IDIAP RESEARCH REPORT

QUESTION ANSWERING IN CONVERSATIONS: QUERY REFINEMENT USING CONTEXTUAL AND SEMANTIC INFORMATION

Maryam Habibi Parvaz Mahdabi
Andrei Popescu-Belis

Idiap-RR-16-2016

JUNE 2016

Centre du Parc, Rue Marconi 19, P.O. Box 592, CH - 1920 Martigny
T +41 27 721 77 11 F +41 27 721 77 12 info@idiap.ch www.idiap.ch

Question Answering in Conversations: Query Refinement Using Contextual and Semantic Information

Maryam Habibi[1]

Humboldt-Universität zu Berlin, Institut für Informatik, Unter den Linden 6, 10099 Berlin, Germany

Parvaz Mahdabi, Andrei Popescu-Belis

Idiap Research Institute, Rue Marconi 19, 1920 Martigny, Switzerland

Abstract

This paper introduces a query refinement method applied to questions asked by users to a system during a meeting or a conversation that they have with other users. To answer the questions, the proposed method leverages the local context of the conversation along with semantic resources, either WordNet or word embeddings from word2vec. The method first represents the local context by extracting keywords from the transcript of the conversation, which is obtained from a real-time Automatic Speech Recognition (ASR) system and may contain noise. It then expands the queries with keywords that best represent the topic of the query, i.e. expansion keywords accompanied by weights indicating their topical similarity to the query. Finally, semantically related terms are added, using two options: either synonymous terms drawn from WordNet or similar words based on distributed representations in a low-dimensional word embedding space learned using word2vec. To evaluate the system, we introduce a dataset (named AREX for AMI Requests for Explanations) and an evaluation metric based on relevance judgments collected by crowdsourcing. We compare our query expansion approach with other methods, over queries from the AREX dataset, showing the superiority of our method when either manual or automatic

[1]Work performed while at the Idiap Research Institute.

transcripts of the AMI Meeting Corpus are used.

Keywords: Query Refinement, Query Expansion, Context Modeling, Speech-based Information Retrieval, Evaluation of Information Retrieval.

1. Introduction

In this paper, we propose a new query refinement method applied to clarification questions asked by people during a meeting. For instance, if the meeting participants discuss the design of a remote control, a participant may need additional information about the acronym"LCD". Our goal is to design a system answering the participant's query for more explanations about "LCD", in this case by displaying the most helpful Wikipedia pages. However, out of its context, such terms often have several potential interpretations. Here, the acronym "LCD" can refer to the 'lowest common denominator' or the 'Lesotho Congress for Democracy', in addition to 'liquid-crystal display', which is the correct interpretation in this context. A service such as `www.acronymfinder.com` would typically list all possible interpretations (in this case, 44 for 'LCD') but would not offer any help to disambiguate them, apart ranking them based on popularity.

Assuming that spoken questions can be properly detected by a system, our aim in this paper is to address the problem of their potential ambiguity. We propose to use the local context of the conversation, as well as additional semantic knowledge, to refine the initial query by expanding it implicitly with additional words, obtained from a real-time Automatic Speech Recognition (ASR) system. Previous query refinement techniques enrich queries either interactively, by asking users to validate certain words, or automatically, by adding relevant words from an external data source. However, interacting with users for query refinement may distract them from their current conversation, while using an external data source outside the users' local context may cause misinterpretations without a proper disambiguation of the query. To address these challenges, several previous studies have attempted to use the local context of users' activities,

without requiring user interaction [1, 2]. However, as we will show, they are not entirely suitable for a conversational environment, because of the nature of the vocabulary and the errors introduced by the ASR system.

The techniques we will use to model the local context and to provide semantically-related expansion terms are designed specifically for such conversational environments, for intelligent personal assistants that answer clarification questions within a human-human conversation. The contributions of this paper are therefore the following ones:

1. The local context of an explicit query is represented by a keyword set that is automatically obtained from the conversation fragment preceding the query using a robust keyword extraction method that we proposed previously [3, 4]. We assign a weight value to each keyword, based on its topical similarity to the explicit query, to reduce the effect of the ASR noise, and to recognize appropriate interpretations of the query.
2. Furthermore, we perform semantic query expansion (SQE), by searching for variants of query words that seem insufficiently represented in the results, using two approaches: WordNet synonyms [5], or words with similar representations in a low-dimensional embedding space built using word2vec [6].
3. To evaluate the improvement brought by our method, we constructed the AREX dataset (AMI Requests for Explanations with Relevance Judgments for their Answers), a dataset which is publicly available at `www.idiap.ch/dataset/arex`. This dataset contains a set of explicit queries inserted into conversations of the AMI Meeting Corpus [7], along with a set of human relevance judgments over sample retrieval results from Wikipedia for each query. The dataset is accompanied by an automatic evaluation metric based on Mean Average Precision (MAP).
4. The experiments show the superiority of our technique over previous ones and its robustness against unrelated keywords or ASR noise. Additionally, while query expansion with contextual knowledge already outper-

forms previous techniques, semantic query expansion further increases the relevance of the resulting documents. Among the two semantic query expansion approaches, the results show that word embeddings outperform WordNet.

The paper is organized as follows. In Section 2, we review existing methods for query refinement or expansion. In Section 3, we describe the proposed query refinement method based on the conversational context. Section 4 explains how the AREX dataset was constructed, using crowdsourcing to obtain relevance judgments, and specifies the evaluation metric associated to it. Section 5 presents and discusses the experimental results obtained with human-made transcripts of the AMI Meeting Corpus and with the output of a real-time ASR system.

2. Related Work

Several methods for the refinement of explicit queries asked by users have been proposed in the field of information retrieval, and are often referred to as query expansion techniques [8]. Query expansion methods hypothesize one or more words or terms to add to a query by recognizing its possible interpretations. These methods use knowledge coming either directly from the document corpus over which retrieval is performed [9, 10, 11, 12, 13] or from Web data or personal profiles in the case of Web search [14, 15, 16, 17]. Moreover, query expansion techniques may select suggestions for query refinement either interactively or automatically [8]. An example query expansion technique, called relevance feedback, gathers judgments from users on sample results obtained from an initial query, from which it extracts expansion terms, rather than asking users to rate directly such terms [18, 19, 20].

Such methods are not ideal for refinement of explicit queries asked during a conversation, because they require users to interrupt their conversation. On the contrary, our overall goal is to estimate users' information needs from their explicit queries with as little intrusion as possible. Moreover, using the local

context for query refinement instead of external, non-contextual resources has the potential to improve retrieval results [2].

To the best of our knowledge, only two previous systems have utilized the local context for the augmentation of explicit queries. The JIT-MobIR system for mobile devices [1] used contextual features from the physical and the human environment, but the content of the activities itself was not used as a feature. The WATSON system [2] refined explicit queries by concatenating them with keywords extracted from the documents being edited or viewed by the user. However, in order to apply the same method to a retrieval system for which the local context is a conversation, the keyword lists must avoid considering irrelevant topics from ASR errors. Moreover, unlike written documents which follow generally a planned and focused structure, in a conversation users often turn from one topic to another (an issue we addressed in our previous work [4]), and adding such a variety of keywords to a query might deteriorate the retrieval results [21, 8].

A less studied dimension of query expansion is selective query expansion, which resorts to a diagnosis to identify which parts of a query really need to be expanded. This diagnosis is followed by an intervention on those parts via automatic query refinement and/or interaction with the user [22]. Recently, researchers found that several factors cause vocabulary mismatch [23], such as a query term not being central to the information need, or requiring replacement by synonyms, or being too abstract or too rare. A supervised learning approach with access to past queries was shown to enable the prediction of query terms to be expanded [23].

In this paper, we disambiguate and expand queries that are formulated during a conversation, and propose a dataset to evaluate this task. We first augment the queries using the keywords extracted from the ASR transcript of the conversation by a method which we proposed earlier [3]. In a different previous study [4], we used these keywords to formulate implicit queries for retrieving and recommending relevant documents to participants. In the present study, we improve the retrieval results of explicit queries using expansion terms that

are extracted using external semantic resources like WordNet [5] or word embeddings from word2vec [6]. As we will show, the keywords extracted from the conversation help to obtain more relevant expansion words from external semantic resources.

