual query rewrites. Their experiments show that GPT-2 effectively learns to capture context dependencies, even for hard cases involving long-turn dependencies.

Other researchers tackle the problem of conversational question answering. Vakulenko *et al.* [18] propose a question-rewriting technique able to translate ambiguous requests into semantically equivalent unambiguous questions. The solution is applied to two different question-answering architectures for passage retrieval and for answer extraction from passages. The question-rewriting model is based on a unidirectional Transformer decoder whose input is the (potentially ambiguous) current question plus five previous conversation turns. The model is trained using a set of ground-truth questions manually rewritten by human annotators and is based on a teacher-forcing approach. Given a sequence of previous tokens, the model tries to predict the next token in an output sequence. This approach relies on *manually rewritten* utterances, while we use *manually annotated* utterances where labels help to define the best source from which the ambiguous utterance should be enriched. In a follow-up work [19], Vakulenko *et al.* present an extensive comparison of different question-rewriting methods on the CAsT 2019 and 2020 datasets. They compare original user questions and human rewritten questions against questions automatically rewritten by sequence-generation models based on GPT-2 and by a term-classification method, called *QuReTeC*. Given the conversation, the approach applies BERT to predict the terms that should be added to the current question. The authors also prove that combining different models can improve the performance. In particular, simply appending the terms predicted by QuReTeC to the questions rewritten by a sequence-generation model allows to improve the state-of-the-art ranking performance. Differing from our work, the question-rewriting approaches discussed are based on sequence-generation models and term classification, while we use heuristics for general utterance rewriting.

A different line of research focuses on improving result ranking by incorporating external knowledge. Yang *et al.* propose a learning framework on top of deep neural matching networks that leverages external knowledge for response ranking in information-seeking conversation systems [20]. Information extracted from the pseudo relevance feedback documents of the candidate responses is first used to enrich the original candidate representations. Then, "correspondence" patterns between question and answer terms in external question-answer pairs are incorporated into deep matching networks to help response selection. Experiments on three conversational datasets show that the proposed approach outperforms several baselines.

Work resembling our approach is Voskarides *et al.* [21], where query rewriting for conversational search is modeled as a binary term classification task. For each term in previous turns of the conversation, a classifier decides whether to add it to the current query or not. The model encodes the conversation history and the current query using BERT and uses a term classification layer to predict a binary label for each term in the conversation history. Differing from this work, we design effective classifiers to detect the utterance context and choose the best rewriting strategy to adopt for the utterance to improve the conversation IR system effectiveness.

Query Classification. Early query classification efforts ground in Web mining research, where the characterization of the nature of the queries is an important task for understanding user behavior and improving web search effectiveness. Query logs are analyzed to extract useful information about user interactions, and query classification is exploited for several tasks such as query recommendation [22, 23, 24], query clustering [25], document ranking [26], and session segmentation [27, 28, 29]. Pioneering work on query classification is Lau and Horvitz [30], where the authors study a manually-labeled log from the Excite search engine and classify query reformulation types with the final aim of building a Bayesian model of the user behavior that takes into account also temporal information.

Rie and Xie explicitly target reformulation patterns in user search sessions [31]. In their studies, they manually study 313 search missions and introduce a taxonomy for query classification/reformulation. In 2004 through 2007, to improve retrieval accuracy and efficiency, a week-long, hundreds of millions of queries, general-purpose, commercial web log was used to conduct a variety of query classification and other analysis studies [32, 33, 34, 35]. Resulting topic routing and caching strategies improved efficiency, while query disambiguation algorithms increased accuracy. In 2007, Jansen *et al.* propose an analysis of 1.5 million query reformulations from a large commercial query log [36, 37]. They introduce automatic query classification by exploiting a previous approach by He et al. [38], where the reformulations are identified with a set of pre-defined rules. This classification employs 6 features that model term differences between a pair of queries.

Later, Boldi *et al.* revisit the taxonomy introduced by Rie and Xie by introducing a coarser granularity for reformulations that, although part of the same search mission, are not just simple direct reformulations of the previous query [39, 40]. They provide a new model for query reformulation that involves "generalization" and "specialization" plus two new classes, i.e., "correction, and "parallel move". Moreover, they also extend the previous approach by Jensen by learning from labeled data a query reformulation classifier based on 27 features. The authors show that by exploiting a mix of several signals (cosine similarity, query length, n-grams, etc) they can discover unexpected reformulations like from "dango" to "Japanese cakes" while being able to cover simpler cases that are also modeled by previous approaches.

Like Boldi *et al.* we also train query/utterance classifiers to explicitly model the contextual dependency of a utterance from the whole conversation. We use the result of the classification to feed adaptive utterance rewriting strategies considering the evolution of the multi-turn dialogue of the user with the system.

3. Conversation Understanding Architecture

We now provide formal definitions and describe our conversational understanding architecture. A summary of the notation used herein is summarized in Table 1.

Given a conversation, let u_i denote the current utterance, while $u_1, \ldots, u_{i-1}$ denote the previous utterances of the same conversation. Moreover, we use the term *context* to indicate the theme of an utterance in the multi-turn conversation. Our conversational IR system receives as input utterances. Given the current and previous utterances of the same conversation, its goal is to retrieve a ranked list of documents from a document collection.

To retrieve highly relevant documents, our utterance understanding approach exploits the conversation context in any utterance since natural language utterances might not be self-explanatory. Indeed, if the utterances are used directly as queries, they can be ambiguous or too generic. The system must thus automatically transform the utterances into *queries*, i.e., sequences of keywords suitable for processing and answered by an IR system. As illustrated in the upper half of Figure 1, our conversational system is structured as a three-step cascade architecture, namely *utterance understanding*, *first-stage top retrieval*, and *second-stage neural reranking* [41, 42, 43]. First, the utterance understanding component transforms the input utterance into a suitable query. Second, the first-stage top retrieval component processes the input query and retrieves a list of top K_1 documents from a document collection D. Third, the second-stage neural reranker reorders the K_1 documents received in input to output the final ranked list of $K_2 \ll K_1$ documents.

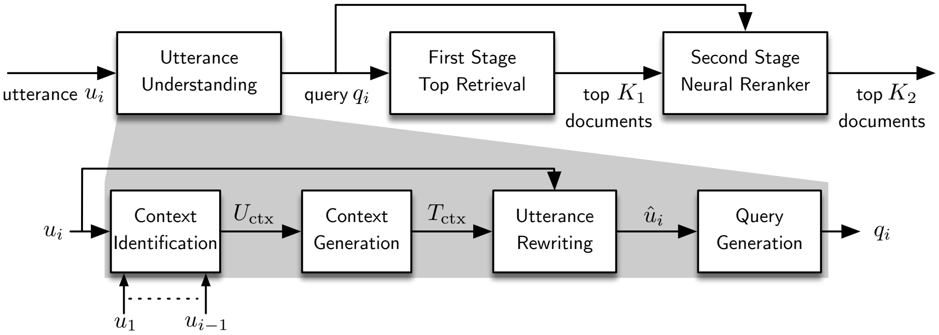

Figure 1: Conversation understanding architecture.

The *utterance understanding* component is our main focus. It transforms the current utterance u_i expressed in natural language into a structured query q_i expressed in an IR-specific query language such as the Indri [44] or Terrier [45] query languages. The conversational search utterances express different facets of the user information need. An utterance can be self-explanatory or vague; it can explore several aspects regarding an implicit topic, can be a specialization, a paraphase, and even a topic switch. All these nuances are hard to capture and even harder to resolve [15]. For such reasons, we propose different utterance understanding techniques to improve the expressiveness of queries. Our utterance understanding strategies transform a potentially vague and ambiguous conversational utterance into a self-explanatory query that can be effectively processed by an IR system to provide highly relevant results to conversational searches.

To address the problem of utterance understanding, i.e., to detect the context of an utterance and leverage it to generate a query for an IR system, we propose the components depicted in the lower-half expansion of Figure 1. The input utterance u_i is provided to a *context identification* component. This component receives as additional input the previous utterances $u_1, \ldots, u_{i-1}$ of the same conversation, and it returns a subset U_{ctx} of the previous utterances considered relevant to identify the context of u_i. The relevant utterances

Table 1: Notation.

Symbol	Definition
D	document collection
K_1	number of documents returned by the first stage top retrieval module
K_2	number of documents returned by the second stage neural reranker module
u_i	utterance at turn i of the current conversation
y_i	classification label of the utterance at turn i of the current conversation
q_i	query at turn i of the current conversation
U_{ctx}	set of the utterances together with their context classification information
c_i	context terms extracted from the utterance u_i
T_{ctx}	set of the context terms together with their context classification information
$\hat{u}_i$	utterance at turn i enriched with context information
$\hat{c}_i$	context terms extracted from the utterance context-enriched utterance u_i
SE	classification label for utterances that are Self Explanatory
FT	classification label for utterances referring to the First Topic in the conversation
PT	classification label for utterances referring to a Previous Topic in the conversation different from the first one

included in U_{ctx} are identified by means of specific utterance classifiers as discussed in Section 3.1.

The selected utterances are then provided to a *context generator* component (see Section 3.2), responsible to extract from them the keywords needed to enrich u_i with the missing information. For example, this information can include noun chunks and/or named entities. The output of this component is a set T_{ctx} of keywords to be used by the *utterance rewriting* component (see Section 3.3) to generate a new enriched utterance $\hat{u}_i$. Note that the U_{ctx} and T_{ctx} sets can be empty if the utterance u_i is self-explanatory and its context is already completely specified in the utterance. The rewritten utterance is then processed by the *query generator* (see Section 3.3), responsible to generate the query to be submitted to the first stage retrieval component. For example, the query generator can include some preprocessing such as stopword removal, stemming and special phrase operator generators [46, 47].

