Modeling Users' Information Needs in a Document Recommender for Meetings

Thèse n. 6760 (2015)
présentée le 30 novembre 2015
à l'Institut de Recherche Idiap
Faculté Sciences et Techniques pour l'Ingénieur (STI)
Programme Doctoral en Génie Électrique (EDEE)
École Polytechnique Fédérale de Lausanne

pour l'obtention du grade de Docteur ès Sciences
par

Maryam Habibi

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

acceptée sur proposition du jury:

Prof. Jean-Philippe Thiran, président du jury
Prof. Hervé Bourlard, directeur de thèse
Dr. Andrei Popescu-Belis, co-directeur de thèse
Dr. Jim Glass, rapporteur
Prof. Stéphane Marchand-Maillet, rapporteur
Dr. Martin Rajman, rapporteur

Lausanne, EPFL, 2015

"Respect your parents. What they tell you is true. Hard work, dedication and faith will get you anything. Imagination will drive itself. You can get anything you want, but you have to have faith behind all your ideas. Stick to your goals and have an undying faith."
— Russel Simmons

I dedicate my thesis work to my loving parents who always loved me unconditionally and whose words of encouragement and push for tenacity have taught me to work hard for the things that I aspire to achieve.
This work is also dedicated to my beloved husband who has been a constant source of support and encouragement during the challenges of graduate school and life. I am truly thankful for having you in my life.
I also dedicate this work and give special thanks to my brother. He has never left my side and is very special as well.

Acknowledgements

"Let us be grateful to people who make us happy; they are the charming gardeners who make our souls blossom."
— Marcel Proust

First and foremost my thanks go to Allah for providing me everything that I needed for this research.
I would like to express my deepest gratitude to my advisors Andrei Popescu-Belis and Hervé Bourlard for their kind inspiration, consultation, and understanding. I could not have been able to complete this work if Andrei did not patiently and seriously read my thesis and provided me with valuable comments to improve it.
I am also 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) and Sinergia MODERN project, as well as to the Hasler Foundation for the REMUS project (n. 13067, Re-ranking Multiple Search Results for Just-in-Time Document Recommendation).
Furthermore, the Idiap Research Institute and all its staff provided an environment of friendliness, flexibility, and enjoyable learning.
I am thankful to my thesis committee who reviewed the present thesis in great detail and provided insightful remarks and comments that greatly improved the quality and clarity of the final version.
To my colleague and friends at Idiap, I thank them for their companionship and for providing a so pleasurable and friendly working atmosphere. The moments of leisure shared together helped to overcome some more difficult moments.
To my closer friends, I express my special gratitude for their unconditional friendship, support, patience throughout these years.
Last, but not least, I would like to thank my family, my parents, my brother and my husband, for their unconditional support, encouragement and love, and without which I would not have come this far.

Lausanne, October 2015 M. H.

Abstract

People are surrounded by an unprecedented wealth of information. Access to it depends on the availability of suitable search engines, but even when these are available, people often do not initiate a search, because their current activity does not allow them, or they are not aware of the existence of this information. Just-in-time retrieval brings a radical change to the process of query-based retrieval, by proactively retrieving documents relevant to users' current activities, in an easily accessible and non-intrusive manner.

This thesis presents a novel set of methods intended to improve the relevance of a just-in-time retrieval system, specifically a document recommender system designed for conversations, in terms of precision and diversity of results. Additionally, we designed an evaluation protocol to compare the proposed methods in the thesis with other ones using crowdsourcing.

In contrast to previous systems, which model users' information needs by extracting keywords from clean and well-structured texts, this system models them from the conversation transcripts, which contain noise from automatic speech recognition (ASR) and have a free structure, often switching between several topics. To deal with these issues, we first propose a novel keyword extraction method which preserves both the relevance and the diversity of topics of the conversation, to properly capture possible users' needs with minimum ASR noise.

Implicit queries are then built from these keywords. However, the presence of multiple unrelated topics in one query introduces significant noise into the retrieval results. To reduce this effect, we separate users' needs by topically clustering keyword sets into several subsets or implicit queries.

We introduce a merging method which combines the results of multiple queries which are prepared from users' conversation to generate a concise, diverse and relevant list of documents. This method ensures that the system does not distract its users from their current conversation by frequently recommending them a large number of documents.

Moreover, we address the problem of explicit queries that may be asked by users during a conversation. We introduce a query refinement method which leverages the conversation context to answer the users' information needs without asking for additional clarifications and therefore, again, avoiding to distract users during their conversation.

Finally, we implemented the end-to-end document recommender system by integrating the ideas proposed in this thesis and then proposed an evaluation scenario with human users in a brainstorming meeting.

Keywords: Just-in-time Retrieval, Keyword Extraction, Diverse Merging of document lists, Query Refinement, Crowdsourcing.

Résumé

Jamais auparavant les humains n'avaient été entourés par autant d'information. Mais accéder à des informations pertinentes n'est possible que si des systèmes de recherche adaptés existent. Toutefois, même lorsque de tels systèmes sont disponibles, les utilisateurs n'initient souvent pas des recherches, soit parce que leur activité en cours ne leur permet de le faire, soit parce qu'ils ne pensent pas que de telles informations existent. Le paradigme de recherche d'information "juste à temps" se distingue radicalement des recherches à base de requêtes explicites. Cette approche propose de trouver des documents pertinents pour l'utilisateur sans la nécessité d'une requête explicite, de manière transparente et non-intrusive.

Cette thèse propose un ensemble de méthodes originales pour l'améliorer la pertinence d'un système de recherche d'information "juste à temps", et plus spécifiquement un système de recommandation des documents conçu pour des conversations. Les méhodes améliorent la précision et la diversité des résultats. De plus, la thèse propose un protocole d'évaluation pour comparer diverses méthodes fondé sur les jugements d'un grand nombre de sujets ("crowdsourcing").

Contrairement à de précédents systèmes qui modélisent les besoins d'information d'utilisateur par le biais de mots clés extraits de textes sans erreurs et clairement structurés, notre système modélise ces besoins à partir de la reconnaissance vocale des conversations. Or, celle-ci contient souvent de nombreuses erreurs de reconnaissance, et la nature des conversations fait qu'elles passent souvent d'un sujet à un autre. Pour répondre à ces défis, nous proposons d'abord une nouvelle méthode d'extraction de mots clés qui génère des mots clés à la fois pertinents et reflétant la diversité des sujets de la conversation. Ces mots clés représentent les besoins d'information des utilisateurs tout en réduisant le bruit dû à la reconnaissance vocale. Des requêtes implicites sont ensuite construites à partir de ces mots clés. Toutefois, dans la mesure où celles-ci peuvent inclure plusieurs sujets, nous proposons de séparer ceux-ci en regroupant les mots-clés selon leur sujet, formant ainsi plusieurs requêtes implicites.

Nous définissons également une méthode pour fusionner les résultats de plusieurs requêtes, préparées comme ci-dessus à partir d'un fragment de conversation, afin de générer une liste concise, diverse et pertinente de documents à recommander. La méthode évite d'interrompre les utilisateurs avec un grand nombre de suggestions.

De plus, nous traitons aussi le problème des requêtes explicites que les utilisateurs souhaiteraient poser au système. Nous proposons une méthode pour affiner les requêtes qui utilise les mots du contexte de la conversation, sans chercher à établir un dialogue avec les utilisateurs pour préciser ces requêtes, afin de ne pas perturber le cours de la conversation.

Enfin, nous avons implémenté le système de recommandation de documents, en intégrant toutes les idées proposées dans cette thèse, et avons proposé un scenario d'évaluation avec des utilisateurs humains en situation de réunion.

Mots clés : recherche d'information, extraction de mots clés, fusions de listes de résultats, affinage de requêtes.

Contents

List of Figures

List of Tables

Notations

General notations in the just-in-time retrieval framework

t A conversation fragment, pages 25–27, 35–41, 57, 85–87.

N Number of words in the conversation fragment t, pages 35–36.

T The set of conversation fragments used for evaluation contains the fragment t, pages 25–27, 85–87.

z, z_k A topic obtained by LDA topic modeling, pages 34–41, 57–58, 67–68, 82–84.

Z The set of topics contains the topic z, pages 34–41, 57–58, 67–68, 82–84.

$p(z|w)$ The distribution over the topic z of the word w, pages 34–41, 57–58.

$p(w|z)$ The topic-word distribution, the contribution of the word w in the construction of the topic z, pages 67–67.

$p(z|d)$ The document-topic distribution, the distribution of topic z in the document d with respect to other topics, pages 67–67.

β_z The weight of the topic z in a fragment t, pages 34–41, 57–58.

c, c_i A keyword selected from a conversation fragment t, pages 34–41, 57–58, 67, 82–84.

C A set of keywords extracted from a conversation fragment t, pages 34–41, 57–58, 67, 82–84.

k Number of keywords in the keyword set C, pages 36–41.

$r_{C,z}$ The contribution of keyword set C towards the topic z, pages 36–41.

$R(C)$ The cumulative reward value of the keyword set C over the set of topics Z, pages 36–41.

λ The scaling exponent of the diverse reward function called the diversity factor, pages 36–41.

$h(w, C)$ The greedy algorithm maximizes the cumulative reward function $R(C)$, pages 40–41.

$s_{c_i,z}$ The topical similarity of the keyword c_i to the entire conversation fragment t with respect to the topic z, pages 57–58.

Q The union of implicit queries called the collective query, page 67.

Q_i An implicit query, pages 57–58, 67–71.

we_{im,Q_i} The weight of the implicit query Q_i, pages 67–68, 67–71.

$Q_{implicit}$ The set of implicit queries contains Q_i, pages 57–58, 67–70.

M Number of implicit queries, pages 57–58, 67–70.

d a document retrieved for a query, pages 67–71.

l_i The list of relevant documents to the implicit query Q_i, pages 67–71.

we_{l_i} The weight of the list l_i , pages 67–71.

L The set of document lists retrieved for the set of implicit queries $Q_{implicit}$, pages 67–71.

$P(Q_i)$ The topical representation of the implicit query Q_i, pages 67–71.

$P(d_j)$ The topical representation of the document d_j, pages 67–71.

S The set of the most representative documents recommended to users, pages 67–71.

$r_{S,i}$ The topical similarity of a subset of S which are selected from the list l_i to the collective query Q, pages 68–71.

$R(S)$ The cumulative reward value of the document set S over all the lists, pages 68–71.

$g(d,S)$ The greedy algorithm maximizes the cumulative reward function $R(S)$, pages 69–71.

$Q_{explicit}$ An explicit query asked by a user, pages 82–84.

$w_{ex,i}$ One the words which forms the explicit query $Q_{explicit}$, pages 82–84.

we_{c_i} The topical similarity of an expansion word c_i to the explicit query $Q_{explicit}$, pages 82–84.

γ The expansion parameter, pages 82–84.

$RQ(\gamma)$ The expanded query with parameter γ, pages 82–84.

Notations related to the evaluation protocol for implicit queries

v A worker or a subject recruited for evaluation using crowdsourcing, pages 25–27.

V A set of workers contains v, pages 25–27.

a An option judged by the worker v, pages 25–27, 85–87.

A A set of options contains a, pages 25–27, 85–87.

$S_{tv}(a)$ The judgment of the worker v to the option a of the fragment t, pages 25–27.

$\mu_{S_{tv}(a)}$ The expected value of the judgment of the worker v to the option a over all the fragments, pages 25–27.

$\sigma_{S_{tv}(a)}$ The standard deviation of the judgment of the worker v to the option a over all the fragments, pages 25–27.

$S'_{tv}(a)$ The averaged judgments of all workers to the option a of the fragment t, pages 25–27.

$\mu_{S'_{tv}(a)}$ The expected value of the variable $S_t(a)$ over all the fragments, pages 25–27.

