Keyword Extraction and Clustering for Document Recommendation in Conversations

Maryam Habibi and Andrei Popescu-Belis

***Abstract*—This paper addresses the problem of keyword extraction from conversations, with the goal of using these keywords to retrieve, for each short conversation fragment, a small number of potentially relevant documents, which can be recommended to participants. However, even a short fragment contains a variety of words, which are potentially related to several topics; moreover, using an automatic speech recognition (ASR) system introduces errors among them. Therefore, it is difficult to infer precisely the information needs of the conversation participants. We first propose an algorithm to extract keywords from the output of an ASR system (or a manual transcript for testing), which makes use of topic modeling techniques and of a submodular reward function which favors diversity in the keyword set, to match the potential diversity of topics and reduce ASR noise. Then, we propose a method to derive multiple topically separated queries from this keyword set, in order to maximize the chances of making at least one relevant recommendation when using these queries to search over the English Wikipedia. The proposed methods are evaluated in terms of relevance with respect to conversation fragments from the Fisher, AMI, and ELEA conversational corpora, rated by several human judges. The scores show that our proposal improves over previous methods that consider only word frequency or topic similarity, and represents a promising solution for a document recommender system to be used in conversations.**

***Index Terms*—Document recommendation, information retrieval, keyword extraction, meeting analysis, topic modeling.**

I. Introduction

HUMANS are surrounded by an unprecedented wealth of information, available as documents, databases, or multimedia resources. Access to this information is conditioned by the availability of suitable search engines, but even when these are available, users often do not initiate a search, because their current activity does not allow them to do so, or because they are not aware that relevant information is available. We adopt in this paper the perspective of *just-in-time retrieval*, which answers this shortcoming by spontaneously recommending documents that are related to users' current activities. When these activities are mainly conversational, for instance when users participate in a meeting, their information needs can be modeled as implicit queries that are constructed in the background from the pronounced words, obtained through real-time automatic speech recognition (ASR). These implicit queries are used to retrieve and recommend documents from the Web or a local repository, which users can choose to inspect in more detail if they find them interesting.

The focus of this paper is on formulating implicit queries to a just-in-time-retrieval system for use in meeting rooms. In contrast to explicit spoken queries that can be made in commercial Web search engines, our just-in-time-retrieval system must construct implicit queries from conversational input, which contains a much larger number of words than a query. For instance, in the example discussed in Section V-B below, in which four people put together a list of items to help them survive in the mountains, a short fragment of 120 seconds contains about 250 words, pertaining to a variety of domains, such as 'chocolate', 'pistol', or 'lighter'. What would then be the most helpful 3–5 Wikipedia pages to recommend, and how would a system determine them?

Given the potential multiplicity of topics, reinforced by potential ASR errors or speech disfluencies (such as 'whisk' in this example), our goal is to maintain multiple hypotheses about users' information needs, and to present a small sample of recommendations based on the most likely ones. Therefore, we aim at extracting a relevant and diverse set of keywords, cluster them into topic-specific queries ranked by importance, and present users a sample of results from these queries. The topic-based clustering decreases the chances of including ASR errors into the queries, and the diversity of keywords increases the chances that at least one of the recommended documents answers a need for information, or can lead to a useful document when following its hyperlinks. For instance, while a method based on word frequency would retrieve the following Wikipedia pages: 'Light', 'Lighting', and 'Light My Fire' for the above-mentioned fragment, users would prefer a set such as 'Lighter', 'Wool' and 'Chocolate'.

Relevance and diversity can be enforced at three stages: when extracting the keywords; when building one or several implicit queries; or when re-ranking their results. The first two approaches are the focus of this paper. Our recent experiments with the third one, published separately [1], show that re-ranking of the results of a single implicit query cannot improve users' satisfaction with the recommended documents. Previous methods for formulating implicit queries from text (discussed extensively in Section II-B) rely on word frequency

Manuscript received September 25, 2013; revised September 01, 2014; accepted February 06, 2015. Date of publication February 19, 2015; date of current version March 06, 2015. This work was supported in part by the Swiss National Science Foundation (SNSF) through the IM2 NCCR on Interactive Multimodal Information Management (www.im2.ch) and in part by the Hasler Foundation for the REMUS project (n. 13067, Re-Ranking Multiple Search Results for Just-in-Time Document Recommendation). The associate editor coordinating the review of this manuscript and approving it for publication was Dr. James Glass.

The authors are with the Idiap Research Institute and École Polytechnique Fédérale de Lausanne (EPFL), 1920 Martigny, Switzerland (e-mail: maryam.habibi@idiap.ch; andrei.popescu-belis@idiap.ch).

Digital Object Identifier 10.1109/TASLP.2015.2405482

or TFIDF weights to rank keywords and then select the highest ranking ones [2], [3]. Other methods perform keyword extraction by using topical similarity [4], [5], [6], but do not set a topic diversity constraint.

In this paper, we introduce a novel keyword extraction technique from ASR output, which maximizes the coverage of potential information needs of users and reduces the number of irrelevant words. Once a set of keywords is extracted, it is clustered in order to build several topically-separated queries, which are run independently, offering better precision than a larger, topically-mixed query. Results are finally merged into a ranked set before showing them as recommendations to users.

The paper is organized as follows. In Section II-A we review existing just-in-time-retrieval systems and the policies they use for query formulation. In Section II-B we discuss previous methods for keyword extraction. In Section III we describe the proposed technique for implicit query formulation, which relies on a novel topic-aware diverse keyword extraction algorithm (III-A) and a topic-aware clustering method (III-B). Section IV introduces the data and our method for comparing the relevance of sets of keywords or recommended documents using crowdsourcing. In Section V we present and discuss the experimental results on keyword extraction and document recommendation. We also exemplify the results on one conversation fragment given in Appendix A.

II. State of the Art: Just-in-Time Retrieval and Keyword Extraction

Just-in-time retrieval systems have the potential to bring a radical change in the process of query-based information retrieval. Such systems continuously monitor users' activities to detect information needs, and pro-actively retrieve relevant information. To achieve this, the systems generally extract implicit queries (not shown to users) from the words that are written or spoken by users during their activities. In this section, we review existing just-in-time-retrieval systems and methods used by them for query formulation. In particular, we will introduce our Automatic Content Linking Device (ACLD) [7], [8], a just-in-time document recommendation system for meetings, for which the methods proposed in this paper are intended. In II-B, we discuss previous keyword extraction techniques from a transcript or text.

A. Query Formulation in Just-in-Time Retrieval Systems

One of the first systems for document recommendation, referred to as query-free search, was the Fixit system [9], an assistant to an expert diagnostic system for the products of a specific company (fax machines and copiers). Fixit monitored the state of the user's interaction with the diagnostic system, in terms of the positions in a belief network built from the relations among symptoms and faults, and ran background searches on a database of maintenance manuals to provide additional support information related to the current state.

The Remembrance Agent [10], [11], another early just-in-time retrieval system, is closer in concept to the system considered in this paper. The Remembrance Agent was integrated into the Emacs text editor, and ran searches at regular time intervals (every few seconds) using a query that was based on the latest words typed by the user, for instance using a buffer of 20–500 words ranked by frequency. The Remembrance Agent was extended to a multimodal context under the name of Jimminy, a wearable assistant that helped users with taking notes and accessing information when they could not use a standard computer keyboard, e.g. while discussing with another person [12]. Using TFIDF for keyword extraction, Jimminy augmented these keywords with features from other modalities, for example the user's position and the name of their interlocutor(s).

The Watson just-in-time-retrieval system [13] assisted users with finding relevant documents while writing or browsing the Web. Watson built a single query based on a more sophisticated mechanism than the Remembrance Agent, by taking advantage of knowledge about the structure of the written text, e.g. by emphasizing the words mentioned in the abstract or written with larger fonts, in addition to word frequency. The Implicit Queries (IQ) system [14], [15] generated context-sensitive searches by analyzing the text that a user is reading or composing. IQ automatically identified important words to use in a query using TFIDF weights. Another query-free system was designed for enriching television news with articles from the Web [16]. Similarly to IQ or Watson, queries were constructed from the ASR using several variants of TFIDF weighting, and considering also the previous queries made by the system.

Other real-time assistants are conversational: they interact with users to answer their explicit information needs or to provide recommendations based on their conversation. For instance, Ada and Grace[1] are twin virtual museum guides [17], which interact with visitors to answer their questions, suggest exhibits, or explain the technology that makes them work. A collaborative tourist information retrieval system [18], [19] interacts with tourists to provide travel information such as weather conditions, attractive sites, holidays, and transportation, in order to improve their travel plans. MindMeld[2] is a commercial voice assistant for mobile devices such as tablets, which listens to conversations between people, and shows related information from a number of Web-based information sources, such as local directories. MindMeld improves the retrieval results by adding the users' location information to the keywords of conversation obtained using an ASR system. As far as is known, the system uses state-of-the-art methods for language analysis and information retrieval [20].

In collaboration with other researchers, we have designed the Automatic Content Linking Device (ACLD) [7], [8] which is a just-in-time retrieval system for conversational environments, especially intended to be used jointly by a small group of people in a meeting. The system constantly listens to the meeting and prepares implicit queries from words recognized through ASR. A selection of the retrieved documents is recommended to users. Before the solutions proposed in this paper, the ACLD modeled users' information needs as a set of keywords extracted at regular time intervals, by matching the ASR against a list of keywords fixed before the meeting. We showed that this method

[1]See http://ict.usc.edu/prototypes/museum-guides/.

[2]See http://www.expectlabs.com/mindmeld/.

outperforms the use of the entire set of words from a conversation fragment as an implicit query [21]. Moreover, experiments with the use of semantic similarity between a conversation fragment and documents as a criterion for recommendation have shown that, although this improves relevance, its high computation cost makes it unpractical for just-in-time retrieval from a large repository [22, 4.12].

These findings motivated us to design an innovative keyword extraction method for modeling users' information needs from conversations. As mentioned in the introduction, since even short conversation fragments include words potentially pertaining to several topics, and the ASR transcript adds additional ambiguities, a poor keyword selection method leads to non-informative queries, which often fail to capture users' information needs, thus leading to low recommendation relevance and user satisfaction. The keyword extraction method proposed here accounts for a diversity of hypothesized topics in a discussion, and is accompanied by a clustering technique that formulates several topically-separated queries.