3.1. Context Identification

The identification of an utterance's context can be challenging due to a plethora of different problems, such as back references to topic/entities mentioned in previous utterances, i.e., zero anaphoras, or context changes, i.e., topic shifts. We focus on these problems of lack or change of context by investigating novel ways to identify the context – possibly missing – of an utterance, given the previous utterances in the same

Table 2: Example of a multi-turn conversation with their classification label.

Turn	Utterance	Label
1	Is Red Bull bad for you?	SE
2	Can it kill you?	FT
3	How much can you drink in a day?	FT
4	What is taurine?	SE
5	What are its health effects?	PT
6	In general, what are the effects of consuming energy drinks?	SE
7	Why are they harmful when mixed with alcohol?	PT
8	What is the argument for their age restriction to kids?	PT
9	Where are they banned to minors?	PT

conversation.

The context identification component receives as input the current utterance u_i along with the sequence of previous utterances of the conversation $u_1, \ldots, u_{i-1}$. Its goal is to understand if u_i is self explanatory or needs to be enriched with context extracted from the previous utterances in the conversation. Hence, the context identification component identifies a set of the utterances together with their context classification information U_{ctx} necessary to provide the contextual information for rewriting u_i (see Alg. 1). The core of the component is the Classify function (line 1) taking as input a utterance and the previous utterance class pairs (if any) and returning three main classes of utterance that are then used to specify the content added to U_{ctx} (lines 3, 5 and 7).

By carefully inspecting the utterances in the publicly available CAsT dataset, we notice that the conversations exhibit some common characteristics that can be exploited to identify the context of a generic utterance u_i. An example of multi-turn conversation and of the kinds of utterances observable in the conversational dataset is reported in Table 2. We refer to the exemplary utterances in this conversation to introduce the three different cases for utterance context identification discussed below:

- The first utterance u_1 in all conversations is Self Explanatory (SE) since it always contains an explicit mention to the initial topic of the new conversation. In some cases, these utterances include acronyms difficult to resolve, e.g., "What is nominal GDP?", or refer to a rare topic, e.g.,"What is paleo?" Anyway, they always contain all the context needed to understand the information need of the user. The turn 1 utterance in Table 2 exemplary illustrates this case recurring in all the conversations of the dataset. Even utterances in positions different from the first one can be SE. We observe that quite often SE utterances appearing in the middle of a conversation introduce a change in the conversation's topic.

Algorithm 1: The context identification algorithm

```
Input  : A list $C$ of $i-1$ (utterance, class) pairs
         The current utterance $u_i$
Output: A list $U_{\mathsf{ctx}}$ of 1, 2, or $i$ (utterance, class) pairs
ContextIdentification($C$):
    switch Classify($u_i, C$) do
        case SE do
            $U_{\mathsf{ctx}} \leftarrow [(u_i, \texttt{SE})]$
        case FT do
            $U_{\mathsf{ctx}} \leftarrow [(u_1, \texttt{SE}), (u_i, \texttt{FT})]$
        case PT do
            $U_{\mathsf{ctx}} \leftarrow C + [(u_i, \texttt{PT})]$
    return $U_{\mathsf{ctx}}$
```

In our conversation example in Table 2, we have two cases of this kind: turn 4, where we observe the mention to "taurine", a new topic introduced in the conversation, and in turn 6, where we encounter another topic drift: "taurine" → "energy drinks". Independent of their position in the multi-turn conversation, SE utterances do not need to be enriched with context extracted from other utterances. Thus, for all of them, the context identification component simply outputs $U_{\mathsf{ctx}} = \{(u_i, \texttt{SE})\}$.

- The First Topic (FT) in the conversation often dominates its context. Many utterances are in fact not SE but refer implicitly to the topic introduced by the first utterance of the conversation. In our conversation example in Table 2, this case is represented by the utterance at turn 2 which carries as context the first utterance of the conversation. For all FT utterances, the context identification component outputs $U_{\mathsf{ctx}} = \{(u_1, \texttt{SE}), (u_i, \texttt{FT})\}$.

- The last case is when u_i implicitly refers to a Previous Topic (PT) mentioned in an utterance different from the first one. In our conversation in Table 2, we observe PT utterances at turns 5, 7, 8, and 9. These utterances only become understandable if enriched with context extracted from utterances 4 and 6. We experiment with different strategies for rewriting PT utterances (see Section 3.3). Thus, when our context identification component encounters a PT utterance, it outputs the sequence of all previous and current utterances along with their labels $U_{\mathsf{ctx}} = \{(u_1, y_1), \ldots, (u_{i-1}, y_{i-1}), (u_i, \texttt{PT})\}$, with $y_1, \ldots, y_{i-1} \in \{\texttt{SE}, \texttt{FT}, \texttt{PT}\}$.

This classification is used in our utterance understanding component to select the adaptive strategies for utterance rewriting.

Algorithm 2: The context generation algorithm

```
   Input  : A list $U_{\mathsf{ctx}}$ of 1, 2, or $i$ (utterance, class) pairs
   Output: A list $T_{\mathsf{ctx}}$ of 0, 1, or $i-1$ (context, class) pairs
   ContextGeneration($U_{\mathsf{ctx}}$):
1      switch $y_i$ do
2          case SE do
3              $T_{\mathsf{ctx}} \leftarrow []$
4          case FT do
5              $c_1 \leftarrow$ Extract($u_1$)
6              $T_{\mathsf{ctx}} \leftarrow [(c_1, \mathtt{SE})]$
7          case PT do
8              $T_{\mathsf{ctx}} \leftarrow []$
               for $k \leftarrow 0$ to $i-1$ do
9                  $c_k \leftarrow$ Extract($u_k$)
10                 $T_{\mathsf{ctx}} \leftarrow T_{\mathsf{ctx}} + [(c_k, y_k)]$
11     return $T_{\mathsf{ctx}}$
```

3.2. Context Generation

The context generation component (see Alg. 2) receives as input U_{ctx}, containing the contextual information required to understand the information need expressed in utterance u_i, as explained in Section 3.1. The component extracts the context information by analyzing the utterances in U_{ctx} to generate the context T_{ctx}, which in turn is passed to the utterance rewriting component detailed in Alg. 3. More formally, given an utterance u_i, we denote with c_i the context extracted from the utterance. Such context consists of a list of tokens extracted from the utterance with the help of the context generation component, i.e., the Extract function (lines 5, 9). Every context c_i is paired with the corresponding utterance label y_i, and hence T_{ctx} is defined as a list of (c_i, y_i) pairs depending on U_{ctx}. Note that the labels $y_1, \ldots, y_i$ will be used to apply different rewriting strategies in the utterance rewriting component.

Part-of-speech tagging, named entity resolution, dependency parsing and co-reference resolution are the linguistic analysis components which help identify significant pieces of text and which we exploit in our context generation strategies to compute the different contexts. We extract various linguistic features from the utterances[3]. The context generation component offers different methodologies, each one exploiting a distinct set of linguistic features:

- *Context extraction* uses dependency parsing for identifying significant noun chunks, e.g., objects or subjects different from pronouns. For example, given the utterance "*Is Red Bull bad for you?*", this

[3] By using the SPACY library available at `https://spacy.io/usage/linguistic-features`.

method returns the noun chunk "*Red Bull*".

- *Context on cue* is based on cues (e.g., "*tell me about*") and tries to capture a context change that might be significant for the conversation from that point on. It is based on dependency parsing for extracting the *current context* from the utterance. For example, given the utterance "*Tell me about taurine.*", this method returns "*taurine*".
- *Context binder* captures all previous noun chunks, different from pronouns, which are either subjects or objects, to enrich the context of the conversation up to the current turn. For example, given the utterance at Turn 5 "*What are its health effects?*", this method returns a set of noun chunks from previous utterances"{*Red Bull, drink in a day, taurine*}". With conversation progression, context increases. This brings benefits in case the conversation is mono-thematic (high recall), but can lead to lower precision when wanting to answer a very specific question or when a context drift occurs.

Independent of the specific context generation methodology adopted, the output resulting from this component T_{ctx} transforms the input U_{ctx}, by extracting from each utterance the context keywords, and passes them to the next component, utterance rewriting, together with the label, as follows:

- if $U_{\mathsf{ctx}} = \{(u_i, \mathtt{SE})\}$ then $T_{\mathsf{ctx}} = \{\}$, meaning no context information if u_i is self explanatory (SE);
- if $U_{\mathsf{ctx}} = \{(u_1, \mathtt{SE}), (u_i, \mathtt{FT})\}$ then $T_{\mathsf{ctx}} = \{(c_1, \mathtt{SE})\}$, meaning the context information is extracted from the first utterance u_1 since the current utterance u_i is labeled as first topic (FT);
- if $U_{\mathsf{ctx}} = \{(u_1, y_1), \ldots, (u_{i-1}, y_{i-1}), (u_i, \mathtt{PT})\}$ then $T_{\mathsf{ctx}} = \{(c_1, y_1), \ldots, (c_{i-1}, y_{i-1})\}$, meaning that the context information is extracted from all the previous utterances $u_1, \ldots, u_{i-1}$ if the current utterance u_i is labeled as previous topic (PT).

3.3. Utterance Rewriting

The utterance rewriting component (see Alg. 3) takes as input the current utterance u_i and T_{ctx}, the context keywords produced by the previously explained component, and rewrites the current utterance u_i according to one of the utterance rewriting strategies presented below. Our strategies exploit the utterance classification provided by the context identification component to select the best source of context depending on the class. The output is an enriched utterance $\hat{u}_i$ with the augmented context terms and can be converted in a query for the information retrieval system by the query generator component.

Given an utterance classified as self-explanatory (SE), the utterance rewriting component receives as input $T_{\mathsf{ctx}} = []$, meaning that the current utterance u_i does not need any additional context (lines 1, 2). Otherwise, the original utterance u_i must be enriched with context extracted by the context extraction component. We propose different rewriting strategies for the propagation of context.