$\sigma_{S'_{tv}(a)}$ The standard deviation of the variable $S_t(a)$ over all the fragments, pages 25–27.

c_v The reliability value assigned to the judgments of the worker v, pages 25–27.

$CS_t(a)$ The comparative relevance value of the option a for the fragment t, pages 25–27.

H_t The entropy of all workers' judgments for the fragment t, pages 25–27.

D_t The level of task difficulty for the fragment t to be judged by workers, pages 25–27.

$CS(a)$ The comparative relevance score over all fragments, pages 25–27.

e_v The agreement between the worker v with the expert, pages 27–28.

z_t A mono-topic dialogue represented by a document from the Fisher Corpus, page 44.

Z_t The set of mono-topic dialogues in a conversation fragment artificially made from the Fisher Corpus, page 44.

$Nr_{z_t,i-1}$ Number of relevant keywords to the mono-topic dialogue z_t up to the rank i, page 44.

$Jr(c_i, z_t)$ The relevance of the keyword c at the rank i to the mono-topic dialogue z_t, page 44.

α A factor to measure diversity using the α-NDCG evaluation method, page 44.

Notations regarding the evaluation of explicit queries

$s_{t,j}(a)$ The proportion of the selection of the choice a for the document j in the task designed for the fragment t by all the workers, pages 85–87.

$H_{t,j}$ The entropy of workers' judgments for the document j and the fragment t, pages 85–87.

$s'_{t,j}(a)$ The $s_{t,j}(a)$ score weighted by a function of $H_{t,j}$, pages 85–87.

gr_{tj} The global relevance value for the document j and the fragment t, pages 85–87.

$AveP_{tO}(k')$ The average precision of the system O at the rank k' for the fragment t, pages 85–87.

$MAP_O(k')$ The mean average precision of the system O at the rank k', pages 85–87.

$RS_{O_1,O_2}(k')$ The relative MAP score improvement of the system O_1 over the system O_2 at the rank k', pages 85–87.

EC A subset of the keyword set C contains keywords from ASR noise, page 90.

pn The proportion of noisy keywords added by a query refinement method to an explicit query, page 90.

1 Introduction

Human beings face an unexpectedly high volume of information, available as documents, databases, or multimedia resources. However, humans often do not initiate a search to access new information, because their current activity does not allow them to do so, or because they are not aware that relevant information is available.

Just-in-time retrieval systems automate the process of information access. They continuously look at their users' activities to capture their information needs, and proactively retrieve information that is potentially relevant to their information needs. These distinctive features make them different from ordinary search engines and personalized software which require a query from users.

In this thesis, we develop our research study within the framework of the Automatic Content Linking Device (ACLD), which is a particular just-in-time retrieval system intended to be used in conversations, specifically by a small group of people interacting in a meeting. The main motivation to design such a just-in-time retrieval system is to help meeting participants to find documents from the Web or from local databases which contain facts related to the current conversation fragment, while the participants do not have the time to search for these facts during their conversation. Therefore, the ACLD assists them by running the search in the background, and offering the potentially valuable results to the participants. The participants can easily and quickly refer to the results when they need them, which might happen at crucial moments during a meeting.

1.1 Initial Framework of the ACLD

The initial version of the ACLD constantly listens to the users' conversation and proactively retrieves documents that might be useful to them, in real time. The ACLD receives as input raw words from an Automatic Speech Recognition (ASR) system at regular time intervals. The raw words are preprocessed by stop word removal and word stemming. Then, the ACLD attempts to represent users' information needs as "implicit queries". These queries are made of sets of keywords extracted from the conversation fragment by simply matching the preprocessed ASR output against a predefined list of content words. The queries are submitted to a retrieval system to suggest documents from the Web or local databases (e.g. including fragments of previous conversations) to the users.

Similar to other just-in-time retrieval systems, the ACLD displays results in a condensed form, along with other types of information. The system displays: (1) the transcript of the conversation; (2) the keywords extracted from the transcript that were used to build implicit queries, in a tag-cloud format; the links to the recommendations obtained by running implicit queries on a search engine, either (3) over a local database or (4) over the Web. Users can easily have access to the content of documents by clicking on the links.

1.2 Motivation

In the present thesis, we propose a set of original, theoretically-grounded techniques to increase the low precision of the initial ACLD in terms of retrieval results. In addition, we enable the system to answer queries which are explicitly asked by users, because the initial ACLD cannot properly answer them. Although we have shown that the results of explicit queries are more precise than those of obtained by implicit ones in case users do formulate explicit queries, the explicit ones require more effort from users. Our recommendation system can implicitly prepare queries from the users' conversation, so users do not need to make the effort of formulating queries. Using implicit queries, the system may provide some results that users may be not aware of. In this section, we discuss the limitations of the initial version of the ACLD for answering both implicit queries and explicit ones, and explain why existing techniques cannot overcome these issues.

In contrast to the short queries which are explicitly addressed to the general commercial web search engines, the just-in-time-retrieval systems must construct implicit queries from the words that are written or spoken by users during their activities, which contain a much larger number of words than the short queries. Moreover, the retrieval results should be presented in a non-intrusive manner (typically a very concise list) to avoid distracting users. The just-in-time retrieval systems should also be able to refine explicit queries without asking for additional clarifications from users to avoid users interruption during their main activities. These differences with ordinary retrieval systems open the door for new research toward just-in-time retrieval systems.

The methods used in previously proposed just-in-time retrieval systems are not directly applicable to systems intended to be used in a conversation, such as the ACLD. Previous just-in-time retrieval systems build implicit queries from written documents which are mostly planed, while the ACLD must build its implicit queries from the ASR transcript of users' conversation. The conversational ASR transcripts can contain off-topic content which is introduced by ASR noise. In addition to the detrimental effect of the ASR noise on retrieval results, the implicit query may often refer to several topics of interest, as the conversations are usually unplanned, with users turning from one topic to another. To answer this problem, in this thesis, we propose a new keyword extraction technique which automatically identifies the main topics of a conversation fragment, and then extracts keywords for implicit queries by rewarding both their diversity and their relevance to the main topics, in order to maximize the coverage of the main topics that were mentioned in the conversation fragment.

Previous just-in-time retrieval systems prepare a single implicit query from the entire keyword set extracted from a written document, which is generally focused on a single topic. On the contrary, we propose to prepare several topically-separated implicit queries by dividing the keyword set extracted from each fragment into several subsets, using a topic-aware clustering method. As mentioned above, since conversations are usually unfocused and unplanned, they mention multiple topics; there is a risk that the mixture of topics in a single query will degrade the retrieval results of an implicit query. As we will show, the representation of queries as multiple topically-separated queries improves the retrieval results of the system.

Moreover, just-in-time retrieval systems should present a concise list of documents because users are regularly unwilling to examine a large number of recommended documents – in other words, examining a long list of results distracts the users from their conversation. The solution which was used by previous just-in-time retrieval systems, for a single implicit query, was to cluster its retrieval results and then to select the best representative from each cluster to display. However, this method is only applicable when the clusters have comparable levels of importance. Inspired by previous diverse retrieval re-ranking techniques for single queries with multiple aspects, we propose a new diverse merging technique which is applicable to multiple topically-separated queries. The method merges the retrieval results of implicit queries by rewarding at the same time the topical similarity of documents to the queries as well as the diversity over lists of documents.

In the case of explicit short queries asked by users from a just-in-time retrieval system, the retrieval results can be erroneous in a similar way as other short queries addressed to a general purpose search engine. Here, the use of the context could help to better determine the users' information needs. Several techniques have been proposed in the field of information retrieval for the refinement of explicit short queries. They interactively or automatically select relevant interpretations of queries obtained from a data source or a relevant part of it. However, users of the ACLD might be reluctant or unable to interactively specify their real needs during their activities. Moreover, using a data source outside the users' local context may cause the real users' intent to be misinterpreted. Existing solutions take advantage of users' local context and refine explicit queries by inserting keywords elicited from written documents in the context, which are (again) planned and focused. However, this is less suitable for conversational environments, because of the nature of the vocabulary and the errors introduced by the ASR system. In this thesis, we propose a query refinement technique which inserts keywords extracted from the transcript of the conversation fragment preceding the query, along with a weight value that represents the topical similarity of each expansion keyword to the explicit query.

1.3 Contributions of the Thesis

In this section, we first present an overall perspective on the thesis by situating its contributions in the framework of the initial ACLD system described above (1.3.1). Then, we provide a detailed overview of the content of the thesis, divided into chapters (1.3.2).

1.3.1 Proposed Methods for the ACLD

In this thesis we propose new approaches for the ACLD, as schematically shown in Fig 1.1. Starting from the users' conversation transcribed using ASR and divided into fragments, we prepare implicit queries, at regular time intervals, with the following approach. Similarly to the initial version, first we apply stop word removal and stemming techniques to extract meaningful words from the transcript. However, in contrast to the existing system[1], we propose a two-stage approach to the formulation of implicit queries, as represented in Figure 1.2.

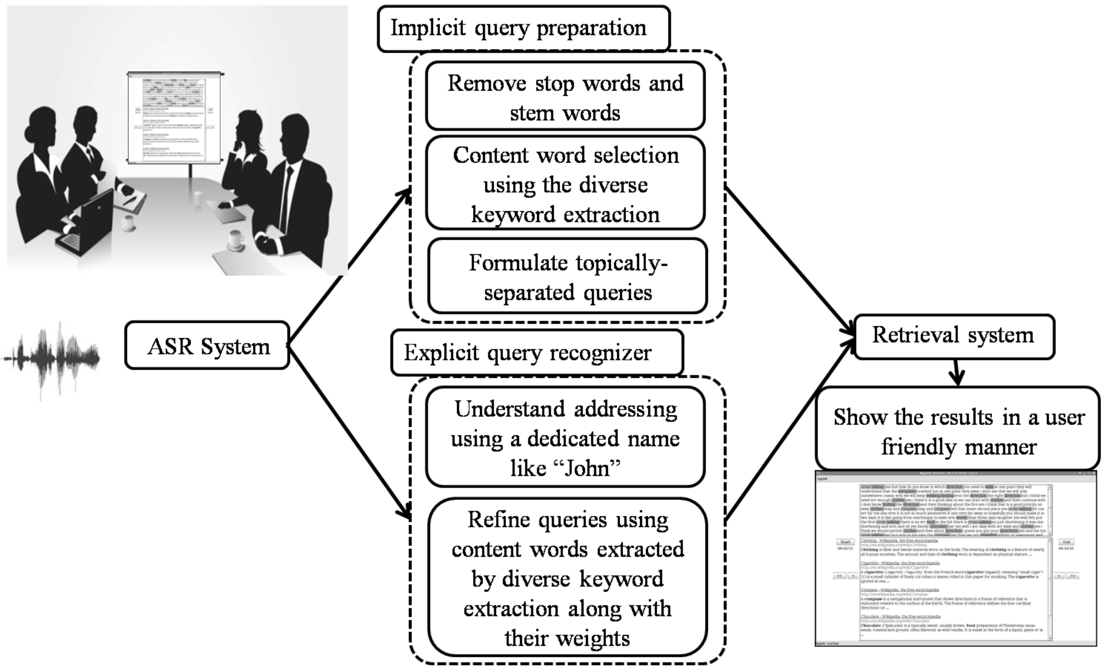

Implicit query preparation
Remove stop words and stem words
Content word selection using the diverse keyword extraction
Formulate topically-separated queries
ASR System
Explicit query recognizer
Understand addressing using a dedicated name like "John"
Refine queries using content words extracted by diverse keyword extraction along with their weights
Retrieval system
Show the results in a user friendly manner

Figure 1.1: Representation of the proposed ACLD system. The ACLD receives fragments of ASR transcript of users' discussion, and prepares implicit queries by: (1) applying stopword removal and stemming; (2) performing diverse keyword extraction; and (3) constructing multiple topically-separated queries. Then, the retrieval results of the implicit queries are merged into a short and concise list of documents. Moreover, the system can recognize explicit queries and expand them using the context. Finally, the system displays and justifies through keywords the retrieval.