B. Keyword Extraction Methods

Numerous methods have been proposed to automatically extract keywords from a text, and are applicable also to transcribed conversations. The earliest techniques have used word frequencies [2] and TFIDF values [3], [23] to rank words for extraction. Alternatively, words have been ranked by counting pairwise word co-occurrence frequencies [24]. These approaches do not consider word meaning, so they may ignore low-frequency words which together indicate a highly-salient topic. For instance, the words 'car', 'wheel', 'seat', and 'passenger' occurring together indicate that automobiles are a salient topic even if each word is not itself frequent [25].

To improve over frequency-based methods, several ways to use lexical semantic information have been proposed. Semantic relations between words can be obtained from a manually-constructed thesaurus such as WordNet, or from Wikipedia, or from an automatically-built thesaurus using latent topic modeling techniques such as LSA, PLSA, or LDA. For instance, keyword extraction has used the frequency of all words belonging to the same WordNet concept set [4], while the Wikifier system [5] relied on Wikipedia links to compute another substitute to word frequency. Hazen also applied topic modeling techniques to audio files [26]. In another study, he used PLSA to build a thesaurus, which was then used to rank the words of a conversation transcript with respect to each topic using a weighted point-wise mutual information scoring function [27]. Moreover, Harwath and Hazen utilized PLSA to represent the topics of a transcribed conversation, and then ranked words in the transcript based on topical similarity to the topics found in the conversation [6]. Similarly, Harwath *et al.* extracted the keywords or key phrases of an audio file by directly applying PLSA on the links among audio frames obtained using segmental dynamic time warping, and then using mutual information measure for ranking the key concepts in the form of audio file snippets [28]. A semi-supervised latent concept classification algorithm was presented by Celikyilmaz and Hakkani-Tur using LDA topic modeling for multi-document information extraction [29].

To consider dependencies among selected words, word co-occurrence has been combined with PageRank [30], and additionally with WordNet [31], or with topical information [32]. For instance, Riedhammer *et al.* considered the dependencies among surrounding words by merging n-gram information obtained from WordNet with word frequency, in order to extract keywords from a meeting transcript [33]. To reduce the effect of noise in the meeting environments, this method removed all n-grams which appear only once or are represented by longer n-grams with the same frequencies. However, as shown empirically in [30], [32] such approaches have difficulties modeling long-range dependencies between words related to the same topic. In another study, part-of-speech information and word clustering techniques were used for keyword extraction [34], while later this information was added to TFIDF so as to consider both word dependency and semantic information [35]. In a recent paper, a word clustering technique was introduced by [36] based on the word2vec vector space representation of a word in which the dependencies between each word and its surrounding words are modeled using a neural network language model [37], [38]. However, although they considered topical similarity and dependency among words, the above methods did not explicitly reward diversity and therefore might miss secondary topics in a conversation fragment.

Supervised machine learning methods have been used to learn models for extracting keywords. This approach was first introduced by Turney [39], who combined heuristic rules with a genetic algorithm. Other learning algorithms such as Naive Bayes [40], Bagging [41], or Conditional Random Fields [42] have been used to improve accuracy. These approaches, however, rely on the availability of in-domain training data, and the objective functions they use for learning do not consider the diversity of keywords.

III. Formulation of Implicit Queries from Conversations

We propose a two-stage approach to the formulation of implicit queries. The first stage is the extraction of keywords from the transcript of a conversation fragment for which documents must be recommended, as provided by an ASR system (Subsection III-A). These keywords should cover as much as possible the topics detected in the conversation, and if possible avoid words that are obviously ASR mistakes. The second stage is the clustering of the keyword set in the form of several topically-disjoint queries (Subsection III-B).

A. Diverse Keyword Extraction

We propose to take advantage of topic modeling techniques to build a topical representation of a conversation fragment, and then select content words as keywords by using topical similarity, while also rewarding the coverage of a diverse range of topics, inspired by recent summarization methods [43], [44]. The benefit of *diverse keyword extraction* is that the coverage of the main topics of the conversation fragment is maximized. Moreover, in order to cover more topics, the proposed algorithm will select a smaller number of keywords from each topic. This is desirable for two reasons. First, as we will see in Section III-B on keyword clustering, this will

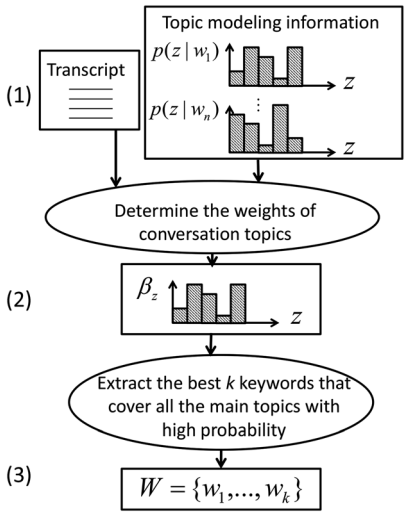

Fig. 1. The three steps of the proposed keyword extraction method: (1) topic modeling, (2) representation of the main topics of the transcript, and (3) diverse keyword selection.

lead to more dissimilar implicit queries, thus increasing the variety of retrieved documents. Second, if words which are in reality ASR noise can create a main topic in the fragment, then the algorithm will choose a smaller number of these noisy keywords compared to algorithms which ignore diversity.

The proposed method for diverse keyword extraction proceeds in three steps, represented schematically in Fig. 1 (a first version of this method appeared in [45]). First, a topic model is used to represent the distribution of the abstract topic z for each word w noted $p(z|w)$ as depicted in Fig. 1. The abstract topics are not pre-defined manually but are represented by latent variables using a generative topic modeling technique. These topics occur in a collection of documents–preferably, one that is representative of the domain of the conversations. Second, these topic models are used to determine weights for the abstract topics in each conversation fragment represented by β_z. Finally, the keyword list $W = \{w_1, \ldots, w_k\}$ which covers a maximum number of the most important topics are selected by rewarding diversity, using an original algorithm introduced in this section.

Modeling Topics in Conversations: Topic models such as Probabilistic Latent Semantic Analysis (PLSA) or Latent Dirichlet Allocation (LDA) [46] can be used as off-line topic modeling techniques to determine the distribution over the topic z of each word w, noted $p(z|w)$, from a large amount of training documents. LDA implemented in the Mallet toolkit [47] is used in this paper because it does not suffer from the overfitting issue of PLSA, as discussed in [46].

When a conversation fragment is considered for keyword extraction, its topics are weighted, each by β_z which is obtained by averaging over all probabilities $p(z|w_i)$ of the N words w_i spoken in the fragment.

$$\beta_z = \frac{1}{N} \sum_{1 \le i \le N} p(z|w_i) \tag{1}$$

Diverse Keyword Extraction Problem: The goal of the keyword extraction technique with maximal topic coverage is formulated as follows. If a conversation fragment t mentions a set of topics Z, and each word w from the fragment t can evoke a subset of the topics in Z, then the goal is to find a subset of k unique words $S \subseteq t$, with $|S| = k$, which maximizes the number of covered topics.

This problem is an instance of the maximum coverage problem, which is known to be *NP*-hard. If the coverage function is submodular and monotone nondecreasing[3], a greedy algorithm can find an approximate solution guaranteed to be within $(1 - \frac{1}{e}) = 0.63$ of the optimal solution in polynomial time [48].

To achieve our goal, we define the contribution of a topic z with respect to each set of words $S \subseteq t$ of size k by summing over all probabilities $p(z|w)$ of the words in the set. Afterward, we propose a reward function, for each set S and topic z, to model the contribution of the set S to the topic z. Finally, we select one of the sets $S \subseteq t$ which maximizes the cumulative reward values over all the topics. The whole procedure is formalized below.

Definition of a Diverse Reward Function: We introduce $r_{S,z}$, the contribution towards topic z of the keyword set S selected from the fragment t:

$$r_{S,z} = \sum_{w \in S} p(z|w) \tag{2}$$

We propose the following reward function for each topic, where β_z represents the weight of topic z over all the words of the fragment to assign a higher weight to topics with higher value and λ is a parameter between 0 and 1. This is a submodular function with diminishing returns when $r_{S,z}$ increases, as proved in Appendix B.

$$f : r_{S,z} \rightarrow \beta_z \cdot r_{S,z}^{\lambda} \tag{3}$$

Finally, the keyword set $S \subseteq t$, is chosen by maximizing the cumulative reward function over all the topics, formulated as follows:

$$R(S) = \sum_{z \in Z} \beta_z \cdot r_{S,z}^{\lambda} \tag{4}$$

In this equation, if candidate keywords which are in fact ASR errors (insertions or substitutions) are associated with topics with lower β_z, as is most often the case, the probability of their selection by the algorithm will reduced, because their contribution to the reward will be small.

Since the class of submodular functions is closed under nonnegative linear combinations [48], $R(S)$ is a monotone non-decreasing submodular function. If $\lambda = 1$, the reward function is linear and only measures the topical similarity of words with the main topics of t. However, when $0 < \lambda < 1$, as soon as a word is selected from a topic, other words from the same topic start having diminishing gains as candidates for selection. Therefore, decreasing the value of λ increases the diversity constraint, which increases the chance of selecting keywords from secondary topics. As these words may reduce the overall relevance of the keyword set, it is essential to find a value of the hyper-parameter λ which leads to the desired balance between relevance and diversity in the keyword set.

[3]A function F is *submodular* if $\forall A \subseteq B \subseteq T \backslash t$, $F(A+t) - F(A) \geq F(B+t) - F(B)$ (diminishing returns) and is *monotone nondecreasing* if $\forall A \subseteq B$, $F(A) \leq F(B)$.