1. *Standard*: if u_i is classified as FT, then it is enriched with context extracted from the first utterance, i.e., c_1 (line 8). Consider, for example, the conversation shown in Table 2. The second utterance is not self-explanatory and is classified as FT. As a consequence, "Can *it* kill you?" becomes "Can *Red Bull* kill you?". If u_i is classified as PT, then the context is extracted from the previous utterance, i.e., c_{i-1} (line 10). For example, the fifth utterance "What are *its* health effects?" which is classified as PT becomes "What are *taurine* health effects?" as it is enriched with the topic from the previous utterance (i.e., the fourth).
2. *Enriched*: similar to (1), but if u_i is classified as PT, then the context is extracted from the previous enriched utterance $\hat{u}_{i-1}$, namely $\hat{c}_{i-1}$ (line 15). As an example, the sixth utterance from Table 2 "What is the argument for *their* age restriction to kids?" is classified as PT. It is rewritten using the enriched previous utterance, namely, "Why are *energy drinks* harmful when mixed with alcohol?" So, it becomes "What is the argument for *energy drinks* age restriction to kids? *harmful mixed alcohol*."
3. *Last* SE: this method always propagates the context extracted from the last seen SE utterance (line 17), regardless of the current utterance label (FT or PT), i.e., c_j with $j = \max\{k : 0 < k < i \ \wedge \ y_k = \texttt{SE}\}$ (line 5). Note that the last seen SE utterance is not necessarily the utterance of the previous turn, i.e., u_{i-1}. As an example, "Where are *they* banned to minors?" becomes "Where are *energy drinks* banned to minors?"
4. *First and last* SE: Similar to (3), but the rewriting strategy uses context from both the last seen SE utterance and from the first utterance, i.e., c_1 and c_j, with c_j defined as in (3) (line 19). For instance, the utterance "Where are *they* banned to minors?" becomes "Where are *energy drinks* banned to minors? *Red Bull*" as the topic extracted from the first utterance is always added to the question with the purpose of enriching its context.
5. *First or last* SE: If u_i is classified as FT, it is enriched with the context extracted from the first utterance, i.e., c_1 (line 22). If u_i is classified as PT, then the context is extracted from the last seen SE instead of the previous utterance, i.e., c_j, with c_j defined as in (3) (line 24). For instance, the utterance "Where are *they* banned to minors?"becomes "Where are *energy drinks* banned to minors?" As *energy drinks* is the last seen topic.

The Resolve function takes as input a utterance and a context (line 25). The output of this function is a rewritten utterance $\hat{u}_i$ which includes the proper context in case of implicit or explicit references. In detail, if the utterance contains a third-person pronoun, it is replaced with the context, otherwise the context is concatenated at the end of the utterance. The transformed utterance is then passed to the query generation component that converts it into a query interpretable by a search engine such as a classical or neural IR system. As preprocessing, the IR system will perform standard operations such as lower-casing, stemming, and stopword removal.

Algorithm 3: The utterance rewriting algorithm

```
Input  : The current utterance $u_i$
         The previous rewritten utterance $\hat{u}_{i-1}$
         A list $T_{\mathsf{ctx}}$ of 0, 1, or $i-1$ (context, class) pairs
         The rewriting method $m$
Output: The rewritten utterance $\hat{u}_i$
UtteranceRewriting($T_{\mathsf{ctx}}, \hat{u}_{i-1}, u_i, m$):
    if len($T_{\mathsf{ctx}}$) $= 0$ then
        // $y_i$ is SE
        ctx $\leftarrow$ []
    else
        // $y_i$ is FT or PT
        switch $m$ do
            $j \leftarrow \max\{k : (u_k, \texttt{SE})$ in $T_{\mathsf{ctx}}\}$
            case Standard do
                if $y_i = \texttt{FT}$ then
                    ctx $\leftarrow c_1$
                else
                    ctx $\leftarrow c_{i-1}$
            case Enriched do
                if $y_i = \texttt{FT}$ then
                    ctx $\leftarrow c_1$
                else
                    ctx $\leftarrow$ Extract($\hat{u}_{i-1}$)
            case Last SE do
                ctx $\leftarrow c_j$
            case First and last SE do
                ctx $\leftarrow c_1 + c_j$
            case First or last SE do
                if $y_i = \texttt{FT}$ then
                    ctx $\leftarrow c_1$
                else
                    ctx $\leftarrow c_j$
    $\hat{u}_i \leftarrow$ Resolve($u_i$, ctx)
    return $\hat{u}_i$
```

4. Utterance Classification

We now present the training and evaluation process of the utterance classifiers for context identification. First, we present the datasets employed and the labeling process used to develop a golden truth for the classifiers. Then, we report on the performance of different approaches for our utterance classification task.

4.1. Datasets and Labeling Process

We exploit two public and well-established conversational IR datasets to train the context identification classifiers and assess the performance of our proposed conversational understanding architecture, namely the *CAsT 2019* dataset and the *ConvQuestions* dataset [48]:

- *CAsT 2019 Dataset.* The TREC Conversational Assistant Track (CAsT) 2019[4] provides a dataset with 80 multi-turn conversations, having each from 8 to 12 utterances (748 utterances in total). Relevance judgments, graded on a three-point scale (i.e., 2 very relevant, 1 relevant, and 0 irrelevant), are provided for 20 conversations including 194 out of the 748 utterances. The judgements refer to utterance-passage pairs where the passages are extracted from documents in three collections: (1) TREC CAR (Complex Answer Retrieval), containing $\sim 29M$ passages extracted from $\sim 5M$ Wikipedia articles, (2) MS-MARCO (MAchine Reading COmprehension), composed by $\sim 8M$ passages from answer candidates of the Bing search engine, and (3) the WAPO (WAshington POst) dataset, consisting of $\sim 8M$ passages extracted from $\sim 608K$ news articles. This dataset is used for both classification training and conversational search performance assessment.

- *ConvQuestions Dataset.* Since the CAsT dataset does not provide enough utterances to train effective classifiers for our context identification task, we used also other utterances from the *ConvQuestions* (ConvQ) dataset [48], consisting of 350 conversations. Among these conversations, we selected 214 conversations for a total of $1,010$ utterances that provide relevant examples of our classification classes for context identification. Notice that these utterances are used only for enriching the training set for the classification task. We do not use these conversations for testing the performance of our conversational IR solution, as utterance-passage relevance judgements are missing for them.

The utterances in the CAsT and ConvQ datasets were manually labeled by 5 annotators according to our 3 classification classes (`SE`, `FT`, `PT`)[5]. The human annotators are researchers and PhD students with a strong background in information retrieval. We measured the inter-annotator agreement with the Fleiss' Kappa [49] indicator that provides us with a consistency measure of the assessors' ratings. It is computed as

[4] http://www.treccast.ai/

[5] To favor reproducibility and further developments, we release the annotated dataset and the source code implementing our solution at the following link: `https://github.com/hpclab/adaptive-utterance-rewriting-conversational-search`.

$\kappa = \frac{\bar{P}-\bar{P}_e}{1-\bar{P}_e}$, where $1-\bar{P}_e$ gives the degree of agreement that is attainable above chance, and $\bar{P}-\bar{P}_e$ gives the degree of agreement actually achieved above chance. If $\kappa = 1$ the agreement is complete, while $\kappa \leq 0$ means no agreement. We registered a Fleiss' Kappa value of 0.945 which, according to the table for interpreting κ values provided in [50], corresponds to an "almost perfect" agreement.

4.2. Utterance Classification

The context identification component aims at understanding if an utterance is self-explanatory, i.e., SE, or it needs to be enriched with context extracted from the previous turns in the conversation, i.e., FT or PT. To automatically associate one of the three class labels to the utterances processed by our conversational search architecture, we develop different classifiers that approach the task by performing either a cascade of two *binary* classification steps or one *multilabel* classification, as depicted in Figure 2.

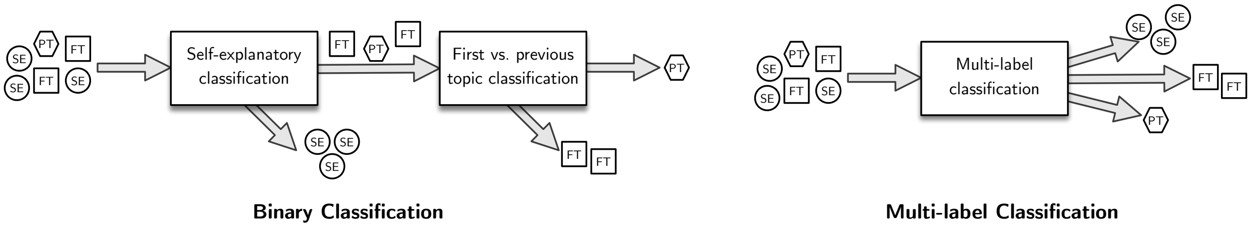
PT
FT
SE
SE
FT
SE
Self-explanatory classification
FT
PT
FT
First vs. previous topic classification
PT
SE
SE
SE
FT
FT
Binary Classification
PT
FT
SE
SE
FT
SE
Multi-label classification
SE
SE
SE
FT
FT
PT
Multi-label Classification

Figure 2: Binary classification cascade (left) and multi-label classification (right).

In the first approach, utterance classification is implemented as a cascade of two stages: the first stage distinguishes between SE and non-SE utterances, where the former (SE) does not need any additional context, while the latter (non-SE) misses some context and needs enrichment. For those utterances that are classified as non-SE, the second stage determines if the utterance is FT, i.e., it needs context from the first utterance, usually introducing the general topic of the conversation, or PT, i.e., it needs context from a previous utterance different from the first one. In the multi-label classification, there is only one classifier, simply getting as input an utterance and returning one of the three labels, i.e., SE, FT, PT.