In the first stage, we propose a diverse keyword extraction technique from the list of content

[1] Which formulates each implicit query by selecting a subset of meaningful words based on matching to a predefined list of keywords.

words through steps 1 to 3 of Figure 1.2. 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 into the form of several topically-disjoint queries to reduce the noisy effect of mixture of topics in a query in step 4 of Fig 1.2. Briefly, at step 1, a topic model is used to represent the distribution of the abstract topic z for each word w noted $p(z|w)$. 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. At step 2, these topic models are used to determine weights for the abstract topics in each conversation fragment represented by β_z. At step 3, the keywords $C = \{c_1, ..., c_k\}$ which cover a maximum number of the most important topics are selected by rewarding diversity, using an original algorithm. Finally, at step 4, the implicit queries are constructed by clustering the keyword set into several topically-separated subsets, each one corresponding to an abstract topic. Each subset forms an implicit query, Q_i, and is weighted by we_{im,Q_i} based on the importance of the topic to which it is associated.

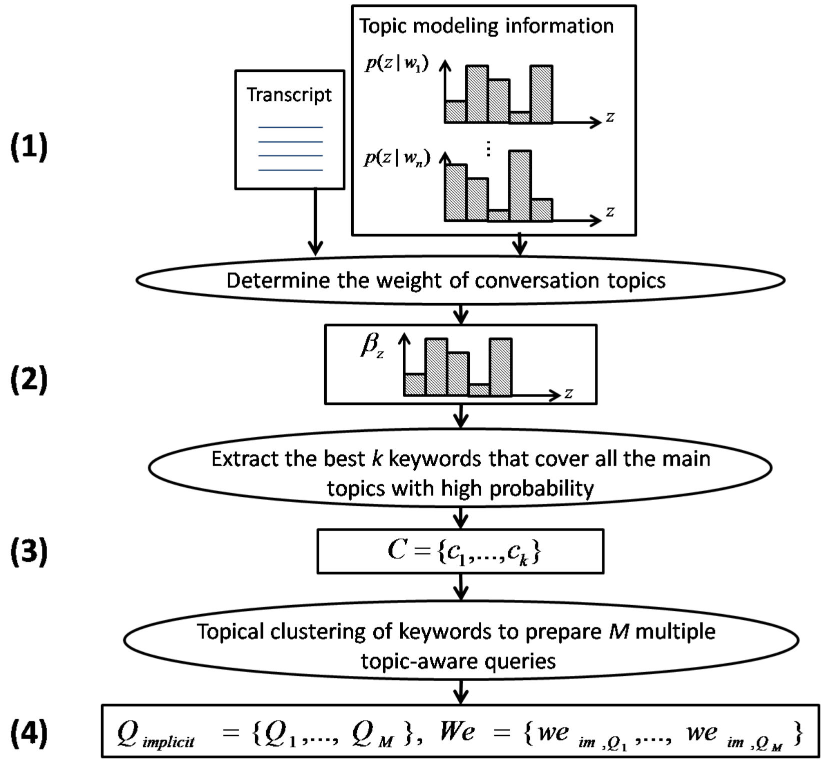

(1)
Transcript
Topic modeling information
$p(z|w_1)$
z
$p(z|w_n)$
z
Determine the weight of conversation topics
(2)
β_z
z
Extract the best k keywords that cover all the main topics with high probability
(3)
$C = \{c_1, ..., c_k\}$
Topical clustering of keywords to prepare M multiple topic-aware queries
(4)
$Q_{implicit} = \{Q_1, ..., Q_M\}, \ We = \{we_{im,Q_1}, ..., we_{im,Q_M}\}$

Figure 1.2: The four steps of the proposed implicit query formulation method: (1) topic modeling; (2) representation of the main topics of the transcript; (3) diverse keyword selection; and (4) formulation of topically-separated implicit queries along with their weights.

To obtain document results from the implicit queries, similarly to the existing ACLD, we submit each implicit query to a search engine, as shown in the first step of Figure 1.3. However, to manage several lists of relevant articles (each retrieved for one implicit query), we propose a diverse merging method leading to a concise list of results shown to users. The merging algorithm rewards diversity by decreasing the gain of selecting documents from a list as the number of its previously selected documents increases. First, we represent the union of the implicit queries constructed for the conversation fragment and the lists of document results using topic modeling information as shown in the second step of Figure 1.3. Then we merge documents by rewarding the topical similarity of documents to the queries as well as the coverage of different lists based on the importance of the list which is specified by the weight of implicit queries (step 3 of Fig. 1.3) normalized over the sum of the weights of all implicit queries prepared for a fragment.

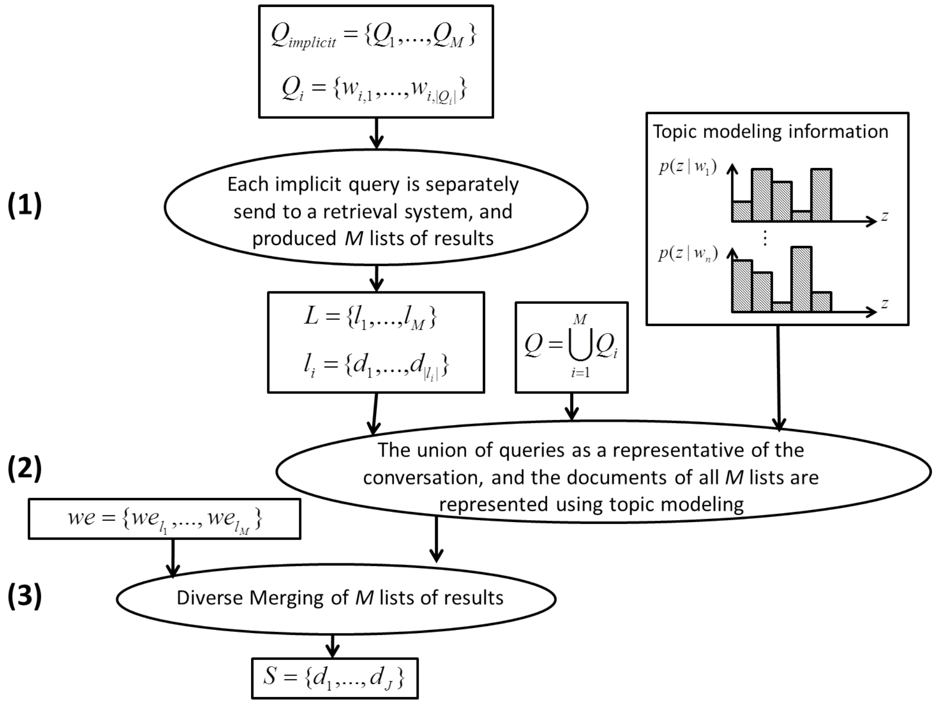

$Q_{implicit} = \{Q_1,...,Q_M\}$
$Q_i = \{w_{i,1},...,w_{i,|Q_i|}\}$
(1)
Each implicit query is separately send to a retrieval system, and produced M lists of results
Topic modeling information
$p(z|w_1)$
$p(z|w_n)$
z
$L = \{l_1,...,l_M\}$
$l_i = \{d_1,...,d_{|l_i|}\}$
$Q = \bigcup_{i=1}^{M} Q_i$
(2)
The union of queries as a representative of the conversation, and the documents of all M lists are represented using topic modeling
$we = \{we_{l_1},...,we_{l_M}\}$
(3)
Diverse Merging of M lists of results
$S = \{d_1,...,d_J\}$

Figure 1.3: The three steps to prepare a concise list of documents from the retrieval results of several topically-separated queries: (1) separately submitting M implicit queries to a retrieval system to create M lists of results; (2) representing the document results and the set of keywords C using topical information; and (3) merging the retrieval results of M lists to prepare the final list of recommendations.

We have taken advantage of the above theoretical framework to allow the ACLD to also process explicit queries submitted to it. The users can simply address the system by using a pre-defined unambiguous name, which is robustly recognized by the ASR (e.g. "John"). We expand the explicit queries using the keywords extracted by the proposed diverse keyword extraction method, as shown in Fig 1.4. In the first step, the explicit query with the terms $w_{ex,i}$ and the keywords in the set C which are extracted from the conversation are represented using topical information. In the second step, each keyword in the list $c_i \in C$ is given a weight based on its topical similarity to the entire query (noted we_{c_i}). In the final step, this keywords along with their weights are added to the explicit query.

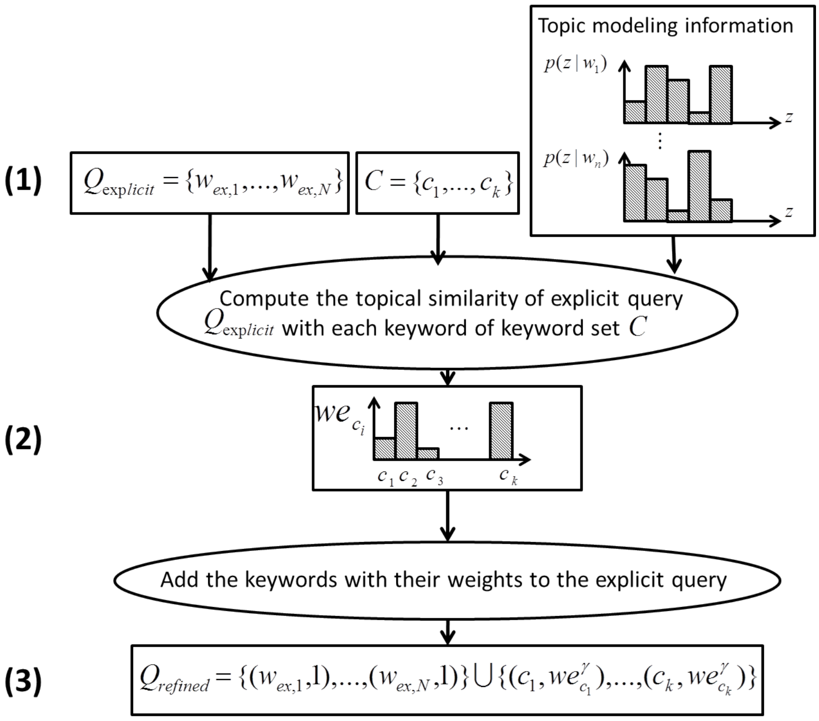
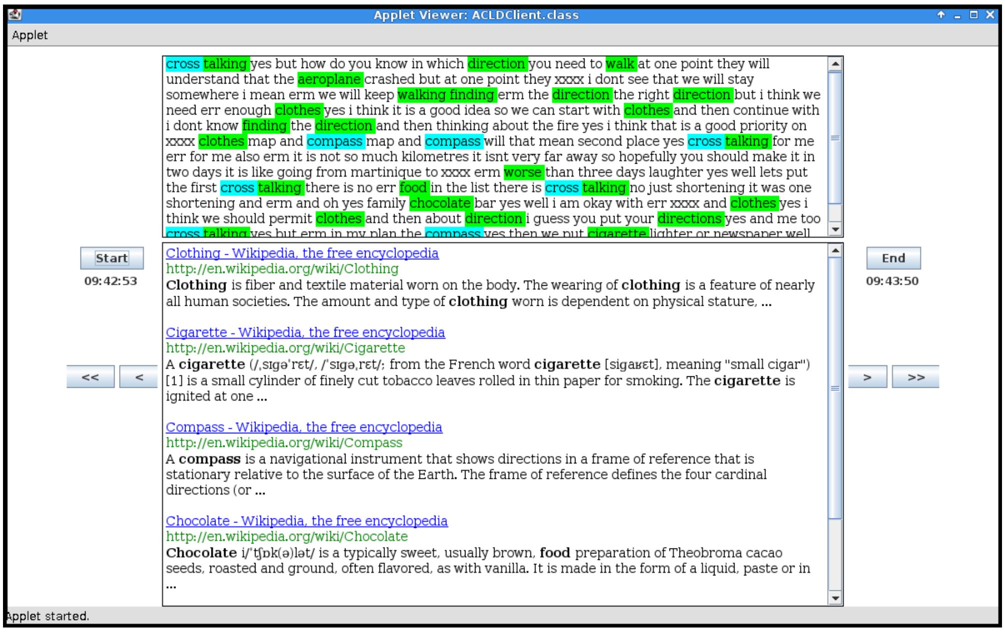

Topic modeling information
$p(z \mid w_1)$
z
$p(z \mid w_n)$
z
(1)
$Q_{explicit} = \{w_{ex,1},\ldots,w_{ex,N}\}$
$C = \{c_1,\ldots,c_k\}$
Compute the topical similarity of explicit query $Q_{explicit}$ with each keyword of keyword set C
(2)
we_{c_i}
c_1 c_2 c_3 c_k
Add the keywords with their weights to the explicit query
(3)
$Q_{refined} = \{(w_{ex,1},1),\ldots,(w_{ex,N},1)\} \cup \{(c_1, we^{\gamma}_{c_1}),\ldots,(c_k, we^{\gamma}_{c_k})\}$

Figure 1.4: The three stages of the proposed query refinement method: (1) topic modeling; (2) computation of topical similarity of keywords to the explicit query; and (3) appending the keywords to the explicit query along with their weights.