3. Content-based Query Refinement

The application framework considered in this paper is inspired from the Automatic Content Linking Device [24, 25, 26], which monitors a conversation between its users, for instance within a business meeting, and makes spontaneous recommendations of relevant documents. Our system extends the framework by allowing its users to formulate explicit spoken queries to retrieve documents, in particular to obtain explanations about notions (words, terms, or acronyms) that they might ignore. The documents can be retrieved from the Web or from a specific repository: throughout this paper, our repository is the English Wikipedia obtained from the Freebase Wikipedia Extraction (WEX) dataset from Metaweb Technologies.[2]

The users can simply address the system by using a pre-defined unambiguous proper name (such as "John"), which is robustly recognized by the real-time automatic speech recognition system (ASR) component [27]. More sophisticated strategies for addressing a system in a multi-party dialogue context have been studied [28, 29], but they are beyond the scope of this paper, which is concerned with processing the query itself. Once the results are generated by the system, they are displayed on each user's device (typically the laptop they use during the meeting) or on a shared projection screen.

To answer an explicit query Q, we first refine it by expanding it with related keywords which are likely to increase the relevance of results by disambiguating the short explicit query. We refine the query using a two-stage approach: firstly (Section 3.1) we extract topically-related keywords from the local context of the

[2]Version dated 2009-06-16, see `http://download.freebase.com/wex`.

conversation, and secondly (Section 3.2) we consider the words from the query which are under-represented in the intermediary retrieval results (retrieved with the query at the first stage) and add either their synonyms from WordNet, or words with a similar representation in a low-dimensional embedding space built using word2vec. After the second stage, we re-run the query to obtain the final results.

3.1. Query Expansion Using Words from the Local Conversational Context

The process of query refinement starts by modeling the local context using the transcript of a short conversation fragment immediately preceding the query. We use the same fixed length for all the fragments, though more sophisticated strategies are under consideration too. From the local context, we extract a keyword set C using a diverse keyword extraction technique that we previously proposed [3, 4], which maximizes the coverage of the fragment's topics with keywords; this technique considers the topical similarity of the keywords with the conversation and preserves the diversity of the mentioned topics.[3]

We then weigh the extracted keywords by using a filter that assigns a weight m_i to each keyword $kw_i \in C \setminus Q$, with $0 \leq m_i < 1$, based on the normalized topical similarity of the keyword to the explicit query. The weight is computed using cosine similarity between the keyword and the query vectors in the topic space, as follows:

$$m_i = \frac{\sum_{z \in Z} p(z|Q)p(z|kw_i)}{\sqrt{\sum_{z \in Z} p(z|kw_i)^2}\sqrt{\sum_{z \in Z} p(z|Q)^2}} \quad (1)$$

In this equation, Z is the set of abstract topics which correspond to latent variables inferred using a topic modeling technique over a large collection of documents, and $p(z|kw_i)$ is the distribution of topic z in relation to the keyword kw_i. Similarly, $p(z|Q) = (\sum_{q \in Q} p(z|q))/|Q|$ is the averaged distribution of topic z in relation to the query Q made of query words q.

[3] We omitted the details of the construction of keyword set C here as it is out of the scope of this paper, and is described in previous papers [3, 4].

The topic distributions are created using the LDA (Latent Dirichlet Analysis) topic modeling technique [30], implemented in the Mallet toolkit [31]. The topic models are learned over a large subset of the English Wikipedia with around 125,000 randomly sampled documents, following insights from previous studies [32]. Similarly, we fixed the number of topics at 100 [32, 33].

Each query Q is thus refined by adding additional keywords extracted from the fragment, with a certain weight. Note that we do not weigh all the words of the fragment, but only those selected as keywords, in order to avoid expanding the query with words that are relevant to one of the query aspects but not to the main topics of the fragment. We obtain a parametrized refined query $RQ(\lambda)$ which is a set of weighted keywords, i.e. pairs of (word, weight):

$$RQ(\lambda) = \{(q_1, 1), \ldots, (q_{|Q|}, 1), (kw_1, m_1^\lambda), \ldots, (kw_{|C|}, m_{|C|}^\lambda)\} \tag{2}$$

In other words, the refined query RQ contains $|Q|$ words from the explicit query Q with weight 1, and $|C|$ expansion keywords from the keyword set C with a weight proportional to their topic similarity to the query (calculated according to Eq. 1). Although in this paper, with the AREX dataset, we focus on single-term queries (i.e. clarification questions on acronyms, hence $|Q| = 1$), the method can be applied more generally to queries of arbitrary length $|Q| \geq 1$.

The λ parameter in Eq. 2 has the following role. If $\lambda = \infty$, the refined query is the same as the initial explicit query (with no refinement) because $0 \leq m_i < 1$ and thus all keyword weights are zero. By setting λ to 0, the query is like the one used in the Watson system [2], giving the same weight to the query words and to the keywords representing the local context. Because the keywords are related to topics that have various relevance values to the explicit query, we will set the intermediate value $\lambda = 1$ in our experiments, to weigh each keyword based on its relevance to the topics of the query. The value of λ could be optimized if more training data were available.

To illustrate the terms extracted by each refined query $RQ(\lambda)$ and clarify the role of λ parameter, we consider an example from one of the queries in our dataset, using the ASR transcript of the conversation fragment presented

in the Appendix of this paper. The query is: "I need more information about LCD", therefore it bears on the acronym "LCD". The keywords extracted by our method [3, 4] for this fragment are the following ones: $C = \{$'interface', 'design', 'decision', 'recap', 'user', 'control', 'final', 'remote', 'discuss', 'sleek', 'snowman'$\}$, where three keywords ('recap', 'sleek', and 'snowman') are in fact ASR noise.

The proposed method for refining the query, $RQ(1)$ from Eq. 2 with $\lambda = 1$, assigns in this particular example a weight of zero to keywords unrelated to the conversation topics, and to those due to ASR noise as well. Therefore, the corresponding expanded query is: $RQ(1) = \{$('lcd', 1.0), ('control', 0.7), ('remote', 0.4), ('design', 0.1), ('interface', 0.1), ('user', 0.1)$\}$. These values are obtained using the cosine similarity in the topic space from Eq. 1, and are based on a summation of the importance of the respective keyword and of the query in each of the dimensions of the topic space, which are uneasy to exemplify as they are not easily interpretable [33]. $RQ(0)$ assigns a weight 1 to each keyword of the list C and uses all of them for expansion, regardless of their importance to the query. Therefore, the expanded query contains many irrelevant words. Finally, $RQ(\infty)$ does not expand the query at all, so the query remains only 'lcd', without any additional information.

3.2. Selective Query Expansion using Semantic Information

While words from the local context of the query are potentially important in helping to disambiguate it, we aim in this second stage to expand this list even further, focusing on expanding the search terms that are not found in relevant documents, probably because synonyms or alternative names are used. Hence, our second stage in query expansion starts with a predictive analysis to select search terms which likely lead to vocabulary mismatch, as follows.

Considering the initial set of results from the first stage, we look for search terms from the initial query which are not present in the top k retrieved documents in the ranked list obtained when running the query Q with the expansion terms from the local context, obtained as described above. This happens

likely because the actual use of a concept name (surface form) in a document differs from the query term chosen by the user or those retrieved from the conversational context. These terms are selected as problematic query terms or vocabulary mismatches.

The presence or absence of each query term is checked in the 'title' and 'content' fields of the top 15 documents (Wikipedia pages) retrieved by the $RQ(1)$ method. If the query term is present in fewer than half of the retrieved documents, we consider it a vocabulary mismatch. To address this problem, we use two alternative methods to expand the problematic query terms, inspired by our previous work in information monitoring [34].

Our first selective query expansion method, noted $RQ(1)$-*SQE-WN*, uses synonyms from WordNet (hence the 'WN' notation). We expand the top five terms from the parametrized refined query $RQ(1)$ (as defined in Eq. 2) which are marked as problematic ones, using the synsets extracted from the WordNet semantic dictionary [5].