We follow a standard train/test methodology by splitting the manually labeled utterances into two sets: the training set for learning the classifier and the test set for checking its predictive performance. More specifically, we use all the labeled utterances from the ConvQ dataset and 554 utterances without relevance assessments from the CAsT dataset for training, while the remaining 194 test utterances with relevance assessments from CAsT are used for testing the classification accuracy. As detailed in the following, we tackle our utterance classification task by training classifiers based on gradient boosting decision trees [51] and on the BERT [52] bidirectional transformer neural network.

GBDT classifiers. We first perform the classification employing Gradient-Boosted Decision Trees (GDBT), a non-parametric supervised learning method [51]. Gradient boosting is a broad concept that identifies a group of machine learning algorithms that combine many weak learners together to create a strong predictive model. The goal is to create a model that predicts the class of an utterance by learning decision trees inferred from a dense feature representation of utterances and labels. For training the classifiers, we specifically use the LightGBM[6] implementation of GDBT.

Features. The input representation for GBDT classifiers is based on two sets of *sentence*-level and *conversation*-level hand-crafted features. The complete list of features is reported in Tables 3 and 4. Named entities are detected using both the Spacy[7] and TagMe[8] toolkits. The set of features listed in Table 3 is used for the binary classifier aimed at deciding whether the utterance is `SE` or not. For this task, we build a rich set of sentence-level features, with the help of the linguistic features provided by Spacy, enabling the model to discriminate between utterances that lack context or those that contain all the elements for generating a possibly successful query. For example, the presence of third person pronouns is an indication that the utterances are not self-explanatory.

The set of conversation-level features listed in Table 4 is instead designed to model the relation of the current utterance with the previous utterances in the conversation. These features are used by the second classifier of the classification cascade aimed at distinguishing between `FT` and `PT` utterances. We look at the depth of the current utterance within the conversation (i.e., turn) and how distant it is from the last `SE`, either immediately after or several turns after. Another group of features are referring to the likelihood of the current utterance to be the next after the first utterance or the previous utterance, in other words we would like to answer the question: is the current utterance more likely to follow the first or the previous utterance. For extracting these features we use the next sentence prediction BERT-base model, and give in input two pairs, the current with the first utterance or the current with the previous utterance. We use the model available in the HuggingFace library[9]. The last four features in Table 4 consider the cosine similarity between the current utterance and the first or previous utterances in the conversation, either considering the full-text utterances or only the noun chunks. The cosine similarity is computed on sentence transformers embeddings from the RoBERTa model[10].

BERT classifiers. We employ the BERT bidirectional transformer network [52] to perform utterance classification as a cascade of two binary classifiers (Figure 2 (left)). The BERT model we use is fine-tuned on MS-MARCO and is publicly available as part of the Hugging Face transformers library[11]. We train the `SE` vs. non-`SE` classification layer on top of this BERT model by using binary cross-entropy loss and a sigmoid

[6] `https://github.com/microsoft/LightGBM`

[7] `https://spacy.io/usage/linguistic-features`

[8] `https://tagme.d4science.org/tagme/`

[9] `https://huggingface.co/transformers/model_doc/bert.html#bertfornextsentenceprediction`

[10] `https://huggingface.co/sentence-transformers/roberta-large-nli-stsb-mean-tokens`

[11] `https://huggingface.co/nboost/pt-bert-base-uncased-msmarco`

Table 3: Sentence-level features used for the SE vs. non-SE GBDT classifier.

Numerical features	Binary features
Utterance length	Is it a complete sentence?
Number of tokens	Is there a question mark?
Number of noun chunks	Is it a what phrase (e.g., “what is”)?
Number of question words (e.g., what, where)	Is there a question word (e.g., what, where)?
Number of question phrases (e.g., “how much”)?	Is there a question phrase (e.g., “how many”)?
Number of named entities (detected by Spacy)	Is there a named entity (detected by Spacy)?
Number of named entities (detected by TagMe)	Is there a named entity (detected by TagMe)?
Number of nouns	Is there a noun?
Number of adjectives	Is there an adjective?
Number of comparative adjectives	Is there a comparative adjective?
Number of adverbs	Is there an adverb?
Number of comparative adverbs	Is there a comparative adverb?
Number of pronouns	Is there a pronoun?
Number of 3rd person pronouns	Is there a 3rd person pronoun?
	Is there a cue phrase (e.g., “tell me about”)?

activation on the output. The classifier is trained for 5 epochs by employing a learning rate of 5×10^{-5}. We also use a validation set to stop the training process whenever no more gain over the loss is observed. The validation set is generated by randomly sampling 20% of the utterances in the training set.

The FT vs. PT classifier is trained in two different ways. In the first implementation, we employ the same methodology used for training the SE vs. non-SE, but differently from the previous case, here we exploit as training examples only the utterances that are not labeled as SE in the ground truth, leaving the training set with 1,096 utterances. In the second implementation, FT vs PT classification exploits the available context given by all the previous utterances of the conversation. In this case, we use BERT for a sentence pair classification task, where one sentence is the context built as a concatenation of all the previous utterances of the conversation, and the second sentence is the current utterance to be classified as FT or PT. We call this classifier “BERT with Context” ($\text{BERT}_{\texttt{ctx}}$). As before, we exploit 1,096 utterances for training the network and randomly sample 20% of the training set to be used as validation set during training. The best performance of the classifier on the validation set is observed after 6 epochs using a learning rate of 1×10^{-4}.

BERT multi-label classification. We also employ the same BERT model fine-tuned on MS-MARCO to perform multi-label utterance classification. Unlike the previous cascade of BERT binary classifiers, we train one single classifier that, given an utterance, provides the class that the utterance most likely belongs to (Fig-

Table 4: Conversation-level features used for the FT vs. PT GBDT classifier

Description	Type
Turn in the conversation	Numerical
Is it after a SE utterance?	Binary
Distance from the last SE utterance	Numerical
Likelihood after first utterance	Numerical
Likelihood after previous utterance	Numerical
Cosine similarity with the first utterance	Numerical
Cosine similarity with the previous utterance	Numerical
Cosine similarity with the first utterance (using only noun chunks)	Numerical
Cosine similarity with the previous utterance (using only noun chunks)	Numerical

ure 2 (right)). To do so, we train a multi-label classifier on top of BERT by using a categorical cross-entropy loss and a softmax activation as an output layer. We call this classifier "Multilabel BERT" ($\text{BERT}_{\mathsf{mlt}}$). We employ in this case all the 1,564 labeled utterances as training set and the remaining 194 utterances as test set. We train the classifier for 51 epochs by employing a learning rate of 5×10^{-5}. As in the previous case, we also use 20% of the training set as validation set to perform early stopping of the training process.

4.3. Classification Results

We experimentally evaluated the utterance classifiers discussed above. Table 5 reports the classification performance in terms of F1 score and weighted F1 score. Moreover, for each classifier, we also report the number of instances (support) for the three classes in the test set, i.e., SE, FT, and PT, and the number of misclassification errors for each class.

The GBDT and BERT classifiers work in a cascade, i.e., the first binary classifier discriminates between SE and non-SE utterances, and those classified as non-SE are then provided to the second stage of the cascade for FT vs PT classification. To avoid bias, we report an evaluation in isolation, where the evaluation of the performance of the second classifier is not biased by the misclassifications of the first binary classifier. The SE vs non-SE classifiers are assessed on all the 194 utterances in the CAsT test set. On the other hand, the FT vs PT classifiers are evaluated on the subset of utterances in the test set labeled as FT or PT, i.e., 126 utterances.

SE *vs. non-*SE *classification.* When looking at Table 5 we first observe that the classifiers predicting the SE class are in general the best performing ones. Specifically, for both the GBDT and BERT cascades, the results confirm that the SE vs non-SE classification task is easier task than the FT vs PT one. The highest performance for the first element of the cascade is achieved by the BERT classifier that achieves F1 scores of 0.87 and 0.93 on the SE and non-SE classes, respectively. The weighted F1 score of the BERT SE vs non-SE classifier is 0.91. Here, the overall performance increases due to the low number of wrong predictions recorded for the non-SE class, i.e., 10 instances out of 126 (7.9%). This result confirms that BERT is robust to type I errors (false positives), i.e., non-SE utterances classified as SE ones. Moreover, the BERT classifier provides also the best robustness w.r.t. type II errors (false negatives), i.e., 8 SE utterances out of 68 (11.8%) classified as non-SE. The GBDT SE vs non-SE classifier achieves lower performance than the BERT one. The number of errors increases to 20 for the SE class and 18 for the non-SE class, while the weighted F1 reaches 0.80, with a drop of 12%.

FT *vs.* PT *classification.* Regarding the second classifier of the cascade, i.e., FT vs PT, we observe a reduction of performance for both BERT and GBDT solutions. This is somehow expected as deciding whether an utterance is FT or PT is more difficult than deciding whether it is self explanatory or not. The reason for that resides in the nature of SE utterances, where the context is always clearly provided, while FT and PT utterances show an increased level of ambiguity that is reflected also in the classification results. The best performance on this task is achieved by the $\text{BERT}_{\mathsf{ctx}}$ classifier with 0.75 of weighted F1 while GBDT and BERT score 0.73 and 0.65, respectively. The error rate for $\text{BERT}_{\mathsf{ctx}}$ is significantly lower than GBDT and BERT for the FT class, i.e., only 8 misclassifications instead of 13 and 23, respectively. Regarding the PT class, $\text{BERT}_{\mathsf{ctx}}$ shows a worse performance than GBDT and BERT, as it misclassifies 23 utterances out of 57 while the former two misclassify 13 and 21 utterances, respectively. The reason why $\text{BERT}_{\mathsf{ctx}}$ and GBDT outperforms BERT is that the latter receives as input only the text of the current utterance. It thus does not have information on the previous utterances in the conversation. GBDT, however, has specific conversation-wise hand-crafted features that model some property possibly discriminating between FT and PT utterances, e.g., the propensity of observing PT utterances towards the end of a conversation rather than at the beginning. This observation lead us to the introduction of $\text{BERT}_{\mathsf{ctx}}$, a BERT-based classifier that exploits the entire conversation to better target the FT vs PT utterance classification. Results confirm the intuition as $\text{BERT}_{\mathsf{ctx}}$ outperforms, in terms of weighted F1, the two former competitors. Moreover, we recall that this assessment is performed in isolation. Thus, the second stage of the GBDT classifier could be also affected in a production scenario by wrong conversation-level features introduced by classification errors of the first stage of the classification cascade.