Given the new proposed approach to document recommendation for conversations, we also designed a new user interface, shown in Figure 1.5. In the interface, users can easily observe the ASR transcript of their conversation as well as the keywords of their conversation, highlighted in green, as a summary of their discussion. Moreover, the links to the relevant document results, along with the first sentence of each of them, marked with the keywords found in the transcript, are displayed as well. To provide an explanation of the results, we show the content words relevant to each document in the transcript highlighted in cyan when the mouse hovers over the link to the document. Moreover, users can move forward or backward through the results by pressing specific buttons.

1.3.2 Thesis Outline

This introduction (**Chapter 1**) aims to provide the big picture of the problem, of the motivations underlying this study, and of the specific goals and contributions of the thesis. The remaining chapters of the thesis are organized as follows. In **Chapter 2**, the context of this study is reviewed, i.e. the literature published on the subject. We survey the existing just-in-time information retrieval systems along with the approaches they used to formulate implicit queries from users current activity. Moreover, we review the current keyword extraction methods, diverse retrieval re-ranking or merging techniques, and query refinement policies. We



cross talking yes but how do you know in which direction you need to walk at one point they will understand that the aeroplane crashed but at one point they xxxx i dont see that we will stay somewhere i mean erm we will keep walking finding erm the direction the right direction but i think we need err enough clothes yes i think it is a good idea so we can start with clothes and then continue with i dont know finding the direction and then thinking about the fire yes i think that is a good priority on xxxx clothes map and compass map and compass will that mean second place yes cross talking for me err for me also erm it is not so much kilometres it isnt very far away so hopefully you should make it in two days it is like going from martinique to xxxx erm worse than three days laughter yes well lets put the first cross talking there is no err food in the list there is cross talking no just shortening it was one shortening and erm and oh yes family chocolate bar yes well i am okay with err xxxx and clothes yes i think we should permit clothes and then about direction i guess you put your directions yes and me too cross talking yes but erm in my plan the compass yes then we put cigarette lighter or newspaper well

Clothing - Wikipedia, the free encyclopedia
http://en.wikipedia.org/wiki/Clothing
Clothing is fiber and textile material worn on the body. The wearing of **clothing** is a feature of nearly all human societies. The amount and type of **clothing** worn is dependent on physical stature, ...

Cigarette - Wikipedia, the free encyclopedia
http://en.wikipedia.org/wiki/Cigarette
A **cigarette** (/ˌsɪɡəˈrɛt/, /ˈsɪɡəˌrɛt/; from the French word **cigarette** [siɡaʁɛt], meaning "small cigar") [1] is a small cylinder of finely cut tobacco leaves rolled in thin paper for smoking. The **cigarette** is ignited at one ...

Compass - Wikipedia, the free encyclopedia
http://en.wikipedia.org/wiki/Compass
A **compass** is a navigational instrument that shows directions in a frame of reference that is stationary relative to the surface of the Earth. The frame of reference defines the four cardinal directions (or ...

Chocolate - Wikipedia, the free encyclopedia
http://en.wikipedia.org/wiki/Chocolate
Chocolate i/ˈtʃɒk(ə)lət/ is a typically sweet, usually brown, **food** preparation of Theobroma cacao seeds, roasted and ground, often flavored, as with vanilla. It is made in the form of a liquid, paste or in ...

Figure 1.5: The new user interface of the ACLD. The interface shows reasons for each suggested document, in terms of keywords, in addition to displaying transcripts and content words.

also review the possible evaluation methods for this study. Through Chapters 3, 4, 5, 6, and 7 (which are briefly summarized below), we describe the proposed solutions along with data and evaluation metrics used for their evaluation. We also provide the results obtained by comparing our solutions with the baselines. Then in **Chapter 8**, we present a scenario for user-centric evaluation of the end-to-end system we implemented, including a new interface, along with the results of a pilot experiment. In **Chapter 9**, the main findings of the thesis are summarized, and future directions are considered to address several remaining issues among the goals of this thesis. At the end, in appendices, we first present a method we proposed for improving the results of machine translation by taking advantage of the keywords extracted using our diverse keyword extraction method. Then, we provide the transcripts for the examples given at the end of several chapters.

Chapter 3: Comparative Evaluation Method Using Crowdsourcing

In Chapter 3, we explore a crowdsourcing approach to evaluate the retrieval results of a just-in-time retrieval system for conversational environments. To validate the approach, we compare several initial versions of the ACLD, including one that uses the entire conversation fragment as a query and two which extract keywords from the fragment and use them directly as a query. We employed two keyword extraction techniques which were used by the initial version of the ACLD.

The chapter defines a comparison method to compare the retrieval results of two different methods used for query formulation, using a crowdsourcing platform, Amazon's Mechanical

Turk, as an alternative to hiring expert workers for offline evaluation and to setting up real meetings to collect clickthrough data for online evaluation. Crowdsourcing, as an offline evaluation method, allows researchers to easily prototype and test their systems, in addition to being a cheap and fast approach. We first design and publish comparison tasks "Human Intelligence Tasks" (HITs) to gather users judgments. Each HIT demonstrates the transcript of a conversation fragment with two lists of documents to be compared to users. Then users should compare the lists by selecting options provided in the HIT (e.g. for a four option HIT design workers should select among these options: "X" is better than "Y", "Y" is better than "X", both are equally good, both are equally poor).

To measure the improvement brought by one method over the others with sufficient confidence using workers' judgments, various qualification control techniques exist. However, they do not consider both workers reliability and the level of task difficulty, or if they consider them, they need a large amount of data with ground truth annotation for each task and worker.

In this chapter, we introduce a new qualification control method called PCC-H which considers workers' reliability and the difficulty of the task at the same time, without the need for large ground truth data sets to validate the workers' judgments. We measure the workers' reliability by the inter-rater agreement of each of them against all the others, and use entropy to weight the difficulty of each comparison task. It is shown that the proposed evaluation method provides similar comparison scores for two different task designs and also is reliable when compared to human judgments. The method is utilized for the comparative evaluation of the methods proposed throughout this thesis.[2]

Chapter 4: Diverse Keyword Extraction from Conversational Transcripts

In Chapter 4, we present a keyword extraction method from conversations which preserves the diversity of topics that may appear even in a short conversation fragment, with the overall goal of providing a set of keywords that are representative of the semantic content of each fragment. As the maximum coverage problem is *NP*-hard, we proposed a new submodular reward function which rewards both the diversity and the relevance of topics in the set of keywords in order to find the keyword set in polynomial time. The proposed method, which is inspired from recent summarization methods, maximizes the coverage of topics that are recognized automatically in manual or ASR transcripts of conversation fragments.

The method is evaluated on excerpts of manual transcripts of the Fisher Corpus and both the manual and the ASR transcripts of conversation fragments from the AMI Meeting Corpus. As presented in Chapter 3, we use crowdsourcing to elicit a large number of comparative judgments for sets of keywords, aiming to evaluate which set is most representative of a fragment. To enhance the readability of the keyword lists, we present a word cloud representation of them to the workers.

[2] This method was presented at the Workshop on Recommendation Utility Evaluation (RUE 2012), a satellite workshop of the 6th Conference on Recommender Systems (RecSys) (Habibi and Popescu-Belis, 2012).

The results demonstrate that our method outperforms two competitive baselines, one based on word frequency, and the other one considering topics but not enforcing diversity. We also show that the final keyword lists that we extract contain a smaller number of words from the ASR noise compared to those obtained by other competitive methods. [3]

To demonstrate applicability to other settings, the method was used to represent the content of a lecture segment, or an entire lecture, within the MUST-VIS system for multimedia navigation and recommendation.[4] The results show that the recommendations automatically provided by the MUST-VIS system are judged to be relevant by human experts.

Chapter 5: Formulation of Implicit Queries from Conversations

In Chapter 5, we address the problem of formulating implicit queries from a set of keywords – extracted as in Chapter 4 above – with the goal of using the keywords to retrieve, for each short conversation fragment, a small number of potentially relevant documents for the conversation participants. We construct two types of implicit queries for each conversation fragment from a keyword list. The first one is simply the entire keyword list, while the second one is a set of queries obtained by dividing the list into several topically-separated subsets. The queries are given to a standard search engine over the English Wikipedia.

We experimented with the manual transcripts of the ELEA conversational corpus (see Section 5.3), because there are enough articles in the Wikipedia to retrieve for the discussion of the ELEA Corpus, and people also jump from one topic to another even in short fragments. We use crowdsourcing to evaluate our method in terms of the relevance or the utility of suggested documents to the meeting participants at the time of the corresponding fragment.

We first show that the retrieval results of the single queries made of the keyword list obtained by the method proposed in Chapter 4 outperforms those of the baseline keyword extraction methods. Then we compare the retrieval results of single queries with the retrieval results of the multiple queries (merged into a single result list). The results show that the representation of users' information needs with multiple topically-separated queries and then merging the retrieval results of queries, each submitted separately to a search engine improve the final list of results.[5]

[3] We presented an initial version of our proposal at the 51th Annual Meeting of the Association for Computational Linguistics (ACL) as a short paper (Habibi and Popescu-Belis, 2013), while the complete study appears in our article in the *IEEE/ACM Transactions on Audio, Speech and Language Processing* (Habibi and Popescu-Belis, 2015b).

[4] The system was the winner of the ACM Multimedia 2013 Grand Challenge on Temporal Segmentation and Annotation (Bhatt et al., 2013).

[5] These results were published as part of our article in the *IEEE/ACM Transactions on Audio, Speech and Language Processing* (Habibi and Popescu-Belis, 2015b).

Chapter 6: Diverse Merging of Document Lists

In Chapter 6, we propose a solution to the problem of building concise, diverse and relevant lists of documents, which can be recommended to the participants of a conversation. These lists are retrieved periodically by submitting multiple implicit queries derived from the pronounced words as described in Chapter 5. Each query is related to one of the topics identified in the conversation fragment preceding the recommendation, and is submitted to a search engine (in our experiments, over the English Wikipedia).