TABLE I
SAMPLE INPUT TO THE GREEDY ALGORITHM

Words	$\mathbf{p(z_1}\vert\cdot)$	$\mathbf{p(z_2}\vert\cdot)$	$\mathbf{p(z_3}\vert\cdot)$	$\mathbf{p(z_4}\vert\cdot)$
w_1	1.00	0.00	0.00	0.00
w_2	0.90	0.00	0.10	0.00
w_3	0.00	0.00	0.20	0.80
w_4	0.10	0.90	0.00	0.00
w_5	0.10	0.10	0.00	0.80

According to a different perspective, the definition of $R(S)$ in Equation 4 can be seen as the dot product in the topic space between the weights β_z obtained from the topic probabilities given the fragment t and the reward function over the sum of topic probabilities $r^{\lambda}_{S,z}$ with a scaling exponent λ and identical coefficients over all topics. However, despite what this apparent similarity suggests, the use of cosine similarity for $R(S)$ would not lead to an appropriate definition because it would not provide a monotone non-decreasing submodular function. Indeed, if vector length normalization is introduced in $R(S)$, for cosine similarity, then we can show that $R(S)$ is no longer monotone submodular, e.g. on the second example in the following subsection.

Examples: We will illustrate the motivation for our definition of $R(S)$ on the following example. Let us consider a situation with four words w_1, w_2, w_3, w_4. The goal is to select two of them as keywords which cover the main topics presented by these four words. Suppose that each word can be related to two topics z_1 and z_2. The probability of topic z_1 for words w_1 and w_2 is 1, and for words w_3 and w_4 it is zero, and vice versa for topic z_2. Therefore, $\beta_{z_1} = \beta_{z_2} = 0.5$. For two sample sets $S_1 = \{w_1, w_2\}$ and $S_2 = \{w_1, w_3\}$ the cumulative rewards are respectively $R(S_1) = 0.5 \cdot (1+1)^{\lambda} + 0.5 \cdot 0^{\lambda}$ and $R(S_2) = 0.5 \cdot 1^{\lambda} + 0.5 \cdot 1^{\lambda}$. Since $R(S_1) \leq R(S_2)$ for $0 < \lambda < 1$, the keyword set S_2 which covers two main topics is selected. If $\lambda = 1$ then the cumulative reward for the two sets S_1 and S_2 is equal, which does not guarantee to select the set which covers both topics.

The example above has the desirable values of $R(S)$ regardless of whether the dot product or the cosine similarity are used for the definition of $R(S)$ in Equation (4). However, this is not always the case. In the example shown in Table I (to which we will refer again below), if we consider $A = \{w_5\}$, $B = \{w_3, w_5\}$ and $\lambda = 0.75$, then $A \subseteq B$ but $R(A) = 0.76 > R(B) = 0.70$ if cosine similarity is used, hence this version of $R(S)$ would not be monotone non-decreasing. If we add keyword w_4 to both keyword sets A and B, then $R(A \cup \{w_4\}) - R(A) = 0.02 < R(B \cup \{w_4\}) - R(B) = 0.09$, hence $R(S)$ would neither have the diminishing returns property, if cosine similarity was used.

Comparison with the Function Used for Summarization: We are inspired by recent work on extractive summarization methods [43][44], to define a monotone submodular function for keyword extraction which maximizes the number of covered topics. This work proposed a square root function as a reward function for the selection of sentences, to cover the maximum number of concepts of a given document. Note that in their work, this function rewards diversity by increasing the gain of selecting a sentence including a concept that was not yet covered by a previously selected sentence. However, we propose a reward function for diverse selection of keywords as a power function with a scaling exponent between 0 and 1, and a coefficient corresponding to the weight of each topic conveyed in the fragment. Therefore, we generalize the square root function and the constant coefficient equal to one for all concepts which is proposed by [43] and [44].

In our reward function, the scaling exponent between 0 and 1 applies diversity by decreasing the reward of keyword selection from a topic when the number of keywords representing that topic increases, and increasing the reward of selecting keywords from the topics which are not covered yet. In contrast to the summarization techniques proposed by [43][44] which add a separate term for considering the relevance and the coverage of the main concepts of the given text by summary sentences, we used a coefficient corresponding to the weight of topics conveyed in the fragment.

Finding the Optimal Keyword Set: To maximize $R(S)$ in polynomial time under the cardinality constraint of $|S| = k$ we present a greedy algorithm shown as Algorithm 1. In the first step of the algorithm, S is empty. At each step, the algorithm selects one of the unselected words from the conversation fragment $w \in t \backslash S$ which has the maximum similarity to the main topics of the conversation fragment and also maximizes the coverage of the topics with respect to the previously selected keywords in S. This is as $h(w, S) = \sum_{z \in Z} \beta_z [p(z|w) + r_{S,z}]^{\lambda}$, where $p(z|w)$ is the contribution to topic z by word $w \in t \backslash S$ which is added to the contribution of the topic z in the set S. The algorithm updates the set S by adding one of the words $w \in t \backslash S$ to the set S which maximizes $h(w, S)$. This procedure continues until reaching k keywords from the fragment t.

Algorithm 1: Diverse keyword extraction.

Input: a given text t, a set of topics Z, the number of keywords k
Output: a set of keywords S
$S \leftarrow \emptyset$
While $|S| \leq k$ **do**
  $S \leftarrow S \cup \{argmax_{w \in t \backslash S}(h(w, S))$ where $h(w, S) = \sum_{z \in Z} \beta_z [p(z|w) + r_{S,z}]^{\lambda}$;
end
return S

Illustration of the Greedy Algorithm: We will exemplify the mechanism of the proposed algorithm using a simple example. Let us consider a conversation fragment with five words, each represented by four topics. The distributions of topics for each word are given in Table I. The topics are thus weighted as follows: $\beta_{z_1} = 0.42$, $\beta_{z_2} = 0.20$, $\beta_{z_3} = 0.06$, and $\beta_{z_4} = 0.32$. We run the algorithm to extract two keywords out of five for $\lambda \in \{.75, 1\}$. In other words, $\lambda = 1$ selects words based on their topical similarity to the main topics of the conversation, and $\lambda = .75$ considers both topical diversity and similarity for keyword extraction.

TABLE II
THE $h(w, S)$ VALUES CALCULATED USING ALGORITHM 1 TO SELECT TWO KEYWORDS OUT OF 5 WORDS FOR $\lambda = .75$ AND 1

	$\lambda = \mathbf{1}$		$\lambda = \mathbf{.75}$	
Words	$\mathbf{h}(\cdot, \emptyset)$	$\mathbf{h}(\cdot, \{\mathbf{w_1}\})$	$\mathbf{h}(\cdot, \emptyset)$	$\mathbf{h}(\cdot, \{\mathbf{w_1}\})$
w_1	**0.420**	–	**0.420**	–
w_2	0.384	**0.804**	0.398	0.690
w_3	0.268	0.688	0.288	0.708
w_4	0.222	0.642	0.259	0.635
w_5	0.318	0.738	0.380	**0.757**

Initially S is empty. The reward values, $h(w, S = \emptyset)$, for all words and $\lambda \in \{.75, 1\}$ are shown in Table II. In the first step of the algorithm, w_1 (the best representative of topic z_1) is added to the set S for both values of λ. In the second step, the $h(w, \{w_1\})$ values are computed for the remaining unselected words, and both λ values, as shown in Table II. According to these values, $\lambda = 1$ selects w_2 as the second word from the topic z_1. However, $\lambda = .75$ selects w_5 as the second keyword (the best representative of topic z_4), the second main topic of the conversation fragment, because it rewards topical diversity in the keyword set.

B. Keyword Clustering

The diverse set of extracted keywords is considered to represent the possible information needs of the participants to a conversation, in terms of the notions and topics that are mentioned in the conversation. To maintain the diversity of topics embodied in the keyword set, and to reduce the noisy effect of each information need on the others, this set must be split into several topically-disjoint subsets. Each subset corresponds then to an implicit query that will be sent to a document retrieval system. These subsets are obtained by clustering topically-similar keywords, as follows.

Clusters of keywords are built by ranking keywords for each main topic of the fragment. The keywords are ordered for each topic by decreasing values of $\beta_z \cdot p(z|w)$. Moreover, in each cluster, only the keywords with a $\beta_z \cdot p(z|w)$ value higher than a threshold (0.01 in the current setting) are kept for each topic z. Note that a given keyword can appear in more than one cluster. Following this ordering criterion, keywords with high value of $p(z|w)$ (i.e. more representative of the topic) will be ranked higher in the cluster of topic z and these keywords will be selected from the topics with high value of β_z. Afterward, clusters themselves are ranked based on their β_z values.

C. From Keywords to Document Recommendations

As a first idea, one implicit query can be prepared for each conversation fragment by using as a query all keywords selected by the diverse keyword extraction technique. However, to improve the retrieval results, multiple implicit queries can be formulated for each conversation fragment, with the keywords of each cluster from the previous section, ordered as above (because the search engine used in our system is not sensitive to word order in queries).

In experiments with only one implicit query per conversation fragment, the document results corresponding to each conversation fragment were prepared by selecting the first d document retrieval results of the implicit query.

In the experiments below with multiple implicit queries, the recommendation lists were prepared by selecting the first document retrieval results of each implicit query and then ranking documents based on the topical similarity of their corresponding queries to the conversation fragment. This is a baseline algorithm, but we describe an improved ranking method in [1].

IV. DATA AND EVALUATION METHODS

Our proposals were tested on three conversational corpora, the Fisher Corpus [49], the AMI Meeting Corpus [50], and the ELEA Corpus [51]. The *relevance of the keywords* was assessed by designing a comparison task and averaging several judgments obtained by crowdsourcing this task through the Amazon Mechanical Turk (AMT) platform. In addition, the α-NDCG measure [52] was used to measure topic diversity in the list of keywords. Afterward, the *quality of implicit queries* was assessed by estimating (again with human judges recruited via AMT) the relevance of the documents that were retrieved when submitting these queries to the Lucene search engine over the English Wikipedia and merging the results as explained above. Here, the conversational data came only from the ELEA Corpus, which offers clearer criteria for assessing the relevance of recommendations than the Fisher and AMI Corpora. We now describe the three corpora and the data extracted from them, as well as the evaluation methods for each task.