The second selective query expansion method, noted $RQ(1)$-*SQE-WV*, finds related terms based on their semantic relationships using low-dimensional vector representations of words, also known as neural word embeddings. We learn first the word embeddings using the Skip-Gram with Negative Sampling (SGNS) algorithm of word2vec [6]. The SGNS technique was shown to perform better than or similar to state-of-the-art methods such as distributional similarity methods and SVM on word similarity tasks [35]. The SGNS model is trained on the English Wikipedia, with 20 negative samples and a context sample size c set to 5. We use the publicly available implementation of SGNS from the Gensim toolkit [36].

SGNS models the co-occurrence of words surrounding a current word w_t within a context window of size c, centered on w_t, which is noted $w_{t-c} : w_{t+c}$. The objective function of SGNS is as follows:

$$\mathcal{L} = \sum_{t=1}^{T} \log p(w_{t-c} : w_{t+c} | w_t) \tag{3}$$

The model has a simplifying assumption when modeling the probability distri-

bution of the contextual words $w_{t-c} : w_{t+c}$. Namely, it considers them independent given the current word w_t, in other words it does not exploit the word order, assuming that the surrounding words are equally important, thus leading to the following equation:

$$p(w_{t-c} : w_{t+c}|w_t) = \prod_{-c \leq j \leq c, j \neq 0} p(w_{t+j}|w_t) \tag{4}$$

The objective is trained in an online fashion using stochastic gradient updates over the observed pairs in the corpus. Then, the global objective is normalized by summing over all the observed (w, c) pairs in the corpus as shown in Eq. 5.

$$\mathbb{P}(w_{t+j} : w_t) = \frac{\exp(\overrightarrow{v}^T_{w_t} \cdot \overrightarrow{v}'_{w_{t+j}})}{\sum_{w=1}^{W} \exp(\overrightarrow{v}^T_{w_t} \cdot \overrightarrow{v}'_{w})} \tag{5}$$

Optimizing the objective function makes observed word context pairs have similar embeddings and unobserved pairs are thrown in random directions in the embedding space. This leads to learning similar word embeddings to words drawn from a local context.

We then calculate the weighted average for the projection weight vectors of the first five problematic query terms (defined as above). Then, the cosine similarity between the mean of the projection weight vectors of the problematic query terms and the vectors of each word in the model is computed. Finally, we select the top-5 most similar words according to the calculated cosine similarity and use them for query expansion.

To illustrate the terms extracted by two selective query expansion approaches using semantic information, we consider an example from one of the queries in our dataset. The query bears again on the acronym "LCD" but with a different conversation fragment than the one presented in the Appendix. The list of keywords extracted for this fragment is: $C = \{$'frequency', 'feedback', 'tft', 'channel', 'remote', 'interference', 'rf', 'interface', 'speech', 'tv', 'sort'$\}$. The analysis done for SQE marks the following words as candidate expansion terms, due to vocabulary mismatches: 'feedback', 'tft', 'lcd', 'remote', and 'interface'. On the

one hand, the synonyms extracted from WordNet are: 'action', 'activity', 'answer', 'natural', 'process', 'reply', 'response', 'liquid', 'crystal', 'alphanumeric', 'digital', 'display', 'distant', 'outside', 'removed', 'outback', 'port', 'computer', 'circuit', and 'program'. On the other hand, the related words extracted using word2vec are: 'graphical', 'adapter', 'crt', 'raster', 'controller', and 'scsi'.

4. Dataset and Evaluation Methods

Our experiments are conducted on the AREX dataset, for "AMI Requests for Explanations and Relevance Judgments for their Answers", which we constructed and made publicly available at `http://www.idiap.ch/dataset/arex`. The dataset contains a set of explicit queries, inserted at various locations of the conversations in the AMI Meeting Corpus [7], as explained below in Section 4.1. The dataset also includes relevance judgments of about 30 documents retrieved per query, which were gathered via the Amazon Mechanical Turk (AMT) crowdsourcing platform. The procedure of collecting relevance judgments will be described in details in Section 4.2. These relevance judgments will be used as ground truth to evaluate a retrieval system automatically in Section 5.

4.1. Explicit Queries in the Dataset

The AMI Meeting Corpus contains conversations about designing remote controls. We selected it for building our dataset because it is one of the largest multi-party conversational corpora (more than 100 hours) for which manual transcripts and suitable real-time ASR systems exist. Often in the discussions, participants mention acronyms, which are a good target for building systematic clarification questions, as they can be spotted automatically. Moreover, acronyms are one of the items which are likely to require explanations because of their potential ambiguity, and several questions in the AMI Corpus already bear upon acronyms. Although the broad domain of the corpus is fixed (and could even be used as knowledge for answering the queries), our goal is to leverage only the local topics, which are quite diverse [26], so that our solution advances the state of the art for unrestricted conversations.

Our dataset contains explicit queries with the time of their occurrence in the AMI Corpus. Since the number of naturally-occurring queries in the corpus is insufficient for evaluating our system, we artificially generated and inserted a number of queries about acronyms (though our query expansion technique is applicable to any explicit query), using the following procedure. Initially, utterances containing an acronym X are automatically detected. Then, we formulate explicit queries such as "I need more information about X", and insert them after the utterances containing the acronym (see for instance the example in the Appendix).

Seven acronyms, all-but-one related to the domain of remote controls, are considered: *LCD* (liquid-crystal display), *VCR* (videocassette recorder), *PCB* (printed circuit board), *TFT* (thin-film-transistor liquid-crystal display), *NTSC* (National Television System Committee), *IC* (integrated circuit), and *RSI* (repetitive strain injury). These acronyms occur 74 times in the AMI Corpus and are preceded by 74 different conversation fragments in our dataset. Therefore, AREX contains a total of 74 explicit queries and transcripts of conversation fragments.

We used both manual and ASR transcripts of the fragments from the AMI Corpus in our experiments. The ASR transcripts were generated by the AMI real-time ASR system for meetings [27], with an average word error rate (WER) of 36%. In addition, for experimenting with a variable range of WER values, we have simulated the potential speech recognition mistakes as in [4], by applying to the manual transcripts of these conversation fragments three different types of ASR noise: deletion, insertion and substitution. In a systematic manner, i.e. altering all occurrences of a word type, we randomly selected the conversation words, as well as the words to be inserted, from the vocabulary of the English Wikipedia. The simulated ASR noise percentage varied from 10% to 30%, because the best recognition accuracy reaches around 70% in conversational environments [37]. However, noise was never applied to the explicit query itself.

4.2. Evaluation Using the Dataset

To produce ground truth relevance judgments, we follow a classic approach for evaluating information retrieval [38] based on the pooling of several retrieval systems. We build a reference set of retrieval results by merging the lists of the top 10 retrieval results from four different query expansion methods used to answer the queries. Three out of four query expansion methods were described in Section 3.1, namely $RQ(0)$, $RQ(1)$ and $RQ(\infty)$. For the first two, we have limited the weighting to the first 10 keywords extracted from each fragment, following several previous studies [8], thus speeding up query processing. The fourth one builds a query which consists of only the keywords extracted from the conversation fragments, with no words from the queries. The main role of this method is to extend the variety of documents to be rated, and as it generally leads to irrelevant documents (negative examples), it will not be evaluated below.

The retrieval results are obtained by the Apache Lucene search engine over the English Wikipedia. We found that each explicit query had at least 31 different results for all the 74 fragments, and we decided to limit the reference set to 31 documents for each query. Each conversation fragment preceding a query is set at about 400 words long, for reasons that we will analyze empirically in Section 5.1.

We designed a set of tasks to gather relevance judgments from human subjects. We showed to the subjects the transcript of a conversation fragment ending with the query: "I need more information about X" with 'X' being one of the acronyms considered here. This was followed by a control question about the content of the conversation, and then by the list of 31 document results that we had gathered. The human subjects (i.e. judges) had to decide on the relevance value of each document by selecting one of the three options among 'irrelevant', 'somewhat relevant' and 'relevant' (noted below as $A = \{a_0, a_1, a_2\}$). In other words, the subjects evaluated whether each result is relevant to the explicit query, i.e. whether it clarifies the term on which the query bears. Their answers represent the ground truth to which the outputs of systems will be

compared.