We propose an algorithm for diverse merging of these lists, using a submodular reward function that rewards the topical similarity of documents to the conversation keywords as well as the diversity over lists of document results. We evaluate the proposed method through crowdsourcing over the manual transcript of the ELEA conversational corpus. The results show the superiority of the diverse merging technique over several others which not enforce both topical relevance and diversity simultaneously.[6]

Chapter 7: Refinement of Explicit Queries

In Chapter 7, we introduce a context-based query refinement method applied to queries asked by users during a meeting or a conversation, thus extending the capabilities of the ACLD to answer spoken explicit queries. The proposed technique first implicitly extracts keywords from users' conversation fragments preceding explicit queries to represent the local context, and then refines the queries (actually expanding them) by inserting into them these keywords along with a weight value based on their topical similarity to the initial query.

To evaluate our proposal, we constructed a dataset called AREX: AMI Requests for Explanations and Relevance Judgments for their Answers. This dataset contains a set of explicit queries inserted in several conversations of the AMI Meeting Corpus, along with a set of relevance judgments, over sample retrieval results from Wikipedia, collected by crowdsourcing. Moreover, an automatic evaluation metric based on Mean Average Precision (MAP) is provided with the AREX dataset.

We compare our query expansion approach with other methods in terms of the relevance of the retrieved documents using the AREX data set and metric. This comparison indicates the superiority of our method using either the manual or the ASR transcripts.[7]

[6] This work was presented at the 25th International Conference on Computational Linguistics (Coling), as a long paper (Habibi and Popescu-Belis, 2014).

[7] This work was presented at the 20th International Conference on Applications of Natural Language to Information Systems (NLDB), as a long paper (Habibi and Popescu-Belis, 2015a).

2 Related Work

In this chapter, we first review the previous work related to retrieval evaluation methods, especially based on crowdsourcing, in Section 2.1. Then we review existing just-in-time retrieval systems and the policies they use for automatically understanding users' information needs and building implicit queries to represent them, in Section 2.2. This review motivates us to propose a new keyword extraction technique from conversational transcripts produced by automatic speech recognition (ASR), in order to formulate implicit queries. Hence, we analyze previous methods for keyword extraction from a text or transcript in Section 2.3. In Section 2.4 we survey current techniques for merging and re-ranking lists of search results which are applicable to our just-in-time information retrieval system for conversations. Additionally, we study the main methods which have been proposed for the refinement of short queries that are explicitly asked by users in Section 2.5.

2.1 Retrieval Evaluation Methods

Evaluating the relevance of retrieval results is a difficult task, because it is subjective and expensive to perform. Two well-known methods which are frequently used for this task are the use of clickthrough data or the use of human experts (Thomas and Hawking, 2006). However, in our case, producing clickthrough data or hiring professional workers for relevance evaluation of the results suggested by our just-in-time retrieval system would be overly expensive and challenging.

Moreover, it is not clear that evaluation results provided by a narrow range of experts would be generalizable to a broader range of end users. In contrast, crowdsourcing is relatively easy to prototype and to test experimentally, and also provides a cheap and fast approach for offline evaluation. However, it is necessary to consider some problems which are associated to this approach, mainly the reliability of the workers' judgments (including spammers) and the intrinsic competencies of the workers (Alonso and Lease, 2011).

The Technique for Evaluating Relevance by Crowdsourcing (TERC, see Alonso et al. (2008))

emphasized the importance of qualification control, e.g. by creating qualification tests that must be passed before performing the actual task. However, another study (Alonso and Baeza-Yates, 2011) observed that workers may still perform tasks randomly even after passing qualification tests. Therefore, it is important to perform partial validation of each worker's tasks, and weight the judgments of several workers to produce aggregate scores (Alonso et al., 2008).

Several other studies have focused on Amazon's Mechanical Turk crowdsourcing platform and have proposed techniques to measure the quality of workers' judgments when there is no ground truth to verify them directly (Carletta, 1996; Smyth et al., 1994; Chittaranjan et al., 2011; Karger et al., 2011; Whitehill et al., 2009; Khattak and Salleb-Aouissi, 2011). For instance, Carletta (1996) measured the quality of judgments for a labeling task using the inter-rater agreement and majority voting. Alternatively, expectation maximization (EM) has been used to estimate true labels in the absence of ground truth for an image labeling task (Smyth et al., 1994). In order to improve EM-based estimation of the reliability of workers, the confidence of workers in each of their judgments has been used as an additional feature – the task being dominance level estimation for participants in a conversation (Chittaranjan et al., 2011). As the performance of the EM algorithm is not guaranteed, a new method was introduced by Karger et al. (2011) to estimate reliability based on low-rank matrix approximation.

All of the above-mentioned studies assumed that tasks share the same level of difficulty. To model both task difficulty and workers' reliability, an EM-based method named GLAD (for Generative model of Labels, Abilities, and Difficulties) was proposed by Whitehill et al. (2009) for an image labeling task. However, this method is sensitive to the initialization value, hence a good estimation of labels requires a small amount of data with ground truth annotation (Khattak and Salleb-Aouissi, 2011). In this thesis (see Chapter 3), we will introduce a qualification control technique which predicts both task difficulty and workers' reliability, even if no ground truth is available to validate workers' judgments.

2.2 Just-in-Time Retrieval Systems and their Strategies for Query Formulation

The just-in-time information retrieval paradigm aims to reverse the traditional model of search for information, by offering users the possibility to spontaneously receive recommendations, based on their current activity, in a query-free manner. One of the first systems for document recommendation, referred to as query-free search, was the Fixit system (Hart and Graham, 1997), 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 of the interaction.

The Remembrance Agent (Rhodes and Starner, 1996; Rhodes and Maes, 2000), another early just-in-time retrieval system, is closer in concept to the system considered in this thesis. 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 (Rhodes, 1997). 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 information retrieval system (Budzik and Hammond, 2000) 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 (Czerwinski et al., 1999; Dumais et al., 2004) 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 (Henzinger et al., 2005). Similar to IQ or Watson, queries were constructed from the ASR transcripts using several variants of TFIDF weighting, and considering also the previous queries made by the system.

Other real-time assistants were conversational. They interacted with users to answer their explicit information needs or to provide recommendations based on their conversation. For instance, Ada and Grace[1] were twin virtual museum guides (Traum et al., 2012), which interact with visitors to answer their questions, suggest exhibits, or explain the technology that makes them work. Moreover, a collaborative tourist information retrieval system (Arif et al., 2014, 2012) interacted with tourists to provide travel information such as weather conditions, attractive sites, holidays, and transportation, in order to improve their travel plans. MindMeld[2], another conversation assistant agent, 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 the conversation obtained using an ASR system. As far as is known, the system employes the state-of-the-art methods for language analysis and information retrieval (Zaino, 2014).

In collaboration with other researchers, we have designed the Automatic Content Linking Device (ACLD) (Popescu-Belis et al., 2008, 2011) 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 monitors the users' conversation and prepares implicit queries from

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

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

words recognized through an ASR system. The current system models users' information needs as a set of keywords extracted at regular time intervals, by comparing the words transcribed by the ASR system against a list of keywords fixed before the meeting. We will show in Chapter 3 that this method outperforms the use of the entire set of words from a conversation fragment as an implicit query (Habibi and Popescu-Belis, 2012) . 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 (Yazdani, 2013, 4.12). We will provide a detailed explanation and analysis of this claim in Chapter 3. These findings motivated us to design innovative methods for modeling users' information needs from their conversation.

2.3 Keyword Extraction Methods

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 precision and user satisfaction in terms of the relevance of recommended document results.

Numerous methods have been proposed to automatically extract keywords from a text, and many are applicable also to the ASR transcript of a conversation fragment. The earliest techniques have used word frequencies (Luhn, 1957) and TFIDF values (Salton et al., 1975; Salton and Buckley, 1988) to rank words for extraction. Alternatively, words have been ranked by counting pairwise word co-occurrence frequencies (Matsuo and Ishizuka, 2004). These approaches have not considered 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 (Nenkova and McKeown, 2012).

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, one approach has used the frequency of all words belonging to the same WordNet concept set (Ye et al., 2007), while the Wikifier system (Csomai and Mihalcea, 2007) relied on Wikipedia links to compute a substitute to word frequency.

Hazen (2011a) applied latent topic modeling techniques to audio files. 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 (Hazen, 2011b). Harwath and Hazen (2012) 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. Furthermore, Harwath et al. (2013) 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. In addition, a semi-supervised latent concept classification algorithm was presented by Celikyilmaz and Hakkani-Tur (2011) using LDA topic modeling for multi-document information extraction.

Graph-based methods are also used for keyword extraction. For instance, word co-occurrence has been combined with PageRank (Mihalcea and Tarau, 2004), and additionally with WordNet (Wang et al., 2007a), or with topical information (Liu et al., 2010). However, as shown empirically by Mihalcea and Tarau (2004) and by Liu et al. (2010), such approaches have difficulties modeling long-range dependencies between words related to the same topic, which occur frequently in texts and conversations. Moreover, Liu et al. (2009a) have shown that graph-based approaches are not appropriate to extract keywords from conversational transcripts due to lack of well structure in these transcripts.

To consider dependencies among selected words, Riedhammer et al. (2008) 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. To reduce the effect of noise in meeting environments, this method removed all n-grams which appear only once or are included in longer n-grams with the same frequencies. In another study, part-of-speech information and word clustering techniques were used for keyword extraction (Liu et al., 2009b), while later this information was added to TFIDF so as to consider both word dependency and semantic information (Liu et al., 2009a).

In a recent paper, a keyword extraction technique from a set of documents was introduced by Jiang et al. (2015) 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 (Mikolov et al., 2013a,b). They first extracted a set of keywords from the title of all the documents as the initial keyword set and then added words from the abstract and the body of documents. Their method selected words which have higher similarity with the initial set using cosine similarity over their word2vec vector representations. 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, by giving too much importance to the first main topic only.

Supervised machine learning methods have been used to learn models for extracting keywords. This approach was first introduced by Turney (1999), who combined heuristic rules with a genetic algorithm. Other learning algorithms such as Naive Bayes (Frank et al., 1999), Bagging (Hulth, 2003), or Conditional Random Fields (Zhang et al., 2008) have been used to improve accuracy. These approaches, however, rely on the availability of in-domain training data (which is not the case in our setting), and the objective functions they use for learning still do not consider the diversity of keywords. Therefore, we propose a new keyword extraction method in Chapter 4 to maximize the coverage of the main topics discussed in the conversation as well as reduce the effect of the ASR noise within the keyword set.

2.4 Diverse Retrieval Merging and Re-ranking Techniques

Just-in-time retrieval systems, as presented above, have been designed to recommend documents which are potentially relevant to users' activities (Hart and Graham, 1997; Rhodes and Maes, 2000; Popescu-Belis et al., 2008). When using a just-in-time retrieval system, people are generally not willing to inspect a large number of recommended documents, mainly because this would distract them from their main activity. Several solutions to this problem have been proposed.

The Watson just-in-time information retrieval system (Budzik and Hammond, 2000), which is designed for reading or writing activities, clustered the retrieval results and selected from each cluster the best representative, so as to recommend only a short list of document results. However, clustering the results is not suitable for our application, because the mixture of topics in a single query will degrade the document results aimed to be clustered (Bhogal et al., 2007; Carpineto and Romano, 2012), and consequently may have a damaging effect on the clusters' representatives. Moreover, in Chapter 5, we will experimentally show that the list of documents suggested to users by merging the retrieval results of multiple topically-separated queries contains more relevant documents compared to that of a single query. The second part of the method proposed by Budzik and Hammond (2000), which selects the best representative of the clusters in the final document list, will be shown to be helpful in Chapter 5. We will use it there as a simple method to merge the lists of results retrieved for multiple queries; however, its effectiveness relies on having clusters with similar levels of importance (Wu and McClean, 2007).