A. Conversational Corpora Used for Experiments

The Fisher Corpus [49] contains about 11,000 topic-labeled telephone conversations, on 40 pre-selected topics (one per conversation). In our experiments, we used the manual reference transcripts available with the corpus. We created a topic model using the Mallet [47] implementation of LDA, over two thirds of the Fisher Corpus, given the sufficient number of single-topic documents, fixing the number of abstract topics at 40. The remaining data was used to build 11 artificial conversation fragments (1–2 minutes long) for testing, by concatenating 11 times three fragments about three different topics.

The AMI Meeting Corpus [50] contains conversations on designing remote controls, in series of four scenario-based meetings each, for a total of 171 meetings. Speakers were not constrained to talk about a single topic throughout a meeting, hence these transcripts are multi-topic[4]. Since the number of meetings in the AMI Corpus is not large enough for building topic models with LDA, we used a subset of the English Wikipedia with 124,684 articles. Following several previous studies [53], [54], we fixed the number of topics at 100.

We selected for testing 8 conversation fragments, each 2–3 minutes long, from the AMI Corpus. We used both manual and ASR transcripts of these fragments. The ASR transcripts were generated by the AMI real-time ASR system for meetings [55], with an average word error rate (WER) of 36%.

In addition, for experimenting with a variable range of WER values, we applied to the AMI manual transcripts simulated ASR noise (deletion, insertion and substitution). Namely, we randomly deleted words, or added new words, or substituted

[4]The annotation of "episodes" that is provided with the AMI Corpus considers goal-based rather than topic-based episodes which are not usable here.

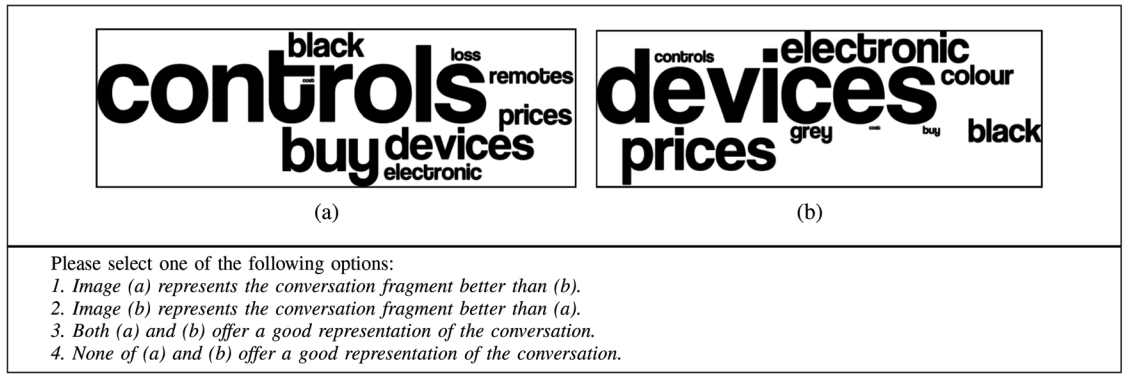

Fig. 2. Example of an evaluation task based on an AMI discussion about the impact of the features of a remote control on its sales. The word clouds were generated using Wordle from the lists produced in this example by: (a) the diverse keyword technique with $\lambda = 0.75$, and (b) a topical similarity method. The latter over-represents the topic 'color' by selecting three words related to it, but misses other topics such as 'remote control', 'losing a device' and 'buying a device' which are also representative of the fragment.

words by other words, in a systematic manner, i.e. all occurrences of a given word type were altered. We randomly selected the word types to be deleted or substituted, as well as the words to be inserted (from the vocabulary of the English Wikipedia), using a variable percentage of noise from 5% to 50%. This simulation technique is actually more challenging to our application than actual ASR errors, which are not created randomly, and tend to spare long content words because they have fewer homophones.

The ELEA Corpus (Emergent LEader Analysis) [51] consists of approximately ten hours of recorded and transcribed meetings, in English and French. Each meeting is a role play game in which participants are supposed to be survivors of an airplane crash, and must rank a list of 12 items with respect to their utility for surviving in the mountain until they are rescued. In our experiments, we considered 5 conversations in English, of about 15 minutes each, and divided their transcripts into 35 segments, of about two minutes each. Similarly to the AMI Corpus, the transcripts in the ELEA Corpus were not sufficient for topic modeling. Therefore, the same subset of the English Wikipedia as for the AMI Corpus was utilized to learn topic models.

B. Evaluation Protocol and Metrics

We designed comparison tasks to evaluate the relevance of extracted keywords and of recommended documents with respect to each conversation fragment. For the former evaluation, we compared the relevance (or representativeness) of two lists of keywords extracted from the same conversation fragment by two different extraction methods. We displayed the transcript of the fragment to a human subject in a web browser, followed below it by several control questions about its content, and then by two lists of keywords (typically, nine keywords in our experiments). To improve readability, the keyword lists were presented using a word cloud representation generated by the Wordle tool[5], in which the words ranked higher are emphasized in the cloud. The subjects had to read the conversation transcript, answer the control questions, and then decide which keyword cloud better represented the content of the conversation fragment. The task is exemplified in Fig. 2, without the control questions or the conversation transcript. A similar method was applied to compare recommended documents, except that two lists of retrieved documents (typically, with seven items each) are shown instead of word clouds, and their potential utility as recommendations to the conversation participants is compared.

[5]http://www.wordle.net

TABLE III
NUMBER OF ANSWERS FOR EACH OF THE FOUR OPTIONS OF THE EVALUATION TASK, FROM TEN JUDGES. THE 8 HITS (A THROUGH H) COMPARE OUR DIVERSE KEYWORD EXTRACTION METHOD (D(.75)) AND THE TOPICAL SIMILARITY (TS) ONE

	A	*B*	*C*	*D*	*E*	*F*	*G*	*H*
Keywords obtained by the topical similarity method are more relevant	4	1	1	1	2	2	1	1
Keywords extracted by the diverse technique (D(.75)) are more relevant	4	1	8	9	6	6	6	8
Both keyword lists are relevant	2	5	1	0	2	2	3	1
Both keyword lists are irrelevant	0	3	0	0	0	0	0	0

In both cases, comparative judgments were integrated over a large number of subjects and conversation fragments. To this end, the tasks were crowdsourced via Amazon's Mechanical Turk as "human intelligence tasks" or HITs. For each comparison task we recruited ten workers completing several HITs in a row, though with a limit to avoid fatigue. The average time spent per HIT was about 2.5 minutes. For qualification control, we only accepted workers with greater than 95% approval rate (i.e. 95% of the worker's submitted HITs for previous tasks had been approved by their requesters) and with more than 1000 approved HITs. We only kept answers from the workers who answered correctly our control questions about each HIT.

Results obtained on the eight manual transcripts of the conversation fragments of the AMI Corpus (noted A–H) are shown in Table III, for the comparison of our proposal for keyword extraction (noted D(.75) as explained in 5.1) with a method using only topic similarity (noted TS). The ten subjects could choose between the four comparative answers presented in Fig. 2, which amount to: 'X better than Y', 'Y better than X', 'both good', or 'both bad'. No answers were rejected for these HITs. The total counts for each answer and each HIT from Table III indicate that subjects agreed strongly on certain answers to some HITs (e.g. C, D, H) but disagreed on others (mainly A). The challenge is therefore to compute a reliable comparative score for two methods from such counts.

We compute comparative relevance scores as follows, based on a method that we proposed in a previous paper [21] for comparing the search results of two document recommender systems for meetings. First, we apply a qualification control factor to the human judgments, reducing the impact of workers who disagree with the majority, using the Pearson correlation of one worker's judgment with the average of all others, noted R_v. We then factor out the impact of undecided HITs by using a measure of the entropy of the answer distribution for each HIT T_j. Finally, we average the results of all judgments and HITs to obtain a comparative score noted CS_a for each set of answers a as shown in Eq. (5), where H is the number of HITs and $X_v(h,a)$ is the judgment of worker v for HIT h and answer list a. When comparing two sets of answers, the sum of the two scores is 100%.

$$\mathrm{CS}_a = \left(\sum_{h=1}^{H} (1 - T_h) \frac{\sum_{v=1}^{V} R_v X_v(h,a)}{\sum_{v=1}^{V} R_v} \right) / \sum_{h=1}^{H} (1 - T_h) \quad (5)$$

To evaluate the diversity of keyword sets, we use the α-NDCG measure proposed for information retrieval [52], which rewards a mixture of relevance and diversity, with equal weights when $\alpha = .5$ as set here. We only apply α-NDCG to the three-topic conversation fragments from the Fisher Corpus, because this is the only dataset with explicitly marked topics. The values are computed according to Eq. (6), where DCG is obtained by Eq. (7) and $\mathrm{DCG}_{\mathrm{ideal}}$ is computed by reordering words in a way that maximizes DCG values in each rank. In Eq. (7), T is the number of mono-topic dialogues included in the conversation, $r_{j,i-1}$ is the number of relevant words to the mono-topic dialogue j until rank i, and $J(w_i, j)$ measures the relevance of the word w in rank i to the mono-topic dialogue j according to Eq. (8). We set the relevance of a word to a conversation topic at 1 when the keyword belongs to the corresponding fragment (one keyword may be relevant to several topics) as shown in Eq. (8). A higher α-NDCG value indicates that keywords from the set are more uniformly distributed across the three topics.

$$\alpha - NDCG[k] = \mathrm{DCG}[k] / \mathrm{DCG}_{\mathrm{ideal}} \quad (6)$$

$$\mathrm{DCG}[k] = \sum_{i=1}^{k} \frac{\sum_{j=1}^{T} J(w_i, j)(1-\alpha)^{r_{j,i-1}}}{\log_2(1+i)} \quad (7)$$

$$J(w_i, j) = \begin{cases} 1 & \text{if} w_i \in \text{dialogue}_j \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

V. Experimental Results

In this section, the diverse keyword extraction technique is compared with two state-of-the-art methods, showing that our proposal extracts more relevant keywords, which cover more topics, and are less likely to be ASR errors. Then, we compare the retrieval results of the implicit queries generated from these keyword lists, either applied entirely or by decomposing them into topically-separated queries, again showing that the lists generated by our method outperform existing methods.