We collected judgments for the 74 explicit queries of our dataset (31 documents each) from 10 subjects per document. The tasks were crowdsourced via AMT, each judgment becoming a "human intelligence task" (HIT). For qualification control, we only accepted subjects with greater than 95% approval rate and with more than 1000 previously approved HITs, and we only kept answers from the subjects who answered correctly the control questions.

We applied furthermore a qualification control factor to the human judgments, inspired from our previous work [39], in order to reduce the impact of "undecided" cases, inferred from the low agreement of the subjects. We computed the following measure of the uncertainty of subjects regarding the relevance of document j:

$$H_{tj} = -\sum_{a \in A} (s_{tj}(a) \ln(s_{tj}(a)) / \ln |A|) \tag{6}$$

where $s_{tj}(a)$ is the proportion in which the 10 subjects have selected each of the allowed options $a \in A$ for the document j and the conversation fragment t. Then, the relevance value assigned to each option a is computed as $s'_{tj}(a) = s_{tj}(a) \cdot (1 - H_{tj})$, i.e. the raw score weighted by the subjects' uncertainty.

To score a new list of documents, we use the ground truth relevance of each document in the reference set, weighted by the subjects' uncertainty. We then measure the mean average precision (MAP) at rank n of a candidate document result list. We start by computing gr_{tj}, the global relevance value for the conversation fragment t and the document j by giving a weight of 2 for each "relevant" answer (a_2) and 1 for each "somewhat relevant" answer (a_1).

$$gr_{tj} = \frac{s^{'}_{tj}(a_1) + 2s^{'}_{tj}(a_2)}{s^{'}_{tj}(a_0) + s^{'}_{tj}(a_1) + 2s^{'}_{tj}(a_2)} \tag{7}$$

Then we calculate $\mathrm{AveP}_{tk}(n)$ the Average Precision at rank n for the conversation fragment t and the candidate list of results of a system k as follows:

$$\mathrm{AveP}_{tk}(n) = \sum_{i=1}^{n} \mathrm{P}_{tk}(i) \triangle r_{tk}(i) \tag{8}$$

where $\mathrm{P}_{tk}(i) = \sum_{c=1}^{i} gr_{tl_{tk}(c)}/i$ is the precision at cut-off i in the list of results l_{tk}, $\triangle r_{tk}(i) = gr_{tl_{tk}(i)} / \sum_{j \in l_t} gr_{tj}$ is the change in recall from document in rank $i-1$ to rank i over the list l_{tk}, and l_t is the reference set for fragment t.

To conclude, we compute $\mathrm{MAP}_k(n)$, i.e. the MAP score at retrieval rank position n for a system k by averaging the Average Precision of all the queries at rank n as follows, where $|T|$ is the number of queries.

$$\mathrm{MAP}_k(n) = \sum_{t=1}^{|T|} \frac{\mathrm{AveP}_{t,k}(n)}{|T|} \tag{9}$$

Finally, we can compare two lists of documents obtained by two systems k_1 and k_2 by using the improvement percentage of the relative MAP score at rank n , defined as follows:

$$\%RelativeScore_{k_1,k_2}(n) = \frac{\mathrm{MAP}_{k_1}(n) - \mathrm{MAP}_{k_2}(n)}{\mathrm{MAP}_{k_2}(n)} \times 100. \tag{10}$$

Therefore, in the experiments below, the improvement or degradation of one system with respect to another one will be measured using the ratio from the above equation. For instance, if a system k_1 has a MAP score (Eq. 9) of 0.5 and a second system k_2 has a MAP score of 0.4, then the improvement of the first with respect to the second one is 25%. – An implementation of this metric is distributed with the AREX dataset.

4.3. Robustness against ASR Noise

We also compare below the two contextual expansion methods, $RQ(0)$ and $RQ(1)$, in terms of the proportion of noisy keywords that each method adds to the refined queries. This proportion is computed by summing up the weight value of the keywords used for query refinement that are in fact ASR errors (their set is noted N_j), normalized by the sum of the weight value of all keywords used for the refinement of the query Q, as follows:

$$pn_Q = \frac{\sum_{kw_i \in (C_Q \cap N_Q)} m_i^{\lambda}}{\sum_{kw_i \in C_Q} m_i^{\lambda}} \times 100\% \tag{11}$$

5. Experimental Results

We provide in this section experimental evidence showing that our proposal outperforms baseline or previous methods for answering spoken clarification queries, including the previous attempt to leverage contextual information from the conversation [2]. Namely, we compare the weighted query expansion methods (introduced in Section 3.1) and their enhancement using selective query expansion (introduced in Section 3.2) against previous methods, in terms of their capacity to retrieve documents that are considered by users as relevant clarifications of the query term. Following a classic information retrieval approach, when comparing results, we consider also the rank or position of each document in the result list: in other words, the goal is to include more relevant documents at earlier positions (higher ranks) in the list.

We use the dataset and the evaluation metrics defined in Section 4, and experiment with both human-made transcripts and ASR output. Our query set contains 74 queries bearing on acronyms (see Section 4.1). Each query follows a conversation fragment, which represents its local context; therefore, there are as many queries as conversation fragments in the dataset, though some queries may bear on the same term. We use one third of these queries (25 out of 74) as the development set on which we tune the parameters of our proposed methods. The remaining 49 queries form our test set, on which we report the results of our evaluation.

We examine the three methods for query expansion presented in Section 3.1. We start by studying the role of the λ parameter in Eq. 2. The $RQ(\infty)$ method actually uses only words from the query, with no refinement. The $RQ(0)$ method refines explicit queries using the approach of the Watson system [2], which corresponds to $\lambda = 0$. The $RQ(1)$ method expands the query with keywords extracted from the conversation fragment based on their topical similarity to the query, and corresponds to $\lambda = 1$ in Eq. 2. This is the first stage of the novel query refinement method proposed in this paper. However, we also evaluate the enhancement of $RQ(1)$ with the SQE-WN and the SQE-WV selective expansion

techniques from Section 3.2. All these methods generate retrieval ranked lists for all the queries in the test set; the ranking of the results will be specifically considered for the evaluation.

We will study the effects of the context window size (i.e., conversation fragment length) on query expansion, showing that $RQ(1)$ outperforms $RQ(\infty)$ and $RQ(0)$ regardless of the context size (except for rank position n=1), and that $RQ(1)$-*SQE* (with either WN or WV) outperforms $RQ(1)$ in all cases (Section 5.2). Then, we will compare these methods, using improvement percentage of the relative MAP score (Eq. 10) at various retrieval rank positions n, on manual transcripts (Section 5.2) and on automatic ones (Section 5.3), confirming that $RQ(1)$+*SQE* outperforms the other methods. Finally, we will exemplify the lists of Wikipedia pages retrieved using the queries expanded by different methods in Section 5.4.

5.1. Setting the Length of the Conversation Fragment

We first fix the length of the conversation fragments used in our study. Although this could be set dynamically, and changed based on several parameters like the content of the query or the amount of information in the fragment, for simplicity we decided to set a fixed length below. To find an appropriate value, we computed the sum of the weights assigned to the keywords extracted from each fragment by $RQ(1)$, and averaged them over 25 queries, which were randomly selected from our dataset to serve as a development set. The values obtained from five repetitions of the experiment with fragment lengths varying from 100 to 500 words in increments of 100 were, respectively: 2.14, 2.32, 2.08, 2.08, and 2.08. Since there is no variation among the last three values, we fix the fragment size to 400 words.

5.2. Comparisons on Manual Transcripts

In this section we first study the effect of the conversation fragment length on the retrieval results of the three following methods: $RQ(1)$, $RQ(\infty)$, and $RQ(0)$. The keyword set used for expansion (see Section 3.1) is extracted from

the manual transcript of the conversation fragment accompanying each explicit query of the test set. The fragments have a fixed length per experiment, and we ran our experiments over varying lengths from 100 to 600 words.