Many studies in information retrieval have addressed the problem of diverse ranking, which can be stated as a tradeoff between finding relevant versus finding a diverse set of results (Robertson, 1997). The existing diverse ranking proposals differ in their diversifying policies and definitions, which can be categorized into implicit methods (Carbonell and Goldstein, 1998; Zhai et al., 2003; Radlinski and Dumais, 2006; Wang and Zhu, 2009) or explicit ones (Agrawal et al., 2009; Carterette and Chandar, 2009; Santos et al., 2010; Vargas et al., 2012).

The implicit approaches assume that similar documents will cover similar aspects of a query, and have to be demoted in the ranking to promote relative novelty and reduce overall redundancy. In one of the earliest approaches, Carbonell and Goldstein (1998) introduced Maximal Marginal Relevance (MMR) to re-rank documents based on a tradeoff between the relevance of document results and the relative novelty as a measure of diversity. MMR was used by Radlinski and Dumais (2006) to re-rank results from a query set which is generated for a user query and represents a variety of potential user intents.

Instead of implicitly accounting for the aspects covered by each document, another option is to explicitly model these aspects within the diversification approach. Agrawal et al. (2009) introduced a submodular objective function to minimize the probability of average user dissatisfaction by producing a set of diversified results that cover different interpretation of a query. For each query, aspects are represented by topics which are modeled using a taxonomy of information available through the Open Directory Project (ODP). Moreover, Carterette and Chandar (2009) represented query aspects as topics estimated from the top ranked documents. Alternatively, Santos et al. (2010) proposed another submodular objective function to maximize coverage and minimize redundancy with respect to query aspects, which were modeled using a keyword-based representation instead of a predefined taxonomy. In Chapter 6, Section 6.5.1, we will show experimentally that the use of diverse re-ranking methods cannot improve the retrieval results of a single implicit query made of the entire keyword set extracted from a conversation fragment.

In fact, our just-in-time retrieval system requires a diverse merging method applicable to the retrieval results of multiple implicit queries, rather than a diverse re-ranking technique intended for a single query (even possibly a multi-aspect one). Several studies have been previously carried out on merging lists of results in information retrieval, mostly for distributed information retrieval, where several lists of results from different search engines for the same query must be merged (Callan, 2000; Aslam and Montague, 2001; Wu and McClean, 2007). However, despite the superficial similarity, the problem here is in fact different, which is why a new approach will be proposed in Chapter 6 of this thesis.

2.5 Query Refinement Methods

Some just-in-time retrieval systems allow users to also express explicitly their information needs. Explicit queries can be ambiguous because the query words can refer to multiple notions. Several methods for the refinement of explicit queries asked by users have been proposed in the field of information retrieval, and are often classified into automatic query expansion techniques and relevance feedback ones (Carpineto and Romano, 2012).

Methods based on query expansion generate one or more hypotheses for query refinement by recognizing possible interpretations of a query, using knowledge coming either from a corpus (Attar and Fraenkel, 1977; Xu and Croft, 1996; Robertson et al., 1999; Carpineto et al., 2001; Bai et al., 2005) or from Web data or personal profiles in the case of Web search (Xu and Croft, 2000; Diaz and Metzler, 2006; Chirita et al., 2007; Park and Ramamohanarao, 2007). Query expansion techniques select suggestions for query refinement either interactively or automatically (Carpineto and Romano, 2012). For instance, relevance feedback gathers judgments obtained from users on sample results obtained from an initial query (Rocchio, 1971; Salton and Buckley, 1997; Lavrenko and Croft, 2001).

These methods are not ideal for the refinement of explicit queries asked during a conversation, because they require to interrupt users during their conversation for obtaining further clarifications. On the contrary, our overall goal (as for implicit queries) is to estimate users' information needs from their explicit queries as unobtrusively as possible. Moreover, using the local context for query refinement instead of external, non-contextual resources has the potential to improve retrieval results (Budzik and Hammond, 2000).

To the best of our knowledge, two previous systems have utilized the local context for the augmentation of explicit queries. The JIT-MobIR system for mobile devices (Alidin and Crestani, 2013) used contextual features from the physical and the human environment, although the content of the activities itself was not used as a feature. The Watson system already introduced above (Budzik and Hammond, 2000) refined explicit queries by concatenating them with the preceding implicit query, which is made of keywords extracted from the documents being edited or viewed by the user. However, in order to apply this 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, in contrast to written documents which generally follow a planned and well-focused structure, in a conversation users often shift from one topic to another. Such expansion terms from ASR noise and irrelevant topics might deteriorate the retrieval results of explicit queries (Jin et al., 2003; Bhogal et al., 2007; Carpineto and Romano, 2012). Therefore, we propose a query refinement technique which expands an explicit query by the keywords extracted from the ASR transcript of users' conversation fragment coming before the query and also is robust to ASR noise and off-topic keywords.

3 Comparative Evaluation Method Using Crowdsourcing

Evaluating the relevance of recommendations produced by the just-in-time retrieval system described in this thesis is a challenging task. The evaluation in use requires the full deployment of the system and the setup of numerous evaluation sessions with realistic meetings. That is why alternative solutions based on simulations are important to find. In this chapter, we propose to run the system over a corpus of conversations and to use crowdsourcing to compare the relevance of results in various configurations of the system. As this method is essential to quantify the advantages of our just-in-time retrieval methods, we have chosen to present it as the first research contribution of the thesis, before the actual work on keyword extraction, query formulation, and merging of result lists.

3.1 Introduction

A crowdsourcing platform, here Amazon's Mechanical Turk (AMT), is helpful to evaluate the relevance of documents recommended by a just-in-time retrieval system for several reasons. First, we can evaluate a large amount of results in a fast and inexpensive manner. Second, workers are sampled from the general public, and have no contact with each other, which might represent a more realistic user model than system developers or graduate students for instance. However, in order to use workers' judgments for relevance evaluation, we have to assess the quality of their evaluations, and to factor out the possible biases of individual contributions.

We formulate in this chapter an evaluation protocol using crowdsourcing, which assesses the quality of workers' judgments by estimating task difficulty and workers' reliability, even if no ground truth to validate the judgments is available. This approach, named Pearson Correlation Coefficient-Information Entropy (PCC-H), builds upon previous studies of inter-rater agreement and uses notions of information theory.

Moreover, we present in this chapter several experiments using the proposed evaluation method to several preliminary designs of the Automatic Content Linking Device (ACLD), a just-in-time retrieval system pre-dating this thesis, which lead to different lists of document recommendations. The results demonstrate that using the keywords extracted (using a dictionary-based method) from a fragment for query formulation provides more relevant list of documents compared to using all the words of the fragment. Besides, the findings presented here also confirm that the system's recommendations cannot appropriately answer users' explicit queries, thus justifying the need for the separate module specified in Chapter 7.

This chapter is organized as follows. Section 3.2 describes the different versions of the system which will be compared. Section 3.3 presents our design of the evaluation micro-tasks on the crowdsourcing platform, here Amazon's Mechanical Turk. In Section 3.4, the proposed PCC-H method for measuring the quality of judgments is defined. Section 3.5 presents the results of our evaluation experiments, which on the one hand validate the proposed method, and on the other hand indicate the comparative relevance of the different versions of the system.

3.2 Versions of the ACLD System Compared in the Study

As it is difficult to assess the utility of a just-in-time retrieval system designed for conversational environments from an absolute perspective, we aim instead at comparing preliminary versions of the ACLD system, in order to assess the improvement (or lack thereof) due to various designs. Here, we will compare four different approaches to the recommendation problem, with the ACLD system (presented in 2.2 above) simply aiming to find the closest documents to a given stretch of conversation.

The four compared versions of the ACLD are the following ones. Two standard versions (Popescu-Belis et al., 2008) differ by the filtering procedure used to construct implicit queries. One of them (noted *AW*) uses all the words (except stopwords) spoken by users during a specific period as an implicit query to retrieve related documents. The other one (noted *KW*) filters the words, keeping only keywords from a pre-defined list related to the topic of the meeting as implicit queries.

Two other methods depart from the standard versions. One of them implements semantic search (noted *SS*) as proposed by Yazdani (2013), which uses a graph-based semantic relatedness measure to perform retrieval (Popescu-Belis et al., 2011). The other one (noted *EQ*) allows users to ask their explicit queries and recognizes their addressing to the system by requiring them to start their explicit queries with a specific unambiguous word chosen as the system's name, such as "John".

In the evaluation experiments presented here, we only use human transcriptions of meetings, to focus on the meta-evaluation of the retrieval strategy itself. We use one meeting (ES2008b) from the AMI Meeting Corpus (Carletta, 2007) in which the design of a new remote control for a TV set is discussed. The explicit users' requests for the *EQ* version are simulated by modifying the transcript at 24 different locations where we believe that users are likely to ask explicit queries. We restrict the search to the Wikipedia pages, mainly because the semantic search system is adapted to this data, using a local copy of it obtained through the Freebase Wikipedia Extraction(WEX) dataset[1] from Metaweb Technologies (version dated 2009-06-16). Wikipedia is one of the most popular general reference works on the Internet, and recommendations over it are clearly of potential interest. But alternatively, all our versions except *SS* could also be executed with non-restricted web searches via the Google API, or could be limited to other web domains or websites.

The 24 fragments of the meeting containing the implicit and explicit queries are submitted for comparison. That is, we want to know which of the results displayed by the various versions at the moment following the implicit or explicit query are considered most relevant by external judges. As the method allows only binary comparisons, we will compare *EQ* with the *AW* and *KW* versions, and then *SS* with *KW*.

3.3 Designing the Comparative Evaluation Tasks

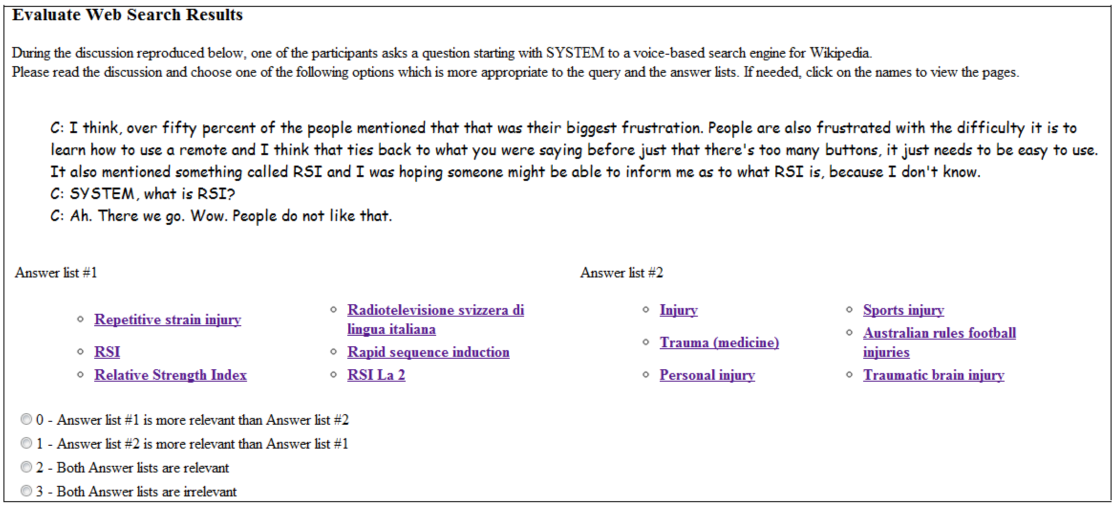
Amazon's Mechanical Turk (AMT) is a crowdsourcing platform which gives access to a large pool of online workers paid by requesters to complete short "Human Intelligence Tasks" (HITs). Once the tasks are designed and published, registered workers that fulfill the requesters' selection criteria are invited by the AMT service to work on HITs in exchange for a small amount of money per HIT (Alonso and Lease, 2011).