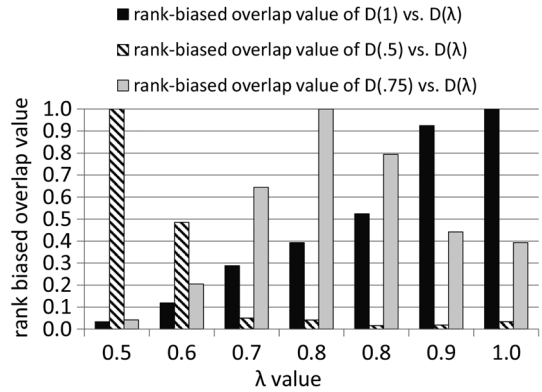

Fig. 3. Comparison of keyword lists generated by D(λ) when $.5 \leq \lambda \leq 1$ using the rank biased overlap metric (RBO) computed between D(λ) and the three keyword lists generated by D(1), D(.5) and D(.75). The differences in RBO are small enough to allow clustering around the three values of λ.

A. Evaluation of Keyword Extraction

Selection of Configurations: We compare several versions of the proposed diverse keyword extraction method (keywords obtained using Algorithm 1) noted D(λ) for $\lambda \in \{.5, .75, 1\}$, a method using only word frequency (noted WF, excluding stopwords), and a recent method based on topical similarity (noted TS) which does not enforce diversity [6]. In fact, D (1) coincides with TS.

The three values of λ are motivated as follows. As the relevance of keywords for D(.5) appeared to be quite low, we did not test lower values of λ. Similarly, we did not test additional values of λ between .5 and 1, apart from .75, because the resulting word lists were very similar to the tested values. This is shown in Fig. 3, where we compare the keyword lists obtained with various values of λ above .5 with keyword lists obtained with the three tested values of λ (.5, .75, and 1), using the rank biased overlap (RBO) [56] as a similarity metric, based on the fraction of keywords overlapping at different ranks.

RBO is computed as follows. Let S and T be two ranked lists, and s_i be the keyword at rank i in S. The set of the keywords up to rank d in S is $\{s_i : i \leq d\}$, noted as $S_{1:d}$. RBO is calculated as in Eq. (9), in which D is the size of the ranked lists.

$$RBO(S,T) = \frac{1}{\sum_{d=1}^{D} \left(\frac{1}{2}\right)^{d-1}} \sum_{d=1}^{D} \left(\frac{1}{2}\right)^{d-1} \frac{|S_{1:d} \cap T_{1:d}|}{|S_{1:d} \cup T_{1:d}|} \quad (9)$$

The variations of RBO across the sampled values of λ, with respect to three main configurations of the system, show that these values can indeed be clustered into three groups: $.5 \leq \lambda < .7$ can be clustered around $\lambda = .5$ as a representative value; $.7 \leq \lambda \leq .8$ can be assigned to the $\lambda = .75$ cluster; and $.8 < \lambda \leq 1$ can be represented by $\lambda = 1$. Therefore, in our human evaluations, we will consider only the D(.5), D(.75) and D(1), and note the latter as TS.

Measuring Topical Diversity: First of all, we compared the four keyword extraction methods (WF, TS, D(.5) and D(.75)) in terms of the diversity of their results over the concatenated fragments of the Fisher Corpus, by using α-NDCG (Eq. (6)) to measure how evenly the extracted keywords were distributed across the three topics of each fragment. Fig. 4 shows results averaged over the 11 three-topic conversation fragments of the

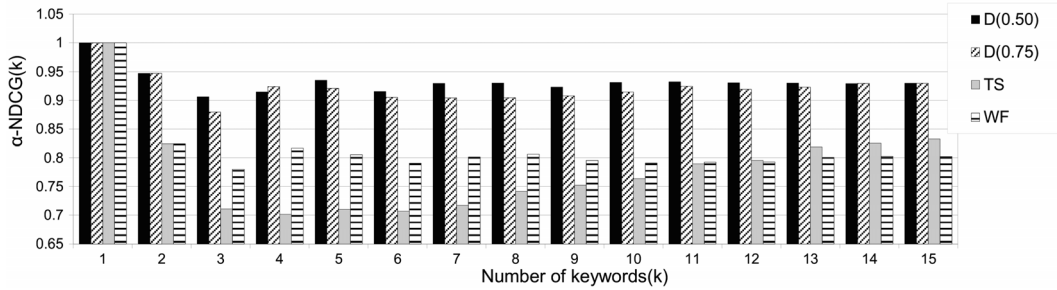

Fig. 4. Average α-NDCG over the 11 three-topic conversation fragments of the Fisher Corpus, for a number of keywords varying from 1 to 15. The most difficult task is to extract exactly one keyword from each topic, hence the lowest scores are for three keywords. The best performing methods are always the diversity-preserving ones, D(.5) and D(.75).

TABLE IV
COMPARATIVE RELEVANCE SCORES BASED ON HUMAN JUDGMENTS FOR FOUR KEYWORD EXTRACTION METHODS. THE FOLLOWING RANKING CAN BE INFERRED: D(.75) > TS > WF > D(.5)

Corpus	Compared methods ($\mathbf{m_1}$ vs. $\mathbf{m_2}$)	Relevance (%)	
		$\mathbf{m_1}$	$\mathbf{m_2}$
Fisher manual transcripts	D(.75) vs. TS	**68**	32
	TS vs. WF	**82**	18
	WF vs. D(.5)	**95**	5
AMI manual transcripts	D(.75) vs. TS	**78**	22
	TS vs. WF	**60**	40
	WF vs. D(.5)	**78**	22
AMI ASR transcripts	D(.75) vs. TS	**79**	21
	TS vs. WF	**65**	35
	WF vs. D(.5)	**73**	27

Fisher Corpus, for various sizes of the keyword set, between 1 and 15. The average α-NDCG values for D(.75) and D(.5) are similar, and they are clearly higher than those for WF and TS for all ranks (except, of course, in the case of a single keyword in which they coincide). The values for TS are particularly low, and only increase for a large number of keywords, demonstrating that TS does not cope well with topic diversity, but on the contrary emphasizes keywords from the dominant topic. The values for WF are more uniform as it does not consider topics at all.

Measuring Keyword Relevance: We performed binary comparisons between the outputs of each keyword extraction method, using crowdsourcing, over 11 fragments from the manual transcripts of the Fisher Corpus and 8 fragments from the manual transcripts of the AMI Corpus. The goal is to rank the methods, so we only report here on the binary comparisons that allowed us to determine the ordering of the four methods, and exclude redundant comparisons.

Table III (in Section IV-B above) shows as an illustration the judgments that were collected when comparing the output of D(.75) with TS on the 8 HITs of the AMI Corpus. Workers tended to disagree about the first two HITs, but then clearly found that the keywords extracted by D(.75) for the subsequent six HITs better represented the conversation compared to TS. For these results, our consolidated relevance score (Eq. (5)) is 78% for D(.75) vs. 22% for TS.

The averaged relevance values (consolidated scores) for all comparisons needed to rank the four methods are shown in Table IV, separately for the manual transcripts of both Fisher and AMI corpora. Although the exact differences vary, the human judgments over both corpora indicate the following ranking: **D(.75) > TS > WF > D(.5)**. The optimal value of λ is thus .75, and with this value, our diversity-aware method D(.75) extracts keyword sets that are judged to be more representative than those extracted by TS or WF. The differences between TS and WF, as well as WF and D(.5) are larger for the Fisher Corpus, likely due to the artificial fragments with three topics, but they are still visible on the natural fragments of the AMI Corpus. The low scores of D(.5) are due to the low overall relevance of keywords. In particular, the comparative relevance of D(.75) vs. D(.5) on the Fisher Corpus is very large (96% vs. 4%).

Measuring Noise Reduction Power: We performed binary comparisons between the keyword lists obtained by four keyword extraction methods, using crowdsourcing, over the 8 fragments from the ASR transcripts of the AMI Corpus produced by the real time ASR system. The averaged relevance values for all comparisons needed to rank these methods are shown at the bottom of Table IV. The differences between comparison values are similar for the ASR transcripts compared to the manual ones, although we notice a degradation of WF due to ASR noise. As D(.75) still outperforms TS, the ranking remains unchanged in the presence of the ASR noise: **D(.75) > TS > WF > D(.5)**.

We have counted the number of keywords selected by each method among ASR errors, which were artificially generated as explained in Section IV-A, as in this case these words are precisely known. The average numbers of such erroneous keywords are shown in Fig. 5 for a noise level varying from 5% to 50% on the AMI Corpus. The results show that D(.75) selects a smaller number of noisy keywords compared to TS and WF. The WF method does not consider topic and only selects words with higher frequency, so it may select noisy keywords if they correspond to a systematic mistake of the ASR system. Conversely, if noisy words are located in insignificant topics, the probability of selection by both TS and D(.75) will be reduced because both select the keywords placed in the main topics. Moreover, if a systematic ASR error generates words that produce a main topic, the advantage of the D(.75) over TS is that it selects a smaller number of noisy keywords.

B. Retrieval Relevance Scores

In this section, we compare the retrieval results of queries built from keyword lists by performing again binary comparisons, using crowdsourcing to collect human judgments. Each query is passed to the Lucene search engine over the English

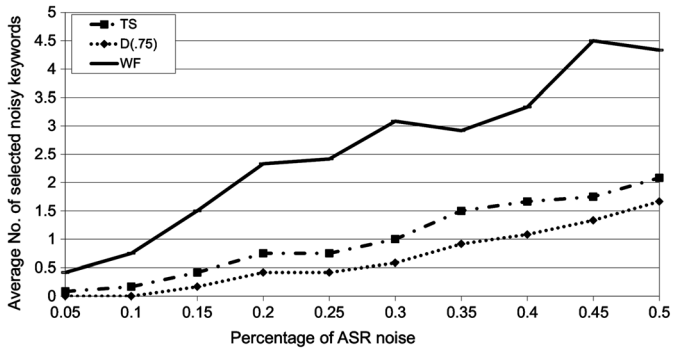

Fig. 5. Average number of noisy keywords chosen by the algorithms over the 8 conversation fragments of the AMI Corpus, for a varying percentage of artificial ASR noise from 5% to 50%. The best performing method is D(.75).