Classification cascade results. As previously mentioned, the results reported in Table 5 refer to an independent evaluation of the different binary classifiers designed but do not provide us with direct information about the performance of any cascade built from the combinations of the two binary classifiers. Hence, in Table 6 we present the performance of our GDBT, BERT, and $\text{BERT}_{\mathsf{ctx}}$ classifiers when integrated into a cascade classification process. Moreover, in the same table we report also the performance of the single-stage multi-label BERT classifier introduced above.

Table 5: Classification performance for the GBDT, BERT and BERT_{ctx} utterance classifiers. We report the results in terms of F1 score, weighted F1 score and the number of misclassifications for each class.

Classifier	Class	Support (%)	# misclassifications	F1	F1 (weighted)
GBDT (SE vs non-SE)	SE	68 (35%)	20	0.72	0.80
	non-SE	126 (65%)	18	0.85	
GBDT (FT vs PT)	FT	69 (55%)	13	0.77	0.73
	PT	57 (45%)	21	0.68	
BERT (SE vs non-SE)	SE	68 (35%)	8	0.87	0.91
	non-SE	126 (65%)	10	0.93	
BERT (FT vs PT)	FT	69 (55%)	23	0.68	0.65
	PT	57 (45%)	21	0.62	
BERT_{ctx} (FT vs PT)	FT	69 (55%)	8	0.88	0.75
	PT	57 (45%)	23	0.60	

As shown in Table 6, the best F1 value for SE utterances classification is obtained with the BERT classifier at the first stage of the cascade, i.e., 0.87. For the FT and PT classification, the best performance are obtained deploying the BERT_{ctx} classifier at the second stage of the cascade: 0.73 and 0.67 for FT and PT, respectively. The best performance obtained by considering the three classes are confirmed by the F1 metric weighted on all classes, namely 0.76. Compared to the best classifier cascade, the BERT_{mlt} classifier shows worse performance as we observe a lower F1 score for all the three classes, as well as on the weighted F1 score.

Our conversational IR pipeline end-to-end evaluation uses the BERT-BERT_{ctx} cascade as its performance exceeds that of the other experimented classification methods.

5. End-to-End Evaluation

We now present an end-to-end evaluation of the effectiveness of our conversational IR architecture by comparing its performance against state-of-the-art competitors.

5.1. Experimental Settings and Baselines

For first-stage retrieval we use Indri[12]. We index the three collections of the CAsT dataset with Indri by removing stopwords. We experiment with and without stemming, and we observe better results with

[12] https://www.lemurproject.org/indri.php

Table 6: Classification performance for different combinations of GDBT, BERT, and BERT$_{\mathsf{ctx}}$ classifiers into a cascade classification process, together with a single-stage multi-label BERT classifier.

Classifier		F1 per class			F1 (weighted)
Stage 1	Stage 2	SE	FT	PT	
GBDT	GBDT	0.72	0.58	0.56	0.62
BERT	GBDT	**0.87**	0.65	0.62	0.72
GBDT	BERT	0.72	0.53	0.55	0.60
BERT	BERT	**0.87**	0.61	0.59	0.70
BERT	BERT$_{\mathsf{ctx}}$	**0.87**	**0.73**	**0.67**	**0.76**
BERT$_{\mathsf{mlt}}$		0.83	0.53	0.41	0.60

the *Krovetz* stemmer. We experiment with several Indri querying methods (e.g., TF-IDF, BM25, inQuery); we achieve the best retrieval performance with the Indri language model with Dirichlet smoothing with parameter $\mu = 2,500$. On all the runs but one (hereinafter CAsT baseline), pseudo-relevance feedback (PRF) was enabled. Specifically, for PRF we use the RM3 algorithm [53] using 20 keywords taken from the top 20 results and a balanced language model mixture, i.e., $\gamma = 0.5$.

Regarding the second-stage neural re-ranker, we follow the state-of-the-art methodology based on relying on a robust contextualized language model pretrained on a large text corpus and then on performing minimal fine-tuning of the model for the considered ranking task. Despite of its simplicity, this approach was shown to provide excellent ranking effectiveness across different information retrieval tasks [54, 2, 55]. Specifically, we use the solution proposed by Nogueira and Cho [2] to re-rank the top K_1 results from the previous stage. The model fine-tunes the BERT base pre-trained model [52] for re-ranking on the MS-MARCO passage retrieval dataset. Consistent with previous results, we choose $K_1 = 200$ since smaller and larger cutoffs (50 and $1,000$ documents) provide worse end-to-end performance [56]. For each query, the top documents retrieved by the first stage are provided as input to the re-ranking step, together with the query rewritten by the utterance understanding component (see Fig. 1). The scores returned for these documents by the neural stage are then used to re-rank them and compute the end-to-end performance metrics.

We explore different settings for our utterance rewriting strategies with the purpose of finding the best methodology for the automatic enrichment of conversational utterances with the proper context. We choose some of the best combinations of utterance rewriting strategies for which we discuss our findings in Section 5.2. We compare the performance of both first-stage retrieval and end-to-end retrieval of our proposed best combinations with the following baseline methods.

- CAsT baseline: CAsT 2019 provides a simple baseline consisting of queries generated from utterances by applying stopword removal and AllenNLP co-reference resolution.

- Query: the same CAsT queries as in the CAsT baseline are processed, using PRF and the Indri language model with Dirichlet smoothing.

- First Query: given a conversation, the current query, q_i, is expanded with the first-turn query, q_1. This is done to perform a simple query rewriting (e.g., $q_1 + q_i$).

- Context Query: given a conversation, the current query is enriched with the first query and the one appearing in the previous turn (e.g., $q_1 + q_{i-1} + q_i$).

- Plain Utterance: utterances provided by CAsT which represent the original user requests, i.e., without performing any rewriting.

- CoRef$_{1,2}$: co-reference resolution finds all the linguistic expressions that refer to the same real-world entity in a natural language text. We experiment two different models for co-referencing: (1) the AllenNLP co-referencing model [4, 57], and (2) the neuralcoref model from the Transformers library[13], which are applied to the original utterances to produce the queries to process.

- First Topic [1]: given a conversation, the current utterance is expanded with the main conversation topic extracted from the first-turn utterance. We rewrite the current utterance by replacing the explicit pronouns (e.g., "Can *it* kill you?" becomes "Can *Red Bull* kill you?") as well as adding the context to the utterance in case of implicit reference (missing pronoun, e.g., "How much can you drink in a day?" will be "How much can you drink in a day? *Red Bull.*").

- Topic Shift [1]: the strategy is similar to First Topic, but it also includes a *Context on Cue* step which aims at identifying context changes on the basis of some cues. We extract the latest context from cue and add it to the current utterance covering both implicit and explicit references cases. In some cases we might decide to keep or remove the first context of the conversation, leaving just the current focus topic, if different from the first.

- Context [1]: the strategy is based on the previous two strategies First Topic and Topic Shift, but it also includes the context collected with the help of a *Context Binder* methodology applied to all the previous utterances in the conversation. The strategy tries to keep all the possible context information offered by all previous utterances. As in the previous case we might decide to either keep or remove the first context of the conversation, given by the first utterance.

[13] https://github.com/huggingface/neuralcoref

We compare all the above methods with our novel strategies discussed in Section 3.3, namely Standard, Enriched, Last SE, First and last SE, First or last SE. These rewriting techniques were used in combination with the utterance labels automatically generated by the cascade of BERT binary classifiers (described in Sec. 4.2) as it yields the highest classification performance.

Moreover, for the evaluation purposes we introduce three other supervised strategies based on some human support. More specifically, two of these strategies, GT Last SE and GT First or last SE are analogous to Last SE and First or last SE, but rely on the golden standard used for training the classifiers instead of labels from the automatic classifier. These two supervised strategies allow us to understand how much the classification errors may jeopardize the quality of the proposed automatic utterance rewriting strategies. Lastly, we asked the same human assessors who labeled the utterances to manually rewrite them. The result of this run (Manual Utterance) can be considered an upper bound for retrieval performance as manual rewriting should ideally outperform any automatic system.

5.2. Results and Discussion

The performances of the proposed utterance rewriting methods in comparison to the baselines detailed above are reported in Tables 7 and 8. Specifically, Table 7 refers to the effectiveness of the first-stage retrieval based on Indri, while Table 8 shows the end-to-end performance of our entire conversational IR pipeline, where the top 200 candidates retrieved by the first stage are scored by the second-stage neural re-ranker. In both the tables, we evaluate the effectiveness of our system with traditional IR metrics: Mean Average Precision (MAP), Mean Reciprocal Rank (MRR), normalized Discounted Cumulative Gain for cutoff at 3 (nDCG@3), and Precision for cutoffs at 1 and 3 (P@1 and P@3). The use of such small cutoffs is common for the conversational IR tasks since the user expects to receive one crisp answer on the top of the list rather than a long list of potentially relevant results. For assessing the performance of the first-stage retrieval components, we introduce in Table 7 also the metric Recall for cutoff at 200 (R@200) as an additional metric providing us information about the quality of the candidate set given as input to the second-stage neural re-ranker. For all methods and a given metric, we assess the statistical significance of the results by using a two-sample t-test with Bonferroni correction [58]. Results are considered significant at $p < 0.01$.