Since it is difficult to find an absolute relevance score, we only aim for comparative relevance evaluation between versions. For each pair of versions, a batch of HITs is designed with their results. Each HIT, as exemplified in Figure 3.1, contains the transcript of a conversation fragment with the two lists of document results to be compared. Only the first six document results are made visible for each version. The lists from the two compared versions are placed in random positions (first or second, i.e. left or right) across HITs, to avoid biases from a constant position.

We experiment with two different HIT designs. The first design offers evaluators a binary choice: either the first list is considered more relevant than the second one, or vice-versa. In other words, workers are obliged to express a preference for one of the two recommendation lists. This encourages decision making, but of course may be inappropriate when the two answers are of comparable quality, though this may be evened out when averaging over workers. The second design gives workers four choices, as in Figure 3.1: in addition to the previous two options, workers can indicate either that both lists appear to be equally relevant, or equally irrelevant. In both designs, workers must select exactly one option. Then, the relevance value of each recommendation list is computed using the PCC-H score defined in the next section.

There are 24 meeting fragments, hence 24 HITs in each batch for comparing pairs of systems, for *EQ* vs. *AW* and *EQ* vs. *KW*. However, the results obtained by the team using semantic

[1] See http://download.freebase.com/wex.



Evaluate Web Search Results

During the discussion reproduced below, one of the participants asks a question starting with SYSTEM to a voice-based search engine for Wikipedia.
Please read the discussion and choose one of the following options which is more appropriate to the query and the answer lists. If needed, click on the names to view the pages.

C: I think, over fifty percent of the people mentioned that that was their biggest frustration. People are also frustrated with the difficulty it is to learn how to use a remote and I think that ties back to what you were saying before just that there's too many buttons, it just needs to be easy to use. It also mentioned something called RSI and I was hoping someone might be able to inform me as to what RSI is, because I don't know.
C: SYSTEM, what is RSI?
C: Ah. There we go. Wow. People do not like that.

Answer list #1

- Repetitive strain injury
- RSI
- Relative Strength Index
- Radiotelevisione svizzera di lingua italiana
- Rapid sequence induction
- RSI La 2

Answer list #2

- Injury
- Trauma (medicine)
- Personal injury
- Sports injury
- Australian rules football injuries
- Traumatic brain injury

○ 0 - Answer list #1 is more relevant than Answer list #2
○ 1 - Answer list #2 is more relevant than Answer list #1
○ 2 - Both Answer lists are relevant
○ 3 - Both Answer lists are irrelevant

Figure 3.1: Illustration of a four-choice HIT: workers read the conversation transcript, examine the two document lists (with recommended documents for the respective conversation fragment from two different systems) and select one of the four comparative options (#1 better than #2, #2 better than #1, both equally good, both equally poor). They also can add a short comment at the end.

search were based on a different segmentation: hence, to compare *SS* vs. *KW*, we obtained 36 HITs. There are 10 workers per HIT, so there are 240 total assignments for *EQ* vs. *KW* and for *EQ* vs. *AW* (with a two-choice and four-choice design for each), and 360 for *SS* vs. *KW*. As workers are paid 0.02 USD per HIT, the cost for the five separate experiments is 33 USD, with an apparent average hourly rate of 1.60 USD. The average observed time per assignment is almost 50 seconds. All five tasks took only 17 hours to be fully performed by workers via AMT. For the qualification control procedure, we accept workers with greater than 95% approval rate (i.e. 95% of the worker's submitted HITs for previous tasks have been approved by their requesters) and with more than 1000 approved HITs.

3.4 The PCC-H Score

Majority voting is frequently used to aggregate multiple sources of comparative relevance evaluation. However, this approach is not directly appropriate to compute the comparative relevance scores from the judgements obtained via crowdsourcing (Carletta, 1996; Smyth et al., 1994; Chittaranjan et al., 2011; Karger et al., 2011; Whitehill et al., 2009; Khattak and Salleb-Aouissi, 2011), because it assumes that all HITs share the same difficulty and all the workers are equally reliable. We will address this issue by introducing a new computation method based on weighted averages of the judgments, considering both the task difficulty and the workers' reliability. The method is called PCC-H, for Pearson Correlation Coefficient-Information Entropy.

3.4.1 Estimating Worker Reliability

First, we apply a qualification control factor to the human judgments noted c_v. The factor reduces the impact of workers who disagree with the majority, using the Pearson correlation of one worker's judgment with the average of all others as computed in Equation 3.1.

$$c_v = \frac{1}{|A|}\sum_{a=1}^{A}\frac{\sum_{t=1}^{|T|}(S_{tv}(a)-\mu_{S_{tv}(a)})(S'_{tv}(a)-\mu_{S'_{tv}(a)})}{(T-1)\sigma_{S_{tv}(a)}\sigma_{S'_{tv}(a)}} \tag{3.1}$$

In this equation, $|T|$ is the number of meeting fragments and $S_{tv}(a)$ is the value that the worker v assigned to the option a (among the two or four possible answers) for the fragment t (i.e. $S_{tv}(a)$ is 1 if option a is selected by worker v, otherwise it is 0). $\mu_{S_{tv}(a)}$ and $\sigma_{S_{tv}(a)}$ are the expected value and the standard deviation of the variable $S_{tv}(a)$ over the entire fragment, respectively. $S'_{tv}(a)$ is the average value that all other workers (except v) assign to the option a of the fragment t. $\mu_{S'_{tv}(a)}$ and $\sigma_{S'_{tv}(a)}$ are the expected value and the standard deviation of the variable $S'_{tv}(a)$ as well.

The value of c_v computed above is used as a weight for computing $CS_t(a)$, which is the relevance value of the document list corresponding to option a of the fragment t, according to Equation 3.2. Note that the values of c_v is in the range of -1 to 1. Although the negative values are appeared, they are few (usually subjects follow each other and there are a small number of them which perform malicious behaviours), and we still use them as weights in our experiments. However, in particular (in all our experiments), they are positive.

$$CS_t(a) = \left(\sum_{v=1}^{V} c_v S_{tv}(a)\right) / \left(\sum_{v=1}^{V} c_v\right) \tag{3.2}$$

3.4.2 Estimating Task Difficulty

In addition to the workers' reliability computed above, the PCC-H method considers the task difficulty for each fragment of the meeting, noted D_t. The goal is to reduce the effect on the final score of those fragments of the meeting in which there is an uncertainty among the workers' judgments.

To reduce the effect of uncertainty in our judgments, we factor out the impact of undecided fragments by using a measure of the entropy of the answer distribution for each fragment t. The entropy of answers for each fragment of the meeting is computed in Equation 3.3. The task difficulty is then calculated as a function of the entropy as written in Equation 3.4, and is used as a weight for each fragment t in Equation 3.5.

$$H_t = -\frac{\sum_{a=1}^{A} CS_t(a)\ln(CS_t(a))}{\ln(|A|)} \tag{3.3}$$

$$D_t = 1 - H_t \tag{3.4}$$

3.4.3 Comparative Relevance Score

We computed the comparative relevance scores for HIT designs with two options and four options. For the two-option HIT designs, a final comparative score, noted $CS(a)$, is computed for each set of judgments. To compare two competitive methods, we average the weighted scores over all the fragments, as shown in Equation 3.5. When comparing two sets of answers, the sum of the two scores is always 100%.

$$\%CS(a) = \frac{\sum_{t=1}^{|T|} D_t\, CS_t(a)}{\sum_{t=1}^{|T|} D_t} \times 100 \tag{3.5}$$

For four-option HIT designs, we provided two different approaches. The first approach computes the average comparative score for each list and task, $CS_t(l)$, which is called the global relevance value for the answer list l and is formulated as Equation 3.6 below:

$$CS_t(l) = CS_t(a_l) + \frac{CS_t(a_b)}{2} - \frac{CS_t(a_n)}{2} \tag{3.6}$$

This can be weighted by task difficulty as formulated in Equation 3.5. In this equation, half of the relevance value of the case in which both lists are relevant $CS(a_b)$ is added as a reward (a_b denotes the answer “both are relevant”), and half of the relevance value of the case in which both lists are irrelevant $CS(a_n)$ is subtracted as a penalty (a_n denotes the answer “none is relevant”) from the relevance value of each answer list $CS(a_l)$ (a_l denotes the answer “answer list l is relevant”). We normalize the values to keep the sum of scores equal to 100%.

In the second approach, the relevance value for an answer list is defined as the probability of being equally relevant or more relevant compared to the other answer list. Assuming that the probability of workers’ judgments for each task has a normal distribution, we can estimate the probability of selecting each option by the workers using maximum likelihood estimation. Therefore, the probability of answer list l_1 being equally relevant or more relevant than answer list l_2 for task t is computed as:

$$CS_t(a_{l_1} \geq a_{l_2}) = CS_t(a_{l_1}) + CS_t(a_b) \tag{3.7}$$

The corresponding value for answer list l_2 is similar to the above formula. Then, the score of answer list l_1 for all the tasks is defined in Equation 3.8 and can be computed similarly for answer list l_2 as well.

$$CS(l_1) = \left(\sum_{t=1}^{|T|} D_t\, CS_t(l_1)\right) / \left(\sum_{t=1}^{|T|} D_t\right) \tag{3.8}$$

This formulation allows us to calculate the confidence interval for each comparative relevance value. Although we do not have the real mean and variance of workers judgments, we can compute the confidence interval using weighted Student's t-distribution. If the comparison scores of two answer lists do not overlap for $p = 0.05$, then the two methods generated the two answer lists are distinguishable, and we can infer that the difference between the results is statistically significant. The confidence interval for each answer list retrieved for the tasks with different task difficulty values is calculated as follows:

$$CI(l_1) = t_{(1-\frac{p}{2},|T|-1)} \sqrt{S^2_{CS}(l_1) / \sum_{t=1}^{|T|} D_t} \tag{3.9}$$

where $S^2_{CS}(l_1)$ is the weighted variance of the probability score computed for the answer list l_1 based on the task difficulty values, as shown below in Equation 3.10, and $t_{(1-\frac{p}{2},|T|-1)}$ is obtained from the t-table of Student's t-distribution.

$$S^2_{CS}(l_1) = \left(\sum_{t=1}^{|T|} D_t (CS_t(l_1) - CS(l_1))^2 \right) / \left(\frac{(|T|-1) \sum_{t-1}^{|T|} D_t}{|T|} \right) \tag{3.10}$$

In this chapter we only present the scores using the first approach for four-option HIT designs, but in all the following chapters we present the relevance scores using both approaches and assess the statistical significance of the improvements brought by our methods using the method presented above.

3.5 Results of the Experiments

Two sets of experiments will be described. First, we attempt to validate the PCC-H method. Then, we apply the PCC-H method to perform a binary comparison between the four preliminary versions of the system presented above. Here, we will first show that the results of implicit queries cannot appropriately cover the answers to explicit queries, and confirm the need for adding an extra module that supports to properly answer users explicit queries. Second, we will validate our motivations for proposing a novel keyword extraction method by demonstrating that using as an implicit query the set of keywords extracted by a baseline method, instead of all the words of a conversation fragment, already improves the recommendation results.

3.5.1 Validation of the Worker Reliability Values

To provide an initial validation of the workers' judgments, we compare the judgments of individual workers with those of an expert. For each worker, the number of fragments for which the answer is the same as the expert's answer is counted, and the total is divided by the number of fragments, to compute accuracy. Then we compare this value with c_v, which is estimated as the reliability measurement for each worker's judgment. The number of agreements between each worker vs. the expert over all HITs e_v and the value of c_v for each worker for one of the batches of HITs is shown in Table 3.1. The numbers indicate an overall agreement between these two values for each worker, with a Pearson correlation coefficient of 0.89 over all workers (strong correlation). In other words, workers who have more similarity with our expert also have more inter-rater agreement with other workers. Since in the general case there is no ground truth (expert) to verify workers judgments, we will rely on the inter-rater agreement values for the other experiments.