Wikipedia articles. We consider two methods for deriving implicit queries from keyword lists, to which we refer as *single query* and respectively *multiple queries*. A single query is made by simply using the entire keyword set, while multiple queries are constructed by dividing the keyword set into multiple topically-independent sub-sets (see Section III-B) and using each sub-set as one implicit query, merging afterward the results into a unique document set (see Section III-C).

In previous versions of our system [7], we used the entire conversation fragment, minus the stopwords, as a single (long) query. We have also evaluated this approach, comparing its results with those of the keyword list based on word frequency (WF). We found that WF outperformed the use of the entire fragment, with 87% vs. 13% comparative relevance. Therefore, we will not discuss this method any further. Several previous just-in-time retrieval systems such as the Remembrance Agent also used WF rather than all words.

Firstly, we built single queries from the keyword sets provided by the D(.75), TS and WF keyword extraction methods, and compared the three resulting document sets. Secondly, we built multiple queries from the same methods and performed similar comparisons between the resulting document sets. Finally, we compared the best results of multiple queries with the best results of single queries. This procedure was used because evaluation with human subjects is time-consuming, therefore we attempted to carried out the minimal number of comparisons allowing the ordering of the methods.

Comparison Across Single Queries: Binary comparisons were performed between the retrieval results from single queries based on D(.75), TS, and WF over 35 fragments from the ELEA Corpus. As explained above in Section IV-B, the workers compared two document lists in terms of the relevance or utility of suggested documents to the meeting participants at the time of the corresponding fragment, represented through its transcript.

The average relevance values (for the metric described in Section IV-B) for the comparisons needed to rank the three methods are shown in Table V. The human judgments indicate the following ranking: **D(.75)** $>$ **WF** $>$ **TS** which shows the superiority of diversity-aware keyword extraction technique in terms of the relevance of the resulting document sets when these keywords are used as a single query.

TABLE V
COMPARATIVE RELEVANCE SCORES OF DOCUMENT RESULT LISTS USING SINGLE QUERIES OBTAINED FROM THREE KEYWORD EXTRACTION METHODS ON THE ELEA CORPUS. THE FOLLOWING RANKING CAN BE INFERRED: D(.75) $>$ WF $>$ TS

Compared methods ($\mathbf{m_1}$ **vs.** $\mathbf{m_2}$)	**Relevance (%)**	
	$\mathbf{m_1}$	$\mathbf{m_2}$
WF vs. TS	**54**	46
D(.75) vs. WF	**58**	42
D(.75) vs. TS	**70**	30

TABLE VI
COMPARATIVE RELEVANCE SCORES OF DOCUMENT RESULTS USING CD(0.75), D(.75), CTS, AND TS ON THE ELEA CORPUS. METHODS USING MULTIPLE QUERIES OUTPERFORM THOSE USING SINGLE QUERIES, AND AMONG THE FORMER, CD(0.75) SURPASSES CTS

Compared methods ($\mathbf{m_1}$ **vs.** $\mathbf{m_2}$)	**Relevance (%)**	
	$\mathbf{m_1}$	$\mathbf{m_2}$
CD(.75) vs. D(.75)	**65**	35
CTS vs. TS	**65**	35
CD(.75) vs. CTS	**62**	38

Comparison Across Multiple Queries: Binary comparisons were then performed between the retrieval results of multiple topically-disjoint queries. Multiple queries were prepared from the keyword lists obtained from the TS and D(.75) keyword extraction methods. The two tested methods are noted CTS and CD(.75) (with 'C' for clusters of keywords), respectively derived from the TS and D(.75) keyword lists. Clustering the results of WF is unpractical since the method does not rely on topic modeling. Human judgments gathered over the 35 fragments from the ELEA Corpus show that CD(.75) outperforms CTS, with an averaged relevance value of 62% vs. 38% as shown in Table VI.

Single Queries Versus Multiple Queries: Finally, we compared single queries with multiple queries derived from the same keyword lists, namely D(.75) and TS, on the 35 fragments from the ELEA Corpus. The averaged relevance values obtained from human judgments, in Table VI, revealed that using multiple queries, for both types of keyword extraction techniques, leads to more relevant document results compared to the single queries, i.e. **CD(.75)** $>$ **D(.75)** and **CTS** $>$ **TS**. For both comparisons, the averaged relevance scores vary in the same proportion, namely 65% to 35%, which also shows that the improvement brought by multiple queries has a similar order of magnitude for D(.75) and for TS.

Example of Document Results: To illustrate the superiority of CD(.75) over the other techniques, we consider an example from one of the conversation fragments of the ELEA Corpus, given in Appendix A. As described in Section IV, the speakers had to select a list of 12 items vital to survive in cold mountainous conditions while they waited to be rescued. The lists of keywords extracted for this fragment by D(.75), TS, and WF are shown in Table VII. As WF does not consider topical information, the keywords it selects ('lighted', 'light', and 'lighter') all revolve around the same notion.

Table VIII shows the topically-aware implicit queries prepared from the keyword lists provided by the D(.75) and TS methods (Table VII) ordered based on their importance in

TABLE VII
EXAMPLES OF KEYWORD SETS OBTAINED BY THREE KEYWORD EXTRACTION METHODS FOR A FRAGMENT OF THE ELEA CORPUS

WF	TS	D(.75)
$S = \{$whiskey, fire, axe, wool, extra, lighted, light, lighter$\}$	$S = \{$chocolate, cigarette, whiskey, whisk, shortening, shoe, pistol, lighter$\}$	$S = \{$chocolate, cigarette, lighter, whiskey, pistol, wool, shoe, fire$\}$

TABLE VIII
EXAMPLES OF IMPLICIT QUERIES BUILT FROM THE KEYWORD LIST EXTRACTED FROM A FRAGMENT OF THE ELEA CORPUS. EACH IMPLICIT QUERY COVERS ONE OF THE ABSTRACT TOPICS OF THE FRAGMENT

CTS	CD(.75)
$q_1 = \{$chocolate, cigarette, whiskey, whisk, shortening, lighter$\}$	$q_1 = \{$chocolate, cigarette, whiskey, lighter$\}$
	$q_2 = \{$shoe, wool, lighter$\}$
$q_2 = \{$shoe, lighter$\}$	$q_3 = \{$pistol, fire, lighter$\}$
$q_3 = \{$pistol, lighter$\}$	$q_4 = \{$wool$\}$

TABLE IX
EXAMPLES OF RETRIEVED WIKIPEDIA PAGES FROM FIVE DIFFERENT METHODS FOR A FRAGMENT OF THE ELEA CORPUS. RESULTS OF DIVERSE KEYWORD EXTRACTION (D(.75)) COVER MORE TOPICS, AND MULTIPLE IMPLICIT QUERIES REDUCE NOISE (CD(.75))

WF	TS	D(.75)	CTS	CD(.75)
Light	Cigarette	Wool	Cigarette	Cigarette
Lighting	Lighter	Cigarette	Shoe	Wool
Light My Fire	Shortening	Lighter	Lighter	Lighter
Lightness	Shorten	25 m rapid fire pistol	Shortening	Mineral wool
Light On	Whisk	Fire safe cigarettes	Lighter than air	Chocolate
In the Light	Fly-whisk	25 m center-fire pistol	Lighter (barge)	Shoe

the fragment. The CTS method starts by covering the first main topic of this fragment with the keywords 'chocolate', 'cigarette', 'whiskey', 'whisk', and 'shortening'. Then it selects 'shoe' and 'pistol' to cover the second and third main topics respectively. However, the CD(.75) method which considers also topical diversity, first selects two keywords to cover the first main topic. Then it selects the third keyword of the first main topic only after the selection of a keyword shared by the first three main topics. Afterward, it selected the keywords 'pistol', 'wool', 'shoe', and 'fire' to respectively cover the second, third and fourth main topics of the fragment.

Finally, Table IX shows the retrieval results (five highest-ranked Wikipedia pages) obtained by WF, TS, D(.75), CTS, and CD(.75). First of all, WF recommends almost no relevant document to participants. Queries made by the diverse keyword extraction technique (D(.75)) retrieve documents which cover the largest number of topics mentioned in the conversation fragment. Moreover, multiple queries (CTS and CD(.75)) retrieve a large number of relevant documents compared to single queries (TS and D(.75)), likely because single queries do not separate the mixture of topics in the conversation, and lead to irrelevant results such as 'Shorten', 'Whisk', 'Fly-whisk' (found by TS) and '25 metre rapid fire pistol', 'Fire safe cigarettes', '25 metre center-fire pistol' (found by D(.75)). In addition, CD(.75) finds documents which cover more topics mentioned in the conversation fragment in comparison to CTS.

VI. CONCLUSION

We have considered a particular form of just-in-time retrieval systems intended for conversational environments, in which they recommend to users documents that are relevant to their information needs. We focused on modeling the users' information needs by deriving implicit queries from short conversation fragments. These queries are based on sets of keywords extracted from the conversation. We have proposed a novel diverse keyword extraction technique which covers the maximal number of important topics in a fragment. Then, to reduce the noisy effect on queries of the mixture of topics in a keyword set, we proposed a clustering technique to divide the set of keywords into smaller topically-independent subsets constituting implicit queries.

We compared the diverse keyword extraction technique with existing methods, based on word frequency or topical similarity, in terms of the representativeness of the keywords and the relevance of retrieved documents. These were judged by human raters recruited via the Amazon Mechanical Turk crowdsourcing platform. The experiments showed that the diverse keyword extraction method provides on average the most representative keyword sets, with the highest α-NDCG value, and leading–through multiple topically-separated implicit queries–to the most relevant lists of recommended documents. Therefore, enforcing both relevance and diversity brings an effective improvement to keyword extraction and document retrieval. The keyword extraction method could be improved by considering n-grams of words in addition to individual words only, but this requires some adaptation of the entire processing chain.