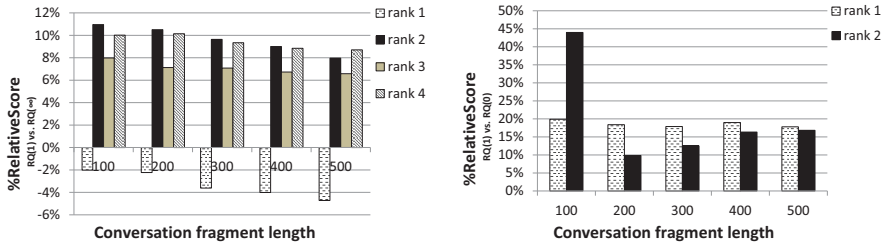

Figure 1: Relative MAP scores (a) of $RQ(1)$ against $RQ(\infty)$ up to rank position 4, and (b) of $RQ(1)$ against $RQ(0)$ up to rank position 2. The scores were obtained using manual transcripts with fragment lengths of 100, 200, 300, 400 and 500 words. $RQ(1)$ outperforms the other two methods, except for $RQ(\infty)$ at rank $n = 1$.

The relative MAP scores of $RQ(1)$ over $RQ(\infty)$ for retrieval rank positions n from 1 to 4 are represented in Figure 1(a). Although $RQ(\infty)$ outperforms $RQ(1)$ at rank position 1, $RQ(1)$ surpasses $RQ(\infty)$ for rank positions 2, 3 and 4. The improvements over $RQ(\infty)$ slightly decrease when conversation fragment length increases, likely because of the topic drift in longer fragments. In fact, when fragment length increases, the proposed method $RQ(1)$ behaves similarly to $RQ(\infty)$ by assigning smaller weight values (close to zero) to the candidate expansion keywords.

The relative MAP scores of $RQ(1)$ over $RQ(0)$ are reported at rank positions $n = 1$ and $n = 2$ in Figure 1(b). We do not report values for lower rank positions, because of the lack of enough relevance judgments for the retrieval results of $RQ(0)$ among the reference set. The improvements over $RQ(0)$ at rank 1 remain approximately constant for different fragment lengths. However, at rank 2, they vary a lot with the length of fragments: the improvement is minimum at fragment length of 200 words, likely because more relevant candidate expansion keywords are present at this length compared to others. The average sum of the weights of the expansion keywords is maximized by our method, $RQ(1)$, at

length 200 words. When smaller or larger fragment lengths are used, the query topics are not completely covered, or the topics in the conversation change respectively. Therefore, the improvement over $RQ(0)$ increases at rank 2 when using length values other than 200 words, thus showing that $RQ(1)$ is more robust to out-of-topic keywords than $RQ(0)$.

The relative MAP scores of RQ(1)-SQE-WN over RQ(1) for different rank positions n from 1 to 4 are illustrated in Figure 2(a). The improvement percentage obtained by $RQ(1)$-*SQE-WN* is lowest at ranks 1 and 2. We hypothesize that this is related to the fact that $RQ(1)$-*SQE-WN* expands a query with all its synonyms and thus it improves the recall but at the expense of lowering the precision at higher ranks (smaller values of n). For $RQ(1)$-*SQE-WN*, the improvement is maximal at a fragment length of 300, where $RQ(1)$-*SQE-WN* obtains a relative improvement of 2% at rank 1 versus a relative improvement of 6% at rank 4. The improvement is minimal at fragment length of 600, which is due to the noisy context words extracted from the conversation fragment for such a large context. Overall, $RQ(1)$-*SQE-WN* obtains an average improvement of 2.4% at rank 1, and of 4.7% at rank 3 over $RQ(1)$ for all fragment lengths, with the maximal improvement obtained at rank 3.

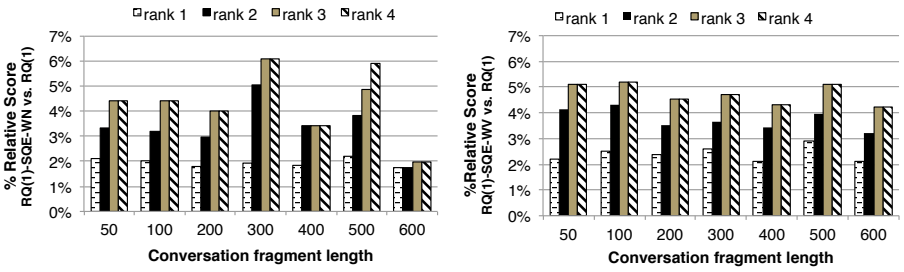

Figure 2: Relative MAP scores (a) of *RQ(1)-SQE-WN* against $RQ(1)$ up to rank position 4, and (b) of *RQ(1)-SQE-WV* against $RQ(1)$ up to rank 4. The scores were obtained using manual transcripts with fragment lengths of 50, 100, 200, 300, 400, 500, and 600 words.

The relative MAP scores of $RQ(1)$-*SQE-WV* over $RQ(1)$ for different rank positions n from 1 to 4 are illustrated in Figure 2(b). $RQ(1)$-*SQE-WV* is more

robust than $RQ(1)$-*SQE-WN* with respect to the variation of the length of the conversation fragment, as the improvement remains considerable when increasing the fragment length. It obtains a relative improvement of 4% over *RQ(1)* at rank 4 for a fragment length of 600 words.

We can see from Figure 2 that both $RQ(1)$-*SQE-WN* and $RQ(1)$-*SQE-WV* outperform $RQ(1)$, as they always obtain positive improvements over $RQ(1)$ on all rank positions and all fragment lengths. Moreover, on average, $RQ(1)$-*SQE-WV* obtains higher improvements compared to $RQ(1)$-*SQE-WN* on all conversation fragment lengths.

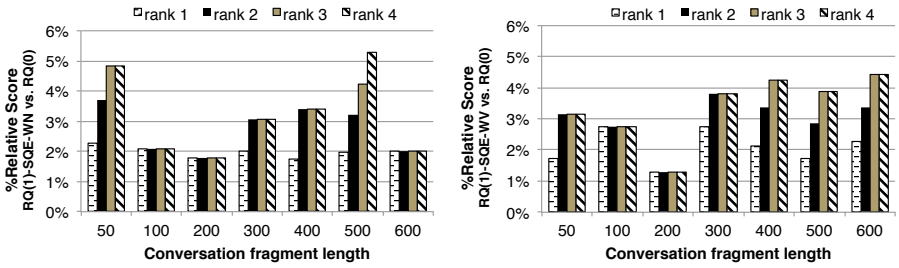

Figure 3: Relative MAP scores of (a) $RQ(1)$-*SQE-WN* against $RQ(0)$ up to rank position 4, (b) of $RQ(1)$-*SQE-WV* against $RQ(0)$ up to rank position 4. The scores were obtained using manual transcripts with fragment lengths of 50, 100, 200, 300, 400, 500, and 600 words.

The relative MAP scores of $RQ(1)$-*SQE-WN* and $RQ(1)$-*SQE-WV* over $RQ(0)$ for retrieval rank positions n from 1 to 4 are represented in Figure 3, showing that both $RQ(1)$-*SQE-WN* and $RQ(1)$-*SQE-WV* obtain superior performance compared to $RQ(0)$ on all rank positions and all fragment lengths. $RQ(1)$-*SQE-WN* obtains maximal improvement of 6% at rank position $n = 4$ for a fragment length of 500 words. The improvements of $RQ(1)$-*SQE-WN* over $RQ(0)$ are approximately the same. They are minimal for fragment lengths of 200, 300 and 600 words, which could be related to incomplete relevance judgments of the results of $RQ(1)$-*SQE-WN*. Actually, $RQ(1)$-*SQE-WN* obtains an average improvement over $RQ(0)$ of 2% at rank $n = 1$ and an average improvement of 3% at ranks $n > 1$ on all fragment lengths. The lowest improvement

is obtained at fragment length 200, which can be related to the noisy context words extracted from the conversation fragment.