By looking at Table 7, reporting the results of the experiments conducted for assessing the performance of first-stage retrieval, we can observe that the proposed utterance rewriting methods outperform by a significant margin most of the baselines considered. All our methods perform well on the first stage. In particular, the performance of Last SE across all metrics except P@3 is slightly superior with respect to the other strategies for first-stage retrieval. It exhibits an MRR and P@1 of 0.5713 and 0.4855, respectively. Last SE shows also better performance compared to the baselines, i.e., First Topic and Topic Shift [1], although the difference is not statistically significant for all metrics.

Despite the good performance measured for our automatic methods deployed on the first stage, if we

Table 7: First-stage retrieval performance. The best results for each metric are reported in bold. We highlight with the superscript a statistical significant differences at $p < 0.01$ between all methods and our best-performing method Last SE according to the Bonferroni-corrected two-sample t-test. Analogously, statistically significant differences between ground truth methods (GT Last SE and GT First and/or last SE) and Manual Utterance are indicated with the the superscript b according to the Bonferroni-corrected two-sample t-test.

	First-stage Retrieval					
	MAP	MRR	nDCG@3	P@1	P@3	R@200
CAsT baseline	0.1299^a	0.3178^a	0.1477^a	0.2254^a	0.2428^a	0.2939^a
Query	0.1741^a	0.4216^a	0.2181^a	0.3353^a	0.3218^a	0.3928^a
First Query	0.2091	0.4931	0.2663	0.3988	0.4008	0.4588
Context Query	0.1903^a	0.4709	0.2315	0.3565	0.4008	0.4263^a
Plain Utterance	0.1416^a	0.3483^a	0.1735^a	0.2849^a	0.2694^a	0.2920^a
CoRef$_1$	0.1691^a	0.4203^a	0.2148^a	0.3410^a	0.3218^a	0.3772^a
CoRef$_2$	0.1845^a	0.4308^a	0.2209^a	0.3526^a	0.3430^a	0.3818^a
Context	0.2053	0.4701^a	0.2401^a	0.3757	0.3680^a	0.4496
First Topic	0.2286	0.5543	0.3041	0.4682	**0.4644**	0.4604
Topic Shift	0.2274	0.5513	0.3017	0.4509	0.4605	0.4680
Standard	0.2296	0.5396	0.2917	0.4566	0.4412	0.4797
Enriched	0.2329	0.5482	0.2964	0.4624	0.4412	0.5033
Last SE (a)	**0.2444**	**0.5713**	**0.3072**	**0.4855**	0.4566	**0.5071**
First and last SE	0.2358	0.5563	0.2975	0.4682	0.4605	0.4835
First or last SE	0.2339	0.5526	0.3018	0.4682	0.447	0.5011
GT Last SE	0.2678^b	0.6139^b	0.3357^b	0.5318^b	0.4933^b	0.5345^b
GT First and last SE	0.2489^b	0.5770^b	0.3149^b	0.4913^b	0.4817	0.5049^b
GT First or last SE	0.2758	0.6258^b	0.3513	0.5434	0.5145	0.5498
Manual Utterance (b)	0.2978^a	0.6859^a	0.3828^a	0.6127^a	0.5491^a	0.5771^a

look at the performance of the GT methods based on the classification ground truth, we see that there is still margin for improvement. We note that the two rewriting strategies exploiting the labels in the golden standard, all ground truth based methods, GT Last SE, GT First and Last SE and GT First or Last SE, perform better than our best automatic methods based on the classifier outcome. This means that the classification errors introduced by the classifier cascade discussed in Section 4.2 impact the effectiveness of our utterance rewriting methods, although the impact on the performance is not statistically significant at $p < 0.01$ according to the Bonferroni-corrected two-sample t-test. Increasing the accuracy of the context-identification classifiers is likely to result in an improved effectiveness of the conversational IR solution and is going to fill the performance gap measured for automatic classification.

Furthermore, we highlight that the performance of the Manual Utterance supervised method is quite close to those of GT Last SE and GT First or Last SE, especially in the case of GT First or Last SE which presents statistically significant differences from Manual Utterance only for MRR (as highlighted by the superscript b), while for all other metrics there are no statistical differences between the two methods.

For example, the performance gap between Manual Utterance and GT First or Last SE is limited to about three points in percentage of MAP, nDCG@3, P@3 and Recall@200. This relatively small difference with respect to an optimal manual rewriting method allows us to claim that the proposed modelling of the utterance understanding problem based on SE, FT and PT classes is sound and the associated rewriting strategies are effective and well designed.

Complementary considerations and insights can be derived when looking at the end-to-end performance reported in Table 8. Here we can see much higher effectiveness figures than with first-stage retrieval, once more witnessing the importance of a re-ranking step based on complex machine-learned ranking models [41, 59, 60]. The BERT-based neural re-ranker seems to exploit better than the first-stage ranker the proposed rewriting strategies. If we look at the values reported in bold in the table we can observe that First and Last SE is the best performing method for what regards MAP, MRR and P@1, P@3 metrics. For example First and Last SE achieves the highest reported values for MRR and P@1 of 0.7330 and 0.6512, respectively, while in terms of NDCG@3 the best performance is 0.4484 corresponding to Last SE.

However, it is worth noting that, due to the relatively small number of utterances in the test set, the performance differences between First and Last SE (a) and the other proposed automatic rewriting methods are not statistically significant. The same holds for First and Last SE (a) and the previously proposed methods First Topic and Topic Shift, although we see improvements for all evaluation metrics. The superiority of First and Last SE method stems from the ability of the BERT ranker to identify the proper context information in the query and the candidate documents. This characteristic likely reduces the impact of some of the classification errors since instead of selecting which enrichment to add, either first or last SE, a method that considers both, first and last SE, proves to be overall advantageous. On the other hand, when looking at the effectiveness of the methods based on the ground truth, we note that GT First or Last SE outperforms the other GT methods, resulting in MRR, nDCG@3, P@1 and P@3 figures that are not statistically different from those achieved with Manual Utterance (b). Thus, as observed for the first-stage results, also on the second stage we have an important margin for improving the effectiveness of our conversational IR pipeline by enhancing the classification accuracy of the context identification component. We see that the performance gap between the methods using automatic classification and the ground truth remains almost constant for the first and second stages, thus suggesting that the performance benefit (penalty) obtained on rightly (wrongly) classified utterances is almost independent of the ranking model adopted.

To summarize, the findings presented validate our proposed conversational IR pipeline described in Figure 1. That is, using our proposed utterance rewriting component, including BERT-based context classification and utterance rewriting leveraging context information extracted from the first and/or last self-explanatory utterances in the conversation, we improve the effectiveness of both the first-stage retrieval and, more critically, the whole end-to-end system performance as compared with existing approaches.

We now present an analysis of the performance based on conversations for the best performing rewriting method highlighted by the end-to-end evaluation of the previous section. As we can see from Table 8, the method First and last SE has the best performance for most of the metrics. We pick this method for analyzing in detail its performance per conversation. For this analysis, we use nDCG@3 as it is often employed for measuring the performance of conversational IR systems, in particular, it has been used in CAsT for the final ranking of the participants' turns. The idea is exploring how some conversations can be more difficult than others. As we can see from Figure 3, some of them are tricky leading to lower performance in term of end-to-end retrieval. In particular, we observe that conversations 54 and 69 have nDCG@3 values equal to 0.10 and 0.13, respectively. These low values stem from misclassifications at critical points of the conversation. For example, in conversation 69 about *sleeping problems*, the misclassification occurs at a lower turn, and the wrong topic is propagated to the follow up utterances. In particular, at turn 5, the utterance "Is melatonin bad for you?" is predicted as PT instead of SE. So, the approach of First and last SE does not use the newly introduced topic "melatonin" as the utterance is not classified as SE and propagates the topic "jet leg." This mistake is repeated in the rest of the conversation where "melatonin" is missing. Conversation 54 is about *Washington D.C.* Although in this conversation, most of the utterance labels are predicted correctly, there is a misclassification at turn 6 "What is the best time to visit the reflecting pools?" Indeed, the utterance is predicted as PT instead of FT, so the approach propagates fist topic ("Washington D.C.") and previous topic ("Spy museum") increasing the noise in the results.

Table 8: End-to-end retrieval performance. The best results for each metric are reported in bold. We highlight with the superscript a statistical significant differences at $p < 0.01$ between all methods and our best performing method First and last SE according to the Bonferroni-corrected two-sample t-test. Analogously, statistically significant differences between ground truth methods (GT Last SE and GT First and/or last SE) and Manual Utterance are indicated with the superscript b according to the Bonferroni-corrected two-sample t-test.