Our current goals are to process also explicit queries, and to rank document results with the objective of maximizing the coverage of all the information needs, while minimizing redundancy in a short list of documents. Integrating these techniques in a working prototype should help users to find valuable documents immediately and effortlessly, without interrupting the conversation flow, thus ensuring the usability of our system. In the future, this will be tested with human users of the system within real-life meetings.

APPENDIX A
CONVERSATION FRAGMENT FROM THE ELEA CORPUS

The following transcript of a four-party conversation (speakers A–D) was submitted to the document recommendation system. The keyword lists, queries and documents retrieved are respectively shown in Tables VII, VIII, and IX.

A: that should be almost the same okay.

B: of the same proportions.

A: so now we have the the axe, the extra clothing and the whiskey.

C: where is the whiskey

A: whiskey is useless, I dont want if nobody is bothered if we have fire but that is -

B: can whisk can whiskey be fire? can whiskey be lighted -

C: err I thought about it and err -

A: yes we may but yes, it may be possible yes whiskey can be good for the fire; but I dont know if it is highly flammable.

B: well it can be it if it can be fired I mean lighted it can be the fluid of the cigarette lighter.

C: yes but I think it is not strong enough.

A: I think it doesnt

D: almost done?

A: almost done. I am to put first the clothes then the axe, and then the whiskey.

B: yes.

A: because I feel like I didnt have that -

B: but but I mean the last two things it doesnt it doesnt matter too much I think.

A: okay. so I am yes we have like the canvas first, then the chocolate, shortening; the cigarette lighter, the pistol; and the newspaper. and then the clothes, the axe, whiskey, map, compass. wool, steel wool. wool.

B: where did you put the newspaper?

A: to err light the fire.

B: okay.

A: and the to light in your shoes, to get warmer.

B: do we have okay. then we have extra shirts and pants. yes.

D: are you done?

APPENDIX B
PROOF FOR SECTION III

LEMMA: $f(r_{S,z})$ defined in Eq. (3) is a monotone nondecreasing submodular function.

PROOF: The property is illustrated in Fig. 6 for various values of λ, and is proven as follows:

To show that $f(r_{S,z})$ is monotone nondecreasing, let A and B be two arbitrary sets of keywords such that $A \subseteq B$, and let us show that $f(r_{A,z}) \leq f(r_{B,z})$. If $C = B \backslash A$, then:

$$f(r_{B,z}) = f(r_{A \cup C,z}) = \beta_z \cdot r^{\lambda}_{A \cup C,z} = \beta_z \cdot (r_{A,z} + r_{C,z})^{\lambda}$$

If we substitute $\beta_z \cdot (r_{A,z} + r_{C,z})^{\lambda}$ with its binomial expansion by an infinite series (because λ is not an integer), then:

$$f(r_{B,z}) = \beta_z \cdot \left(r^{\lambda}_{A,z} + \sum_{k=1}^{\infty} \binom{\lambda}{k} r^{\lambda-k}_{A,z} r^{k}_{C,z} \right).$$

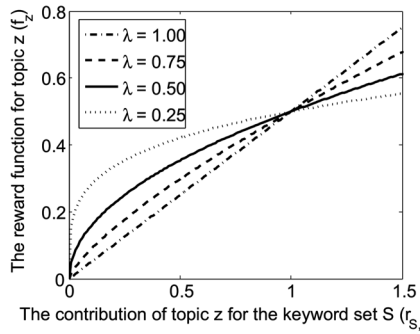

Fig. 6. The reward value given to the topic z is decreasing when the contribution of this topic is increasing for various λ values (1, 0.75, 0.5, and 0.25) and $\beta_z = 0.5$.

Since $r_{A,z}$ and $r_{C,z}$ are obtained by summing over positive probability values, $\sum_{k=1}^{\infty} \binom{\lambda}{k} r^{\lambda-k}_{A,z} r^{k}_{C,z}$ is a positive value. So we conclude that $f(r_{A,z}) \leq f(r_{B,z})$, and the function is monotone.

Second we prove that $f(r_{S,z})$ has the diminishing returns property. Let A and B be two arbitrary sets of keywords such that $A \subseteq B$. Let w be a keyword not in B, and $A' = A \cup \{w\}$ and $B' = B \cup \{w\}$. We will now show that $f(r_{B',z}) - f(r_{B,z}) \leq f(r_{A',z}) - f(r_{A,z})$, which is the diminishing returns property.

$$\begin{aligned} f(r_{A',z}) - f(r_{A,z}) &= \beta_z \cdot r^{\lambda}_{A',z} - \beta_z \cdot r^{\lambda}_{A,z} \\ &= \beta_z \cdot (r_{A,z} + p(z|w))^{\lambda} - \beta_z \cdot r^{\lambda}_{A,z} \end{aligned}$$

If we substitute $\beta_z \cdot (r_{A,z} + p(z|w))^{\lambda}$ with its binomial expansion, as above, then:

$$\begin{aligned} &f(r_{A',z}) - f(r_{A,z}) = \\ &\beta_z \cdot \left(r^{\lambda}_{A,z} + \sum_{k=1}^{\infty} \binom{\lambda}{k} r^{\lambda-k}_{A,z} p(z|w)^k - r^{\lambda}_{A,z} \right) = \\ &\beta_z \cdot \sum_{k=1}^{\infty} \binom{\lambda}{k} r^{\lambda-k}_{A,z} p(z|w)^k. \end{aligned}$$

Similarly, we can establish that

$$f(r_{B',z}) - f(r_{B,z}) = \beta_z \cdot \sum_{k=1}^{\infty} \binom{\lambda}{k} r^{\lambda-k}_{B,z} p(z|w)^k.$$

Since $A \subseteq B$ then $r^{\lambda}_{A,z} \leq r^{\lambda}_{B,z}$. We also know that $(\lambda - k) < 0$ for all positive integers k, because $0 \leq \lambda \leq 1$. So we have $r^{\lambda-k}_{B,z} \leq r^{\lambda-k}_{A,z}$, and consequently $\beta_z \cdot \sum_{k=1}^{\infty} \binom{\lambda}{k} r^{\lambda-k}_{B,z} p(z|w)^k \leq \beta_z \cdot \sum_{k=1}^{\infty} \binom{\lambda}{k} r^{\lambda-k}_{A,z} p(z|w)^k$, which concludes the proof for diminishing returns.

Since it is monotone nondecreasing and has diminishing returns, $f(r_{S,z})$ is a monotone submodular function. □

ACKNOWLEDGMENT

We thank Dairazalia Sanchez-Cortes and the Idiap Social Computing group for access to the ELEA Corpus. We also acknowledge the anonymous reviewers for their precise comments and insightful remarks that improved the quality and clarity of our submission.

REFERENCES

[1] M. Habibi and A. Popescu-Belis, "Enforcing topic diversity in a document recommender for conversations," in *Proc. 25th Int. Conf. Comput. Linguist. (Coling)*, 2014, pp. 588–599.

[2] H. P. Luhn, "A statistical approach to mechanized encoding and searching of literary information," *IBM J. Res. Develop.*, vol. 1, no. 4, pp. 309–317, 1957.

[3] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Inf. Process. Manage. J.*, vol. 24, no. 5, pp. 513–523, 1988.

[4] S. Ye, T.-S. Chua, M.-Y. Kan, and L. Qiu, "Document concept lattice for text understanding and summarization," *Inf. Process. Manage.*, vol. 43, no. 6, pp. 1643–1662, 2007.

[5] A. Csomai and R. Mihalcea, "Linking educational materials to encyclopedic knowledge," in *Proc. Conf. Artif. Intell. Educat.: Building Technol. Rich Learn. Contexts That Work*, 2007, pp. 557–559.

[6] D. Harwath and T. J. Hazen, "Topic identification based extrinsic evaluation of summarization techniques applied to conversational speech," in *Proc. Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2012, pp. 5073–5076.

[7] A. Popescu-Belis, E. Boertjes, J. Kilgour, P. Poller, S. Castronovo, T. Wilson, A. Jaimes, and J. Carletta, "The AMIDA automatic content linking device: Just-in-time document retrieval in meetings," in *Proc. 5th Workshop Mach. Learn. Multimodal Interact. (MLMI)*, 2008, pp. 272–283.

[8] A. Popescu-Belis, M. Yazdani, A. Nanchen, and P. N. Garner, "A speech-based just-in-time retrieval system using semantic search," in *Proc. Annu. Conf. North Amer. Chap. ACL (HLT-NAACL)*, 2011, pp. 80–85.

[9] P. E. Hart and J. Graham, "Query-free information retrieval," *Int. J. Intell. Syst. Technol. Applicat.*, vol. 12, no. 5, pp. 32–37, 1997.

[10] B. Rhodes and T. Starner, "Remembrance Agent: A continuously running automated information retrieval system," in *Proc. 1st Int. Conf. Pract. Applicat. Intell. Agents Multi Agent Technol.*, London, U.K., 1996, pp. 487–495.

[11] B. J. Rhodes and P. Maes, "Just-in-time information retrieval agents," *IBM Syst. J.*, vol. 39, no. 3.4, pp. 685–704, 2000.

[12] B. J. Rhodes, "The wearable Remembrance Agent: A system for augmented memory," *Personal Technol.*, vol. 1, no. 4, pp. 218–224, 1997.

[13] J. Budzik and K. J. Hammond, "User interactions with everyday applications as context for just-in-time information access," in *Proc. 5th Int. Conf. Intell. User Interfaces (IUI'00)*, 2000, pp. 44–51.

[14] M. Czerwinski, S. Dumais, G. Robertson, S. Dziadosz, S. Tiernan, and M. Van Dantzich, "Visualizing implicit queries for information management and retrieval," in *Proc. SIGCHI Conf. Human Factors Comput. Syst. (CHI)*, 1999, pp. 560–567.

[15] S. Dumais, E. Cutrell, R. Sarin, and E. Horvitz, "Implicit queries (IQ) for contextualized search," in *Proc. 27th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2004, pp. 594–594.

[16] M. Henzinger, B.-W. Chang, B. Milch, and S. Brin, "Query-free news search," *World Wide Web: Internet Web Inf. Syst.*, vol. 8, no. 2, pp. 101–126, 2005.