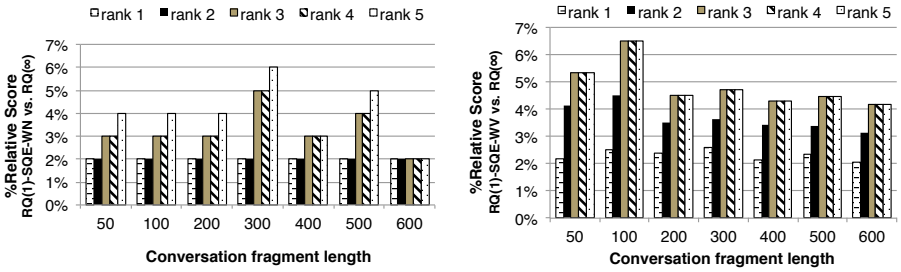

Figure 4: Relative MAP scores of (a) $RQ(1)$-*SQE-WN* against $RQ(\infty)$ up to rank 5, and (b) of $RQ(1)$-*SQE-WV* against $RQ(\infty)$ up to rank 5. The scores were obtained using manual transcripts with fragment lengths of 50, 100, 200, 300, 400, 500, and 600 words.

The relative MAP scores of $RQ(1)$-*SQE-WN* and $RQ(1)$-*SQE-WV* over $RQ(\infty)$ for different ranks n from 1 to 5 are represented in Figure 4, demonstrating the superiority of both $RQ(1)$-*SQE-WN* and $RQ(1)$-*SQE-WV* with respect to $RQ(\infty)$. The improvements obtained by $RQ(1)$-*SQE-WV* are superior to those obtained by $RQ(1)$-*SQE-WN* for all fragment sizes except 300. Finally, $RQ(1)$-*SQE-WV* achieves an improvement of 2% at rank 1 and of 5% at rank 3 for all fragment lengths.

We now study the performance of the proposed query refinement methods on lower retrieval rank positions in the obtained ranked list. To this end, we compare the initial stage of the proposed method, $RQ(1)$, with two previous methods, $RQ(0)$ and $RQ(\infty)$ over the manual transcripts of the queries in the test set, for rank positions n from 1 to 8, with fragments of 400 words preceding each query. The improvements obtained by $RQ(1)$ over the two other method are presented in Figure 5 (for 400 words, the results from Figure 1 are reused in this figure).

The relative MAP scores of $RQ(1)$ over $RQ(\infty)$, except at rank position $n = 1$, demonstrate the significant superiority of $RQ(1)$ over $RQ(\infty)$ (between 7% to 11%) up to rank $n = 6$ on average. There are also on average small

improvements around 2% over $RQ(\infty)$ at rank positions $n = 7$ and 8, because of retrieving the documents which are relevant to both the queries and the fragments by $RQ(\infty)$ (which does not disambiguate the query) at ranks $n = 1, 7$ and 8. The relative MAP scores of $RQ(1)$ over $RQ(0)$ show significant improvements of more than 15% for ranks $n = 1$ and $n = 2$. Although the scores decrease from rank 2, they remain considerably high at around 7%.

Figure 5 shows that $RQ(1)$ is able to achieve consistent improvement over both $RQ(\infty)$ and $RQ(0)$ even when considering a larger portion of the retrieval ranked list, i.e. when increasing retrieval rank position to n=8.

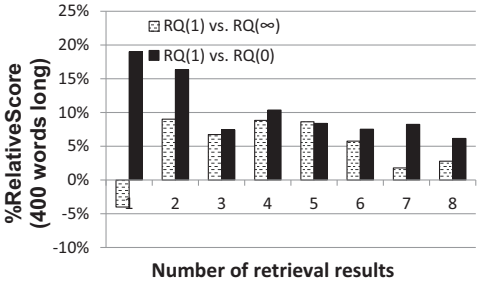

Figure 5: Relative MAP scores of $RQ(1)$ over the two baseline methods $RQ(\infty)$ and $RQ(0)$ up to rank 8, obtained over the manual transcript of the queries in the test set for conversation length of 400 words. $RQ(1)$ surpasses both methods for ranks 2 to 8.

5.3. Comparisons on ASR Transcripts

In this section, we apply the proposed query expansion methods to the ASR transcripts of the conversations from our dataset, in order to consider the effect of ASR noise on the retrieval results of the expanded queries. We experimented with real ASR transcripts with an average word error rate of 36% and with simulated ones with a noise level varying from 10% to 30% (see end of Section 4.1). We computed the average of the scores over five repetitions of the experiment with simulated ASR transcripts, which are randomly generated, and provide below the relative MAP scores of $RQ(1)$ over $RQ(\infty)$ up to rank 3, and over $RQ(0)$ up to rank 2. Moreover, upon manual inspection, we found that there are many relevant documents retrieved in the presence of ASR noise, which have

no judgment in the dataset, because they do not overlap with the 31 documents obtained by pooling four methods.

We compared the two contextual expansion methods, $RQ(0)$ and $RQ(1)$, in terms of the proportion of noisy keywords that each method added to the refined queries. We averaged the values calculated according to Eq. 11 over the 49 explicit queries and the five experimental runs with different random ASR errors. The results shown in Table 1 reveal that the proposed method, $RQ(1)$, is more robust to the ASR noise than $RQ(0)$.

Table 1: Proportion of noisy keywords added to queries depending on ASR noise on $RQ(1)$ and $RQ(0)$. The proportions are computed over 49 explicit queries from the dataset, for a noise level varying from 10% to 30%. $RQ(1)$ is clearly more robust to noise than $RQ(0)$.

ASR noise	10%	20%	30%
RQ(1)	0.78	1.30	2.27
RQ(0)	5.64	12.07	21.07

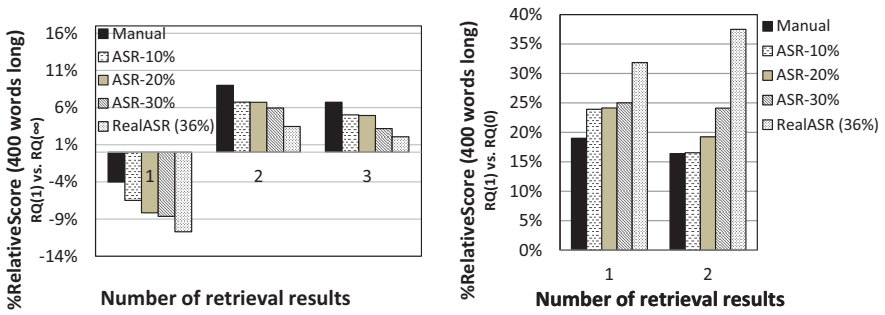

Figure 6: Relative MAP scores of $RQ(1)$ against $RQ(\infty)$ up to rank 3 (a), and against $RQ(0)$ up to rank 2 (b), obtained over the real or simulated ASR transcripts. The results show that $RQ(1)$ outperforms the other two methods.

We also represent the relative scores of $RQ(1)$ over $RQ(0)$ in Figure 6. The improvement over $RQ(0)$ increases when the noise percentage added to the fragments increases, and shows that our method exceeds $RQ(0)$ considerably. Moreover, we compare the retrieval results of $RQ(1)$ and $RQ(\infty)$ (which does not

consider context) in noisy conditions, in Figure 6. Although the improvement over $RQ(\infty)$ slightly decreases with the noise level, $RQ(1)$ still outperforms $RQ(\infty)$ in terms of relevance, and is generally more robust to ASR noise.

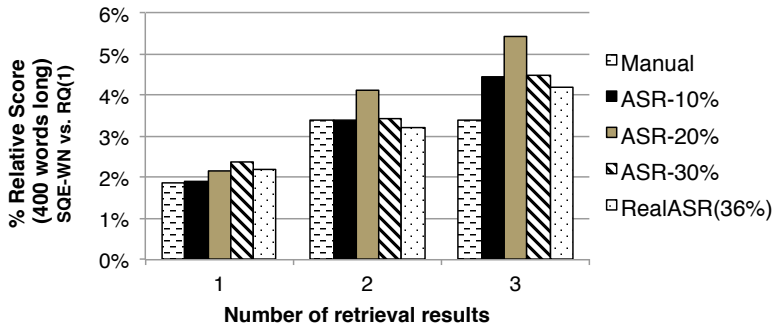

Figure 7: Relative MAP scores of $RQ(1)$-*SQE-WN* against $RQ(1)$ up to rank 3, obtained over the real or simulated ASR transcripts.