	End-to-End Retrieval				
	MAP	MRR	nDCG@3	P@1	P@3
CAsT baseline	0.1316^a	0.4002^a	0.2089^a	0.3064^a	0.2987^a
Query	0.1666^a	0.4723^a	0.2660^a	0.3547^a	0.3624^a
First Query	0.2095^a	0.5945^a	0.3195^a	0.4884^a	0.4477^a
Context Query	0.1905^a	0.5624^a	0.2997^a	0.4535^a	0.4147^a
Plain Utterance	0.1408^a	0.4605^a	0.2791^a	0.3953^a	0.3682^a
CoRef_1	0.1797^a	0.5563^a	0.3474	0.4795^a	0.4405^a
CoRef_2	0.1906^a	0.5718^a	0.3483	0.5000	0.4554^a
Context	0.2098^a	0.6315	0.3692	0.5291^a	0.4961^a
First Topic	0.2335	0.6763	0.3897	0.5954	0.5453
Topic Shift	0.2330	0.6801	0.3967	0.5872	0.5523
Standard	0.2403	0.7058	0.4411	0.6337	0.5756
Enriched	0.2455	0.7194	0.4425	0.6395	0.5814
Last SE	0.2500	0.7284	**0.4484**	0.6453	0.5853
First and last SE (a)	**0.2507**	**0.7330**	0.4343	**0.6512**	**0.5988**
First or last SE	0.2434	0.7200	0.4472	0.6453	0.5814
GT Last SE	0.2680^b	0.7557	0.4584^b	0.6802	0.6143^b
GT First and last SE	0.2588^b	0.7291	0.4303^b	0.6337	0.6066^b
GT First or last SE	0.2786^b	0.7717	0.4742	0.6977	0.6318
Manual Utterance (b)	0.3055^a	0.8006	0.5167^a	0.7168	0.6763

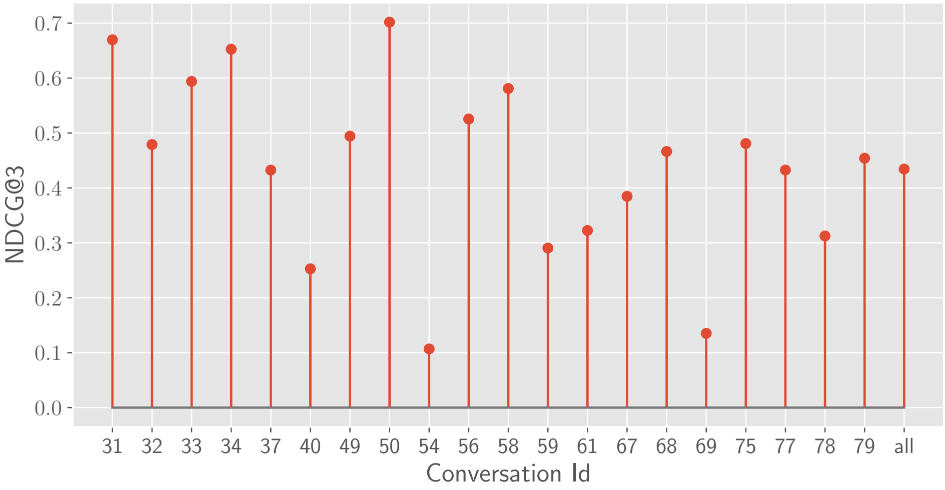

Figure 3: Per-conversation analysis: nDCG@3 averaged over all utterances for each single conversation using First and last `SE`.

6. Conclusion

We investigated the problem of utterance understanding and rewriting for conversational search. We designed automatic methods that, using information derived from the whole conversation, enrich utterances likely to be answered poorly by an IR system optimized to process self-contained queries lacking context information. We designed a fully-fledged conversational system structured as a three-step cascade architecture. Our contribution focuses primarily on the utterance understanding component of such an architecture. This component has the task of performing linguistic analysis on the current utterance and choosing how to rewrite it to improve the end-to-end effectiveness of the conversational system. The proposed rewriting strategy is driven by a classifier that aims at understanding if the utterance is self-explanatory or it misses explicit references to the conversation context. In this latter case, the classifier also helps to identify which context is more suitable for enriching it, given the previous utterances in the conversation.

We conducted a comprehensive experimental evaluation of the performance of our solution by adopting the CAsT 2019 framework [3]. First, we assessed the performance of different supervised binary and multi-label classification models trained to automatically associate one of the three class labels, i.e., `SE`, `FT` or `PT`, to the utterances processed by the conversational search architecture. The analysis considered either the accuracy of the classifiers in isolation and when integrated in the conversational system, where an error of the first binary classifier (`SE` vs. non-`SE`) affects also the result of the second classification (`FT` vs. `PT`). In our context identification task, the experiments conducted showed the superior performance of a cascade of two binary classifiers based on the BERT bidirectional transformer over those based on GBDT ensembles trained on hand-crafted features. Specifically, the best performance was achieved by a BERT-BERT$_{\mathsf{ctx}}$ binary classification cascade with a weighted F1 score of 0.76.

We then compared the retrieval effectiveness of our method to state-of-the-art competitors and baselines. By using IR metrics commonly used for conversational search, we measured both the performance at the first retrieval stage and in an end-to-end setting, where first-stage candidates are re-ranked by a state-of-the-art neural model based on BERT. Our experimental results showed that adding context to the utterances based on the predictions of our classifiers is crucial for the effective retrieval of relevant documents when the reference to the topic of the utterance is obfuscated or totally missing. In detail, experiments showed that pseudo-relevance feedback is beneficial as it further expands the query with keywords extracted from the top K results allowing an improvement of recall. On the other hand, neural re-ranking provides an important boost in precision at very short cutoffs. Results showed that our best approach achieves an improvement of 0.4855 (+215%) for P@1 and 0.4566 (+188%) for P@3 at the first-stage retrieval and of 0.6512 (+213%) for P@1 and 0.5988 (+200%) for P@3 at the neural-based re-ranking stage w.r.t. the CAsT provided baseline. The improvement for nDCG@3 is instead of 0.3072 (+208%) at first-stage retrieval and 0.4484 (+215%) at neural re-ranking stage, respectively.

By comparing the performance of our automatic methods with those based on the golden truth, we observed a margin for improving the effectiveness of our conversational IR pipeline by enhancing the classification accuracy of its context identification component. On the other hand, the supervised methods using the classification labels performed very similarly to the upper bound established by manually rewritten utterances. This observation, together with the relatively poor end-to-end performance measured for the simpler baselines reported, prove that enriching utterances with the proper context is mandatory for retrieval effectiveness since also complex contextualized models alone, e.g., BERT, are not sufficient for utterance understanding, and deserve further investigation.

Acknowledgements. Work partially funded by the EU Horizon 2020 research and innovation programme (BigDataGrapes project, GA no. 780751), by the Italian Ministry of Education and Research in the framework of the CrossLab project (Departments of Excellence) and the OK-INSAID project (GA no. ARS01_00917), and by the University of Pisa in the framework of the PRA 2020 program (AUTENS project, Sustainable Energy Autarky).

References

[1] Ida Mele, Cristina Ioana Muntean, Franco Maria Nardini, Raffaele Perego, Nicola Tonellotto, and Ophir Frieder. Topic propagation in conversational search. In *Proc. ACM SIGIR*, pages 2057–2060, New York, NY, USA, 2020. ACM.

[2] Rodrigo Nogueira and Kyunghyun Cho. Passage re-ranking with bert. ArXiv Preprint 1901.04085, 2019.

[3] Jeffrey Dalton, Chenyan Xiong, Vaibhav Kumar, and Jamie Callan. Cast-19: A dataset for conversational information seeking. In *Proc. SIGIR*, page 1985–1988, New York, NY, USA, 2020. ACM.

[4] Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. AllenNLP: A deep semantic natural language processing platform. In *Proc. NLP-OSS*, pages 1–6, Copenhagen, Denmark, 2018. ACL.

[5] Nicholas J. Belkin. Anomalous states of knowledge as a basis for information retrieval. *Canadian journal of information science*, 5(1):133–143, 1980.

[6] W. Bruce Croft and Roger H Thompson. I3r: A new approach to the design of document retrieval systems. *Journal of the american society for information science*, 38(6):389–404, 1987.

[7] Alexandra Vtyurina, Denis Savenkov, Eugene Agichtein, and Charles LA Clarke. Exploring conversational search with humans, assistants, and wizards. In *Proc. CHI*, pages 2187–2193, New York, NY, USA, 2017. ACM.

[8] Filip Radlinski and Nick Craswell. A theoretical framework for conversational search. In *Proc. CHIIR*, page 117–126, New York, NY, USA, 2017. ACM.

[9] Neelakshi Sarma, Sanasam Ranbir Singh, and Diganta Goswami. Influence of social conversational features on language identification in highly multilingual online conversations. *Information Processing & Management*, 56(1):151 – 166, 2019.

[10] Zhe Liu and Bernard J. Jansen. Questioner or question: Predicting the response rate in social question and answering on sina weibo. *Information Processing & Management*, 54(2):159 – 174, 2018.

[11] Johanne R. Trippas, Damiano Spina, Paul Thomas, Mark Sanderson, Hideo Joho, and Lawrence Cavedon. Towards a model for spoken conversational search. *Information Processing & Management*, 57(2):102162, 2020.

[12] Leila Tavakoli. Generating clarifying questions in conversational search systems. In *Proc. CIKM*, page 3253–3256, New York, NY, USA, 2020. ACM.

[13] Chris Buckley, Gerard Salton, James Allan, and Amit Singhal. Automatic query expansion using smart: Trec 3. In *TREC 3*, pages 69–80, USA, 1995. NIST.

[14] Liu Yang, Hamed Zamani, Yongfeng Zhang, Jiafeng Guo, and W Bruce Croft. Neural matching models for question retrieval and next question prediction in conversation. ArXiv Preprint 1707.05409, 2017.

[15] Gary Ren, Xiaochuan Ni, Manish Malik, and Qifa Ke. Conversational query understanding using sequence to sequence modeling. In *Proc. WWW*, pages 1715 – 1724, New York, NY, USA, 2018. ACM.

[16] Mohammad Aliannejadi, Manajit Chakraborty, Esteban Andrés Ríssola, and Fabio Crestani. Harnessing evolution of multi-turn conversations for effective answer retrieval. In *Proc. CHIIR*, page 33–42, New York, NY, USA, 2020. ACM.

[17] Shi Yu, Jiahua Liu, Jingqin Yang, Chenyan Xiong, Paul Bennett, Jianfeng Gao, and Zhiyuan Liu. Few-shot generative conversational query rewriting. In *Proc. SIGIR*, page 1933–1936, New York, NY, USA, 2020. ACM.