[17] D. Traum, P. Aggarwal, R. Artstein, S. Foutz, J. Gerten, A. Katsamanis, A. Leuski, D. Noren, and W. Swartout, "Ada and Grace: Direct interaction with museum visitors," in *Proc. 12th Int. Conf. Intell. Virtual Agents*, 2012, pp. 245–251.

[18] A. S. M. Arif, J. T. Du, and I. Lee, "Examining collaborative query reformulation: A case of travel information searching," in *Proc. 37th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2014, pp. 875–878.

[19] A. S. M. Arif, J. T. Du, and I. Lee, "Towards a model of collaborative information retrieval in tourism," in *Proc. 4th Inf. Interact. Context Symp.*, 2012, pp. 258–261.

[20] J. Zaino, MindMeld makes context count in search, [Online]. Available: http://semanticweb.com/mindmeld-makes-context-count-search_b42725 2014

[21] M. Habibi and A. Popescu-Belis, "Using crowdsourcing to compare document recommendation strategies for conversations," *Workshop Recommendat. Utility Eval.: Beyond RMSE (RUE'11)*, pp. 15–20, 2012.

[22] M. Yazdani, "Similarity learning over large collaborative networks," Ph.D. dissertation, EPFL Doctoral School in Information and Communication (EDIC), Lausanne, Switzerland, 2013.

[23] G. Salton, C.-S. Yang, and C. T. Yu, "A theory of term importance in automatic text analysis," *J. Amer. Soc. Inf. Sci.*, vol. 26, no. 1, pp. 33–44, 1975.

[24] Y. Matsuo and M. Ishizuka, "Keyword extraction from a single document using word co-occurrence statistical information," *Int. J. Artif. Intell. Tools*, vol. 13, no. 1, pp. 157–169, 2004.

[25] A. Nenkova and K. McKeown, "A survey of text summarization techniques," in *Mining Text Data*, C. C. Aggarwal and C. Zhai, Eds. New York, NY, USA: Springer, 2012, ch. 3, pp. 43–76.

[26] T. J. Hazen, "Topic identification," in *Spoken Language Understanding: Systems for Extracting Semantic Information from Speech*, G. Tur and R. De Mori, Eds. New York, NY, USA: Wiley, 2011, ch. 12, pp. 319–356, 1em plus 0.5em minus 0.4em.

[27] T. J. Hazen, "Latent topic modeling for audio corpus summarization," in *Proc. 12th Annu. Conf. Int. Speech Commun. Assoc.*, 2011, pp. 913–916.

[28] D. F. Harwath, T. J. Hazen, and J. R. Glass, "Zero resource spoken audio corpus analysis," in *Proc. Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2013, pp. 8555–8559.

[29] A. Celikyilmaz and D. Hakkani-Tur, "Concept-based classification for multi-document summarization," in *Proc. Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2011, pp. 5540–5543.

[30] R. Mihalcea and P. Tarau, "TextRank: Bringing order into texts," in *Proc. Conf. Empir. Meth. Nat. Lang. Process. (EMNLP'04)*, 2004, pp. 404–411.

[31] J. Wang, J. Liu, and C. Wang, "Keyword extraction based on pagerank," in *Proc. Adv. Knowl. Disc. Data Mining (PAKDD)*, 2007, pp. 857–864.

[32] Z. Liu, W. Huang, Y. Zheng, and M. Sun, "Automatic keyphrase extraction via topic decomposition," in *Proc. Conf. Empir. Meth. Nat. Lang. Process. (EMNLP'10)*, 2010, pp. 366–376.

[33] K. Riedhammer, B. Favre, and D. Hakkani-Tur, "A keyphrase based approach to interactive meeting summarization," in *Proc. IEEE Spoken Lang. Technol. Workshop (SLT'08)*, 2008, pp. 153–156.

[34] Z. Liu, P. Li, Y. Zheng, and M. Sun, "Clustering to find exemplar terms for keyphrase extraction," in *Proc. Conf. Empir. Meth. Nat. Lang. Process. (EMNLP'09)*, 2009, pp. 257–266.

[35] F. Liu, D. Pennell, F. Liu, and Y. Liu, "Unsupervised approaches for automatic keyword extraction using meeting transcripts," in *Proc. Annu. Conf. North Amer. Chap. ACL (HLT-NAACL)*, 2009, pp. 620–628.

[36] B. Xue, C. Fu, and Z. Shaobin, "A new clustering model based on Word2vec mining on Sina Weibo users' tags," *Int. J. Grid & Distrib. Comput.*, vol. 7, no. 3, pp. 41–48, 2014.

[37] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *Preprint arXiv:1301.3781*, 2013.

[38] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. 26th Annu. Conf. Neural Inf. Process. Syst. (NIPS)*, 2013, pp. 3111–3119.

[39] P. Turney, "Learning to extract keyphrases from text National Research Council Canada (NRC)," Tech. Rep. ERB-1057, 1999.

[40] E. Frank, G. W. Paynter, I. H. Witten, C. Gutwin, and C. G. Nevill-Manning, "Domain-specific keyphrase extraction," in *Proc. 16th Int. Joint Conf. Artif. Intell. (IJCAI'99)*, 1999, pp. 668–673.

[41] A. Hulth, "Improved automatic keyword extraction given more linguistic knowledge," in *Proc. Conf. Empir. Meth. Nat. Lang. Process. (EMNLP'03)*, 2003, pp. 216–223.

[42] C. Zhang, H. Wang, Y. Liu, D. Wu, Y. Liao, and B. Wang, "Automatic keyword extraction from documents using conditional random fields," *J. Comput. Inf. Syst.*, vol. 4, no. 3, pp. 1169–1180, 2008.

[43] H. Lin and J. Bilmes, "A class of submodular functions for document summarization," in *Proc. 49th Annu. Meeting Assoc. Comput. Linguist. (ACL)*, Portland, OR, USA, 2011, pp. 510–520.

[44] J. Li, L. Li, and T. Li, "Multi-document summarization via submodularity," *Appl. Intell.*, vol. 37, no. 3, pp. 420–430, 2012.

[45] M. Habibi and A. Popescu-Belis, "Diverse keyword extraction from conversations," in *Proc. 51st Annu. Meeting Assoc. Comput. Linguist.*, 2013, pp. 651–657.

[46] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, 2003.
[47] A. K. McCallum, *MALLET: A Machine Learning for Language Toolkit*, 2002 [Online]. Available: http://mallet.cs.umass.edu.
[48] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, "An analysis of approximations for maximizing submodular set functions," *Math. Program. J.*, vol. 14, no. 1, pp. 265–294, 1978.
[49] C. Cieri, D. Miller, and K. Walker, "The Fisher Corpus: A resource for the next generations of speech-to-text," in *Proc. 4th Int. Conf. Lang. Resources Eval. (LREC)*, 2004, pp. 69–71.
[50] J. Carletta, "Unleashing the killer corpus: Experiences in creating the multi-everything AMI Meeting Corpus," *Lang. Resources Eval. J.*, vol. 41, no. 2, pp. 181–190, 2007.
[51] D. Sanchez-Cortes, O. Aran, M. Schmid Mast, and D. Gatica-Perez, "A nonverbal behavior approach to identify emergent leaders in small groups," *IEEE Trans. Multimedia*, vol. 14, no. 3, pp. 816–832, Jun. 2012.
[52] C. L. A. Clarke, M. Kolla, G. V. Cormack, O. Vechtomova, A. Ashkan, S. Büttcher, and I. MacKinnon, "Novelty and diversity in information retrieval evaluation," in *Proc. 31st Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2008, pp. 659–666.
[53] J. Boyd-Graber, J. Chang, S. Gerrish, C. Wang, and D. Blei, "Reading tea leaves: How humans interpret topic models," in *Proc. 23rd Annu. Conf. Neural Inf. Process. Syst. (NIPS)*, 2009, pp. 288–296.
[54] M. D. Hoffman, D. M. Blei, and F. Bach, "Online learning for latent dirichlet allocation," in *Proc. 24th Annu. Conf. Neural Inf. Process. Syst.*, 2010, pp. 856–864.
[55] P. N. Garner, J. Dines, T. Hain, A. El Hannani, M. Karafiát, D. Korchagin, M. Lincoln, V. Wan, and L. Zhang, "Real-time ASR from meetings," in *Proc. Interspeech*, 2009, pp. 2119–2122.
[56] W. Webber, A. Moffat, and J. Zobel, "A similarity measure for indefinite rankings," *ACM Trans. Inf. Syst. (TOIS)*, vol. 28, no. 4, pp. 20:1–20:38, 2010.

Maryam Habibi received an M.Sc. (2010) in artificial intelligence and a B.Sc. (2008) in hardware engineering from Sharif University of Technology, Tehran, Iran.

She is currently a Research Assistant at the Idiap Research Institute (Martigny, Switzerland) and a Ph.D. Student at the Electrical Engineering department of the École Polytechnique Fédérale de Lausanne (EPFL) under the supervision of Prof. Hervé Bourlard and Dr. Andrei Popescu-Belis.

She has been a member of a team working on the design and implementation of a telephone call system for Tejarat Bank at ASR GooyeshPardaz Company (Tehran, Iran) and has worked at the communication department of IranAir Company (Tehran, Iran).

Andrei Popescu-Belis graduated from the École Polytechnique (Paris, France) in 1995, with majors in mathematics and computer science. He received an M.S. degree in artificial intelligence from the University of Paris VI in 1996, and a Ph.D. degree in computer science and natural language processing from LIMSI-CNRS, University of Paris XI, in 1999.

He is currently a Senior Researcher at the Idiap Research Institute (Martigny, Switzerland), a Lecturer at the École Polytechnique Fédérale de Lausanne (EPFL), and the head of Idiap's NLP group. He has been a Postdoc at UCSD and a Senior Research Assistant at ISSCO, University of Geneva.

Dr. Popescu-Belis has over 100 peer-reviewed publications in human language technology, information retrieval, and multimodal interactive systems, including two edited books. He has been involved in several large Swiss and international research projects.