Finally, Figure 7 shows the impact of added noise on the performance of $RQ(1)$-*SQE-WN* with respect to $RQ(1)$ for a conversation fragment of 400 words. Increasing the noise does not affect significantly the performance of the $RQ(1)$-*SQE-WN* method.

5.4. Examples of Retrieval Results

To illustrate how $RQ(1)$ surpasses the other techniques, we consider an example from one of the queries of our dataset bearing the acronym "LCD". The terms extracted from this conversation fragment are mentioned at the end of Section 3.1. Table 2 displays the retrieval results obtained for the three methods $RQ(1)$, $RQ(0)$, and $RQ(\infty)$ up to rank 8. All the results of $RQ(1)$ are related to 'liquid-crystal display', which is the correct interpretation of the query, while $RQ(\infty)$ provides three irrelevant documents: 'lowest common denominator' (a mathematic function), 'LCD Soundsystem' (an American dance band), and 'Pakalitha Mosisili' (a politician at Lesotho Congress for Democracy). None of the results provided by $RQ(0)$ addresses 'liquid-crystal display' directly, due to irrelevant keywords added to the query from topics unrelated to

the conversation or from ASR noise.

Table 2: Ranked lists of Wikipedia pages retrieved using $RQ(1)$, $RQ(\infty)$, and $RQ(0)$ for a sample query about "LCD" in the conversation fragment from the Appendix. Results of $RQ(1)$ are the most relevant ones to the query and conversation topics.

RQ(1)	**RQ(∞)**	**RQ(0)**
Liquid-crystal display	Liquid-crystal display	User interface
Backlight	Backlight	X Window System
Liquid-crystal display television	Liquid-crystal display television	Usability
Thin-film transistor	Lowest common denominator	Wii Remote
LCD projector	LCD Soundsystem	Walkman
LG Display	LCD projector	Information hiding
LCD shutter glasses	Pakalitha Mosisili	Screensaver
Universal remote	LG Display	Apple IIc

We provide another series of retrieval examples in Table 3, showing that $RQ(1)$-*SQE-WN* and $RQ(1)$-*SQE-WV* improve over *RQ(1)*. Similar to the previous example, the query bears on the acronym "LCD" (it can be glossed as: "I need more information about LCD") but with a different conversation fragment. The terms extracted from this conversation fragment are presented at the end of Section 3.2.

The retrieval results obtained for this query by the *RQ(1)*, $RQ(1)$-*SQE-WN* and $RQ(1)$-*SQE-WV* methods are displayed in Table 3, ordered by increased relevance from left to right. The results of $RQ(1)$-*SQE-WN* and $RQ(1)$-*SQE-WV* appear indeed to be more relevant to the query than those of $RQ(1)$, and also than those of $RQ(\infty)$ or $RQ(0)$, not shown here. For instance, both $RQ(1)$-*SQE-WN* and $RQ(1)$-*SQE-WV* propose at rank 1 the relevant Wikipedia page 'AU Optronics', which is one of the leading LCD monitor manufacturers. They also find 'FPD-Link' which stands for 'Flat Panel Display Link', the original 1996 high-speed digital video interface for LCD displays. 'EPLaR' (Electronics

on Plastic by Laser Release) is found at rank 8 by *RQ*(1)-*SQE-WV*, and represents a method for manufacturing flexible LCD displays. The correct expansion of the 'LCD' acronym in context is ranked 4th by *RQ*(1)-*SQE-WN*.

Moreover, in this example, *RQ*(1)-*SQE-WN* and *RQ*(1)-*SQE-WV* retrieve relevant Wikipedia pages that do not have judgments in our dataset (such as 'FPD-Link' or 'Samsung Corning Precision Glass'), hence they cannot be scored numerically by our method. Had we performed an evaluation of the actual results (which must be repeated whenever methods change), the obtained scores for the SQE methods would have been even higher.

Table 3: Examples of retrieved Wikipedia pages (ranked lists) using five methods. Ranked lists of Wikipedia pages retrieved using *RQ*(1), *RQ*(1)-*SQE-WN* and *RQ*(1)-*SQE-WV* for a query about "LCD" (on a different conversation fragment than Table 2 above). The SQE methods appear to outperform *RQ*(1).

RQ(1)	**RQ(1)-SQE-WN**	**RQ(1)-SQE-WV**
Composite video	AU Optronics	AU Optronics
Aliasing	Samsung Corning Precision Glass	Native resolution
Thin film transistor liquid crystal display	FPD-Link	Carputer
Klystron	Liquid crystal display	Samsung Corning Precision Glass
Sideband	Super-twisted nematic display	FPD-Link
RF modulator	Thin film transistor liquid crystal display	Thin film transistor liquid crystal display
Spectrum analyzer	Active-matrix liquid crystal display	PowerBook G3
Thin-film transistor	LG Display	EPLaR

6. Conclusion

In this paper, we have proposed an approach to query refinement through expansion, intended for an information retrieval assistant that can answer spoken clarification questions during a meeting. In this framework, we have shown how to leverage the conversational context preceding the query, obtained using ASR, in order to extract and weigh expansion terms that refine the query and improve the relevance of the results. We have proposed a two-stage approach, first weighing the expansion keywords extracted from the context ($RQ(1)$) and then adding further expansion terms obtained either using WordNet ($RQ(1)$-SQE-WN) or a trained word2vec model ($RQ(1)$-SQE-WV).

The proposed methods outperformed several baselines for contextual query refinement, over both manual and ASR transcripts, and $RQ(1)$-SQE-WV slightly outperformed $RQ(1)$-SQE-WN. The results also demonstrated that the proposed method is robust to various ASR noise levels and to the length of the conversation fragment used for expansion. The AREX dataset that enabled these experiments is made public at `www.idiap.ch/dataset/arex`, and can be used for future comparisons of conversational query-based retrieval systems.

Although all the results are obtained using English conversations, documents, and semantic resources, the methods presented in this paper can be easily ported to other languages. If no equivalent of WordNet is available, then only word2vec can be used in the selective query expansion stage, requiring only unstructured document resources for training.

Several research questions should be addressed in the future. One of them is determining automatically the most appropriate size of the context, i.e. conversation fragment, to be considered for query expansion, likely based on topical coherence. Another important question is the generalization of the present methods, but also testing data, to queries bearing on complex terms. Such queries could be possibly elicited from users in an appropriate setting, to obtain more naturally-occurring queries. To make the system operational, a solution should be designed for the detection of queries in the real-time ASR output,

possibly using a specific code name to address the system and indicate that a query is formulated.

Finally, as we proposed earlier for non-query-based recommender systems [26, Chapter 8], the end-to-end system should be evaluated in experiments with human subjects. This requires the definition of an appropriate scenario that encourages users to use spoken queries during a task-oriented conversation, e.g. for brainstorming. Using an A/B testing approach, such experiments could confirm the advantages of using context to refine spoken queries with the methods presented in this paper.

Acknowledgments

The authors are grateful to the Swiss National Science Foundation (SNSF) for its financial support through the IM2 NCCR on Interactive Multimodal Information Management (see `www.im2.ch`), to the Hasler Foundation for the REMUS project (n. 13067, Re-ranking Multiple Search Results for Just-in-Time Document Recommendation), and to the Swiss Commission for Technology and Innovation (CTI/KTI).

References

[1] A. A. Alidin, F. Crestani, Context modelling for Just-In-Time mobile information retrieval (JIT-MobIR), Pertanika Journal of Science & Technology 21 (1) (2013) 227–238.

[2] J. Budzik, K. J. Hammond, User interactions with everyday applications as context for just-in-time information access, in: Proceedings of the 5th International Conference on Intelligent User Interfaces (IUI), 2000, pp. 44–51.

[3] M. Habibi, A. Popescu-Belis, Diverse keyword extraction from conversations, in: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, 2013, pp. 651–657.

[4] M. Habibi, A. Popescu-Belis, Keyword extraction and clustering for document recommendation in conversations, IEEE/ACM Transactions on Audio, Speech, and Language Processing (TASLP) 23 (4) (2015) 746–759.