[18] Svitlana Vakulenko, Shayne Longpre, Zhucheng Tu, and Raviteja Anantha. Question rewriting for conversational question answering. In *Proc. WSDM*, page 355–363. ACM, 2021.

[19] Svitlana Vakulenko, Nikos Voskarides, Zhucheng Tu, and Shayne Longpre. A comparison of question rewriting methods for conversational passage retrieval. In *Proc. ECIR*, pages 418–424. Springer, 2021.

[20] Liu Yang, Minghui Qiu, Chen Qu, Jiafeng Guo, Yongfeng Zhang, W. Bruce Croft, Jun Huang, and Haiqing Chen. Response ranking with deep matching networks and external knowledge in information-seeking conversation systems. In *Proc. SIGIR*, pages 245–254, New York, NY, USA, 2018. ACM.

[21] Nikos Voskarides, Dan Li, Pengjie Ren, Evangelos Kanoulas, and Maarten de Rijke. Query resolution for conversational search with limited supervision. In *Proc. SIGIR*, page 921–930, New York, NY, USA, 2020. ACM.

[22] Ricardo Baeza-Yates and Alessandro Tiberi. Extracting semantic relations from query logs. In *Proc. SIGKDD*, pages 76–85, New York, NY, USA, 2007. ACM.

[23] Fabrizio Silvestri. Mining query logs: Turning search usage data into knowledge. *Foundations and Trends in Information Retrieval*, 4(1-2):1–174, 2010.

[24] Zhiyong Zhang and Olfa Nasraoui. Mining search engine query logs for query recommendation. In *Proc. WWW*, page 1039–1040, New York, NY, USA, 2006. ACM.

[25] Ricardo Baeza-Yates, Carlos Hurtado, and Marcelo Mendoza. Query recommendation using query logs in search engines. In *Proc. EDBT*, pages 588–596, Berlin, Heidelberg, 2004. Springer.

[26] Nick Craswell, Onno Zoeter, Michael Taylor, and Bill Ramsey. An experimental comparison of click position-bias models. In *Proc. WSDM*, page 87–94, New York, NY, USA, 2008. ACM.

[27] Paolo Boldi, Francesco Bonchi, Carlos Castillo, Debora Donato, Aristides Gionis, and Sebastiano Vigna. The query-flow graph: Model and applications. In *Proc. CIKM*, page 609–618, New York, NY, USA, 2008. ACM.

[28] Claudio Lucchese, Salvatore Orlando, Raffaele Perego, Fabrizio Silvestri, and Gabriele Tolomei. Identifying task-based sessions in search engine query logs. In *Proc. WSDM*, pages 277–286, New York, NY, USA, 2011. ACM.

[29] Filip Radlinski and Thorsten Joachims. Query chains: Learning to rank from implicit feedback. In *Proc. SIGKDD*, page 239–248, New York, NY, USA, 2005. ACM.

[30] Tessa Lau and Eric Horvitz. Patterns of search: analyzing and modeling web query refinement. In *UM99 user modeling*, pages 119–128. Springer, Berlin, Heidelberg, 1999.

[31] Soo Young Rieh and Hong Xie. Analysis of multiple query reformulations on the web: The interactive information retrieval context. *Information Processing & Management*, 42(3):751–768, 2006.

[32] Steven M. Beitzel, Eric C. Jensen, Abdur Chowdhury, and Ophir Frieder. Varying approaches to topical web query classification. In *Proc. SIGIR*, page 783–784, New York, NY, USA, 2007. ACM.

[33] Steven M. Beitzel, Eric C. Jensen, Abdur Chowdhury, David Grossman, and Ophir Frieder. Hourly analysis of a very large topically categorized web query log. In *Proc. SIGIR*, page 321–328, New York, NY, USA, 2004. ACM.

[34] Steven M. Beitzel, Eric C. Jensen, Ophir Frieder, David D. Lewis, Abdur Chowdhury, and A. Kolcz. Improving automatic query classification via semi-supervised learning. In *Proc. ICDM*, 2005.

[35] Steven M. Beitzel, Eric C. Jensen, David D. Lewis, Abdur Chowdhury, and Ophir Frieder. Automatic classification of web queries using very large unlabeled query logs. *ACM Trans. Inf. Syst.*, 25(2), April 2007.

[36] Bernard J. Jansen, Amanda Spink, and Bhuva Narayan. Query modifications patterns during web searching. In *Proc. ITNG*, pages 439–444, Washington DC, DC, USA, 2007. IEEE CS.

[37] Bernard J. Jansen, Mimi Zhang, and Amanda Spink. Patterns and transitions of query reformulation during web searching. *International Journal of Web Information Systems*, 3(4):328–340, 2007.

[38] Daqing He, Ayşe Göker, and David J Harper. Combining evidence for automatic web session identification. *Information Processing & Management*, 38(5):727–742, 2002.

[39] Paolo Boldi, Francesco Bonchi, Carlos Castillo, and Sebastiano Vigna. From "dango" to "japanese cakes": Query reformulation models and patterns. In *Proc. WI-IAT*, page 183–190, Washington DC, DC, USA, 2009. IEEE CS.

[40] Paolo Boldi, Francesco Bonchi, Carlos Castillo, and Sebastiano Vigna. Query reformulation mining: models, patterns, and applications. *Information retrieval*, 14(3):257–289, 2011.

[41] Gabriele Capannini, Claudio Lucchese, Franco Maria Nardini, Salvatore Orlando, Raffaele Perego, and Nicola Tonellotto. Quality versus efficiency in document scoring with learning-to-rank models. *Information Processing & Management*, 52(6):1161–1177, 2016.

[42] Jiafeng Guo, Yixing Fan, Liang Pang, Liu Yang, Qingyao Ai, Hamed Zamani, Chen Wu, W. Bruce Croft, and Xueqi Cheng. A deep look into neural ranking models for information retrieval. *Information Processing & Management*, 57(6):102067, 2020.

[43] Nicola Tonellotto, Craig Macdonald, and Iadh Ounis. Efficient query processing for scalable web search. *Foundations and Trends in Information Retrieval*, 12(4–5):319–492, 2018.

[44] Trevor Strohman, Donald Metzler, Howard Turtle, and W Bruce Croft. Indri: A language model-based search engine for complex queries. *Proc. ICIA*, 2(6):2–6, 2005.

[45] Iadh Ounis, Gianni Amati, Vassilis Plachouras, Ben He, Craig Macdonald, and Douglas Johnson. Terrier information retrieval platform. In *Proc. ECIR*, pages 517–519, Berlin, Heidelberg, 2005. Springer.

[46] Craig Macdonald, Nicola Tonellotto, and Iadh Ounis. Efficient and effective selective query rewriting with efficiency predictions. In *Proc. SIGIR*, pages 495–504, New York, NY, USA, 2017. ACM.

[47] Nicola Tonellotto, Craig Macdonald, and Iadh Ounis. Efficient and effective retrieval using selective pruning. In *Proc. WSDM*, pages 63–72, New York, NY, USA, 2013. ACM.

[48] Philipp Christmann, Rishiraj Saha Roy, Abdalghani Abujabal, Jyotsna Singh, and Gerhard Weikum. Look before you hop: Conversational question answering over knowledge graphs using judicious context expansion. In *Proc. CIKM*, page 729–738, New York, NY, USA, 2019. ACM.

[49] Joseph L. Fleiss. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5):378–382, 1971.

[50] J. Richard Landis and Gary G. Koch. The measurement of observer agreement for categorical data. *Biometrics*, 33(1):159–174, 1977.

[51] Zhaohui Zheng, Hongyuan Zha, Tong Zhang, Olivier Chapelle, Keke Chen, and Gordon Sun. A general boosting method and its application to learning ranking functions for web search. In *Proc. NIPS*, pages 1697–1704. Curran Associates, Inc., Red Hook, NY, USA, 2008.

[52] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proc. NAACL-HLT*, pages 4171–4186, Copenhagen, Denmark, 2019. ACL.

[53] Charles L. A. Clark, Stefan Büttcher, and Gordon V. Cormack. *Information Retrieval: Implementing and Evaluating Search Engines.* MIT Press, Cambridge, MA, USA, 2010.

[54] Sean MacAvaney, Andrew Yates, Arman Cohan, and Nazli Goharian. Cedr: Contextualized embeddings for document ranking. In *Proc. ACM SIGIR*, page 1101–1104, New York, NY, USA, 2019. ACM.

[55] Wei Yang, Haotian Zhang, and Jimmy Lin. Simple applications of bert for ad hoc document retrieval, 2019.

[56] Sebastian Hofstätter, Navid Rekabsaz, Carsten Eickhoff, and Allan Hanbury. On the effect of low-frequency terms on neural-ir models. In *Proc. SIGIR*, page 1137–1140, New York, NY, USA, 2019. ACM.

[57] Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. End-to-end neural coreference resolution. In *Proc. EMNLP*, pages 188–197, Copenhagen, Denmark, 2017. ACL.

[58] Tetsuya Sakai. *Laboratory Experiments in Information Retrieval - Sample Sizes, Effect Sizes, and Statistical Power*, volume 40 of *The Information Retrieval Series.* Springer, 2018.

[59] Domenico Dato, Claudio Lucchese, Franco Maria Nardini, Salvatore Orlando, Raffaele Perego, Nicola Tonellotto, and Rossano Venturini. Fast ranking with additive ensembles of oblivious and non-oblivious regression trees. *ACM Transactions on Information Systems*, 35(2):15:1–15:31, 2016.

[60] Francesco Lettich, Claudio Lucchese, Franco Maria Nardini, Salvatore Orlando, Raffaele Perego, Nicola Tonellotto, and Rossano Venturini. Parallel traversal of large ensembles of decision trees. *IEEE Transactions on Parallel and Distributed Systems*, 30(9):2075–2089, 2019.