[5] G. A. Miller, WordNet: a lexical database for English, Communications of the ACM 38 (11) (1995) 39–41.

[6] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: Advances in Neural Information Processing (NIPS), 2013, pp. 3111–3119.

[7] J. Carletta, Unleashing the killer corpus: Experiences in creating the multi-everything AMI Meeting Corpus, Language Resources and Evaluation Journal 41 (2) (2007) 181–190.

[8] C. Carpineto, G. Romano, A survey of automatic query expansion in information retrieval, ACM Computing Surveys (CSUR) 44 (1) (2012) 1–50.

[9] R. Attar, A. S. Fraenkel, Local feedback in full-text retrieval systems, Journal of the ACM (JACM) 24 (3) (1977) 397–417.

[10] J. Xu, W. B. Croft, Query expansion using local and global document analysis, in: Proceedings of the 19th Annual International ACM SIGIR Conference on Research and development in IR, 1996, pp. 4–11.

[11] S. E. Robertson, S. Walker, M. Beaulieu, P. Willett, Okapi at TREC-7: automatic ad hoc, filtering, VLC and interactive track, NIST Special Publication SP (1999) 253–264.

[12] C. Carpineto, R. De Mori, G. Romano, B. Bigi, An information-theoretic approach to automatic query expansion, ACM Transactions on Information Systems (TOIS) 19 (1) (2001) 1–27.

[13] J. Bai, D. Song, P. Bruza, J.-Y. Nie, G. Cao, Query expansion using term relationships in language models for information retrieval, in: Proceedings

of the 14th ACM International Conference on Information and Knowledge Management (CIKM), 2005, pp. 688–695.

[14] J. Xu, W. B. Croft, Improving the effectiveness of information retrieval with local context analysis, ACM Transactions on Information Systems (TOIS) 18 (1) (2000) 79–112.

[15] F. Diaz, D. Metzler, Improving the estimation of relevance models using large external corpora, in: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and development in IR, 2006, pp. 154–161.

[16] P. A. Chirita, C. S. Firan, W. Nejdl, Personalized query expansion for the Web, in: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and development in IR, 2007, pp. 7–14.

[17] L. A. F. Park, K. Ramamohanarao, Query expansion using a collection dependent probabilistic latent semantic thesaurus, in: Advances in Knowledge Discovery and Data Mining, Springer, 2007, pp. 224–235.

[18] J. J. Rocchio, Relevance feedback in information retrieval, in: G. Salton (Ed.), The SMART Retrieval System: Experiments in Automatic Document Processing, Prentice-Hall, Englewood Cliffs, NJ, 1971, Ch. 14, pp. 313–323.

[19] G. Salton, C. Buckley, Improving retrieval performance by relevance feedback, Readings in Information Retrieval 24 (1997) 5.

[20] V. Lavrenko, W. B. Croft, Relevance based language models, in: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and development in IR, 2001, pp. 120–127.

[21] J. Bhogal, A. Macfarlane, P. Smith, A review of ontology based query expansion, Information Processing and Management 43 (4) (2007) 866–886.

[22] L. Zhao, J. Callan, Term necessity prediction, in: Proceedings of the 19th ACM Conference on Information and Knowledge Management (CIKM), 2010, pp. 259–268.

[23] L. Zhao, J. Callan, Automatic term mismatch diagnosis for selective query expansion, in: Proceedings of the 35th Annual International ACM SIGIR Conference on Research and development in IR, 2012, pp. 515–524.

[24] A. Popescu-Belis, E. Boertjes, J. Kilgour, P. Poller, S. Castronovo, T. Wilson, A. Jaimes, J. Carletta, The AMIDA Automatic Content Linking Device: Just-in-time document retrieval in meetings, in: Proceedings of Machine Learning for Multimodal Interaction (MLMI), Utrecht, 2008, pp. 272–283.

[25] A. Popescu-Belis, M. Yazdani, A. Nanchen, P. N. Garner, A speech-based just-in-time retrieval system using semantic search, in: Proceedings of the 49th Annual Meeting of the ACL, Demonstrations Session, 2011, pp. 80–85.

[26] M. Habibi, Modeling users' information needs in a document recommender for meetings, Ph.D. thesis, École Polytechnique Fédérale de Lausanne, n. 6760 (November 2015).

[27] P. N. Garner, J. Dines, T. Hain, A. El Hannani, M. Karafiat, D. Korchagin, M. Lincoln, V. Wan, L. Zhang, Real-time ASR from meetings, in: Proceedings of the 10th Annual Conference of the International Speech Communication Association, 2009, pp. 2119–2122.

[28] D. Bohus, E. Horvitz, Models for multiparty engagement in open-world dialog, in: Proceedings of the 10th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGdial), 2009, pp. 225–234.

[29] D. Wang, D. Hakkani-Tur, G. Tur, Understanding computer-directed utterances in multi-user dialog systems, in: Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2013, pp. 8377–8381.

[30] D. M. Blei, A. Y. Ng, M. I. Jordan, Latent Dirichlet Allocation, Journal of Machine Learning Research 3 (2003) 993–1022.

[31] A. K. McCallum, MALLET: A machine learning for language toolkit, http://mallet.cs.umass.edu (2002).

[32] M. D. Hoffman, D. M. Blei, F. Bach, Online learning for Latent Dirichlet Allocation, in: Proceedings of 24th Annual Conference on Neural Information Processing Systems (NIPS), 2010, pp. 856–864.

[33] J. Chang, J. L. Boyd-Graber, S. Gerrish, C. Wang, D. M. Blei, Reading tea leaves: How humans interpret topic models, in: Proceedings of the 23rd Annual Conference on Neural Information Processing Systems (NIPS), 2009, pp. 288–296.

[34] P. Mahdabi, A. Popescu-Belis, Comparing two strategies for query expansion in a news monitoring system, in: Proceedings of the 21st International Conference on Applications of Natural Language to Information Systems (NLDB), 2016.

[35] O. Levy, Y. Goldberg, I. Dagan, Improving distributional similarity with lessons learned from word embeddings, Transactions of the Association for Computational Linguistics 3 (2015) 211–225.

[36] R. Řehůřek, P. Sojka, Software framework for topic modelling with large corpora, in: Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks, 2010, pp. 45–50.

[37] T. Hain, L. Burget, J. Dines, P. N. Garner, A. El Hannani, M. Huijbregts, M. Karafiat, M. Lincoln, V. Wan, The AMIDA 2009 meeting transcription system, in: Proceedings of INTERSPEECH, 2010, pp. 358–361.

[38] E. M. Voorhees, D. K. Harman (Eds.), TREC: Experiment and Evaluation in Information Retrieval, MIT Press, Cambridge, MA, 2005.

[39] M. Habibi, A. Popescu-Belis, Using crowdsourcing to compare document recommendation strategies for conversations, in: Proceedings of the RecSys Workshop on Recommendation Utility Evaluation: Beyond RMSE (RUE 2011), 2012, pp. 15–20.

Appendix: Transcript of a Conversation Fragment from the AMI Meeting Corpus

We provide below a 150-word fragment of the ASR from a conversation of the AMI Corpus (segmented by the ASR into utterances), which is used as an example in this paper. The discussion is about designing a remote control, and a query appears at the end of the fragment from the AREX dataset. The document results retrieved for the query by three methods are given in Table 2.

A: Okay well .. All sacked .. Right .. Oh i see a kind of detailed design meeting .. Um .. We're gonna discuss the the look-and-feel design user interface design and .. We're gonna evaluate the product .. And .. For .. The end result of this meeting has to be a decision on the details of this remote control like a sleek final decision .. Uh-huh .. Um i'm then i'm gonna have to specify the final design .. In the final report ..

B: Yeah .. So um just from from last time .. To recap .. So we're gonna have a snowman shaped remote control with no LCD display new need for tap bracket so if you're gonna be kinetic power and battery .. Uh with rubber buttons maybe park lighting the buttons with um .. Internal LEDs to shine through the casing .. Um hopefully a job down and incorporating the slogan somewhere as well I think i missed .. Okey .. Um so .. Uhuh .. If you want to present your prototype .. Go ahead ..

C [inserted]: I need more information about LCD.