Table 3.1: The number of agreements between a single worker and the expert over all HITs, and a single worker and the other workers, provided by 10 workers for the *KW* system and four-option HIT design. There is an agreement between these values among workers.

worker #	1	2	3	4	5	6	7	8	9	10
e_v	0.66	0.54	0.54	0.50	0.50	0.50	0.41	0.39	0.36	0.31
c_v	0.81	0.65	0.64	0.71	0.60	0.35	0.24	0.33	0.34	0.12

3.5.2 Validation of the PCC-H Method

In order to show that our method is stable on different HIT designs, we use two different HIT designs for each pair as mentioned in Section 3.3. Firstly, we assume that all the workers are reliable and all the fragments share the same difficulty by assigning equal weights to all the user evaluations and fragments (majority voting) to compute comparative relevance scores for two answer lists of our experiments, which are shown in Table 3.2.

Table 3.2: Comparative relevance scores computed by the assumption that all workers are equally reliable and all tasks are equally difficult. The compared methods are *AW* vs. *EQ* and *KW* vs. *EQ*. The scores are provided for both two-option and four-option HIT designs.

Compared methods (m_1 vs. m_2)	**Relevance (%)**			
	two-option HIT design		**four-option HIT design**	
	m_1	m_2	m_1	m_2
AW vs. *EQ*	30	**70**	26	**74**
KW vs. *EQ*	45	**55**	35	**65**

In this Table, the comparative relevance scores of *KW* vs. *EQ* differ between the two types of HIT design. To overcome this issue, we considered the workers reliability factor. One approach is to consider the workers with low c_v values as outliers, and remove all outliers. For instance, the four workers with lowest c_v, shown in Table 3.1, are considered outliers and are deleted, and then equal weights are given to the remaining six workers. For now, it is still assumed that all the tasks have the same level of difficulty by assigning equal weights to them. The results of comparative evaluation based on removing outliers are shown in Table 3.3.

However, there is no convergence of comparative relevance scores in the different HIT designs. To make these values closer, instead of deleting workers with lower c_v, which might still have potentially useful insights on relevance, it is rational to give a weight to all workers' judgments based on this value as a confidence value. The comparative relevance score for each answer

Table 3.3: Comparative relevance scores computed after removing the judgments of outlier workers with low reliability factor. The method assigns equal reliability to the judgments of the remaining workers. It is also assumed that tasks have the same level of difficulty and are given equal weights. The compared methods are *AW* vs. *EQ* and *KW* vs. *EQ*. The scores are provided for both two-option and four-option HIT designs.

Compared methods (m_1 vs. m_2)	**Relevance (%)**			
	two-option HIT design		**four-option HIT design**	
	m_1	m_2	m_1	m_2
AW vs. *EQ*	24	**76**	13	**86**
KW vs. *EQ*	46	**54**	33	**67**

list of four experiments based on assigning weight c_v to each worker's evaluation, and equal weights to all meeting fragments are shown in Table 3.4.

Table 3.4: Comparative relevance scores computed by considering the workers reliability weights. However, the fragments are still given equal weights with the assumption that all the tasks have the same level of difficulty. The compared methods are *AW* vs. *EQ* and *KW* vs. *EQ*. The scores are provided for both two-option and four-option HIT designs.

Compared methods (m_1 vs. m_2)	**Relevance (%)**			
	two-option HIT design		**four-option HIT design**	
	m_1	m_2	m_1	m_2
AW vs. *EQ*	24	**76**	18	**82**
KW vs. *EQ*	33	**67**	34	**66**

Although the comparative relevance scores are very close under different HIT designs by considering only worker's reliablity factors for *AW* vs. *EQ* comparison, we show that these values approximately converge to the same value for each pair with different HIT designs. As observed in Table 3.4, the comparative relevance scores of *AW* vs. *EQ* are not quite similar for two different HIT designs, although the answer lists are the same. In fact, we observed that, in several cases, there is no strong agreement among workers to decide which answer list is more relevant to that meeting fragment, and we consider that these are "difficult" fragments. Since the source of uncertainty is undefined, we can reduce the effect of that fragment on the comparison by giving a weight to each fragment in proportion of the difficulty of assigning $CS_t(a)$. The comparative relevance values thus obtained for all experiments are represented in Table 3.5. As shown in this table, the comparative relevance values of *AW* vs. *EQ* are now very similar for two-option and four-option HIT designs. Moreover, the difference between the system versions is emphasized, which indicates that the sensitivity of the comparison method has increased.

Moreover, we compare the PCC-H method with the majority voting method and the GLAD method (Generative model of Labels, Abilities, and Difficulties (Whitehill et al., 2009)) for estimating comparative relevance values by considering task difficulty and worker reliability parameters. We run the GLAD algorithm with the same initial values for all four experiments.

Table 3.5: Comparative relevance scores computed by considering both the workers reliability and task difficulty weights, noted as the PCC-H score. The compared methods are *AW* vs. *EQ* and *KW* vs. *EQ*. The scores are provided for both two-option and four-option HIT designs.

Compared methods (m_1 vs. m_2)	**Relevance (%)**			
	two-option HIT design		**four-option HIT design**	
	m_1	m_2	m_1	m_2
AW vs. *EQ*	19	**81**	15	**85**
KW vs. *EQ*	23	**77**	26	**74**

The comparative relevance scores which are computed by the majority voting, the PCC-H and the GLAD methods are shown in Table 3.6.

Table 3.6: Comparative relevance scores computed by majority voting, PCC-H, and the GLAD methods. The compared methods are *AW* vs. *EQ* and *KW* vs. *EQ*. The scores are provided for both two-option and four-option HIT designs.

Compared methods (m_1 vs. m_2)		**Majority voting, PCC-H, GLAD**	
		two-option HIT design	**four-option HIT design**
AW vs. *EQ*	m_1	30%, 19%, 23%	26%, 15%, 13%
	m_2	70%, **81%**, 77%	74%, **85%**, 87%
KW vs. *EQ*	m_1	45%, 23%, 47%	35%, 26%, 23%
	m_2	55%, **77%**, 53%	65%, **74%**, 77%

As shown in Table 3.6, the comparative relevance scores which are computed by the PCC-H method for both HIT designs are very close to those of GLAD for the four-option HIT design. Moreover, the comparative relevance values obtained by the PCC-H method for the two different HIT designs are very similar, which is less the case for majority voting and GLAD. This means that PCC-H is able to calculate the comparative relevance scores independent of the exact HIT design. Moreover, the comparative relevance values calculated using PCC-H are more robust since the proposed method is not dependent on initialization values, as GLAD is. Therefore, using PCC-H for measuring the reliability of workers judgments is also an appropriate method for qualification control of workers from crowdsourcing platforms.

3.5.3 Comparison across Various Versions of the System

As shown in Table 3.6, regardless of the type of the qualification control technique which is used for the comparative evaluation of *AW* vs. *EQ* and *KW* vs. *EQ*, the *EQ* version outperforms the *KW* version and more considerably the *AW* one.

The proposed method is also applied for the comparative evaluation of *SS* vs. *KW* search results (semantic search vs. keyword-based search). The comparative scores are calculated by three different methods as shown in Table 3.7. The first method is the majority voting method which considers all the workers and fragments with the same weight. The second method is

Table 3.7: Comparative relevance scores computed by majority voting, PCC-H, and GLAD. The compared method is *SS* vs. *KW*. The scores are provided for both two-option and four-option HIT designs.

Compared methods		**Majority voting, PCC-H, GLAD**
(m_1 vs. m_2)		**four-option HIT design**
SS vs. *KW*	m_1	88%, **93%**, 88%
	m_2	12%, 7%, 12%

PCC-H and the third one is the GLAD method. It appears that the *SS* version outperforms the *KW* version according to all three scores.

3.6 Conclusion

In all the evaluation steps, the *EQ* system appeared to produce more relevant recommendations than *AW* or *KW*. This means that using *EQ*, i.e. when users ask explicit queries in conversation, improves over the *AW* or *KW* versions, i.e. with spontaneous recommendations. Using *KW* instead of *AW* improved the scores by 10 percent, which indicates the superiority of keywords over the entire fragment words for formulation of implicit queries.

KW can be used as an assistant which suggests documents based on the context of the meeting along with the *EQ* version. Moreover, the *SS* version works better than the *KW* version, which shows the advantage of semantic search. However, the high computation cost of *SS* version makes it untractable for the just-in-time recommendation.

As for the evaluation method, PCC-H outperformed the GLAD method proposed earlier for estimating task difficulty and reliability of workers in the absence of ground truth. Based on the evaluation results, the PCC-H method is acceptable for qualification control of AMT workers or judgments, because it provides a more stable comparative score across different HIT designs. Moreover, PCC-H does not require any initialization.

The comparative nature of PCC-H imposes some restrictions on the evaluations that can be carried out. For instance, if N versions must be compared, this calls in theory for $N*(N-1)/2$ comparisons, which is clearly impractical when N grows. This can be solved if initial hypotheses about the quality of the systems are available, to avoid redundant comparisons. Moreover, an existing approach to reduce the number of pairwise comparisons which are required from human raters (Llorà et al., 2005) could be ported to our context if needed. For progress evaluation, a new version must be compared with the best performing previous version, looking for measurable improvement, in which case PCC-H fully answers the evaluation needs.

As seen in the scores above, there are many instances in which the search results of both versions are irrelevant. In the next chapters we will improve the quality of recommendations by defining new approaches for the formulation of implicit queries or re-ranking of the retrieved results. We also will deal with ambiguous explicit queries by expanding them using the context of the conversation.

4 Diverse Keyword Extraction from Conversational Transcripts

In this chapter, we propose a diverse keyword extraction method which is applicable to conversational fragments that are transcribed by a real-time ASR system. The two main aims with respect to existing methods are: (a) to maximize the number of conversation topics that are covered by the extracted keywords; and (b) to minimize the number of words produced by the ASR noise that are selected as keywords. For this reason, our method rewards at the same time the relevance of the keywords and the diversity of topics they cover. The method is compared with baselines in terms of extracting keywords which better represent the content of a conversation fragment using a crowdsourcing platform to collect a large number of comparative judgments for sets of keywords, as presented in Chapter 3. Specifically, the method is evaluated on fragments from the manual transcripts of the Fisher and the AMI conversational corpora. Besides, we evaluate the method on the ASR transcripts of the AMI corpus to measure the noise reduction power of the proposed method compared to the baselines. The results demonstrate that our method outperforms two other methods, a classical one based on word frequency, and a more recent one considering topics but not enforcing diversity.

To demonstrate its versatility, our method is also applied to the ASR transcripts of video lectures, within Idiap's MUST-VIS system (Multi-factor Segmentation for Topic Visualization and Recommendation). In this application, the keywords are used for the visualization of the content of video lectures or segments, as well as for the computation of the similarity value among them in order to suggest other relevant segments or lectures based on the one that is currently viewed by a user.

4.1 Introduction

The goal of keyword extraction from texts is to provide a set of words that are the best representative of the semantic content of the conversational fragments. For instance, in the example discussed in Section 4.4.5 below, in which four people are discussing about the impact on sales of some features of remote controls (a fragment of 110 seconds including about 380 words), a variety of topics are mentioned, such as “remote control”, “losing a remote control”, “buying a remote control”, and “different suitable colors for remote controls”. What would then be the most representative keyword list to the fragment?

Given the potential multiplicity of topics and the potential ASR errors, our goal is to maintain multiple hypotheses about users’ information needs. Therefore, we aim at extracting a relevant and diverse set of keywords. The diversity of keywords increases the chance of extracting better representative keywords by maximizing the coverage of the main topics conveyed in the conversation. However, current keyword extraction methods are based on word or topic frequencies and do not consider the diversity of topics that may appear even in a short conversation fragment. For example, while a method based on topic similarity but not enforcing the diversity of topics represents the topics pertaining to “electronic devices”, “buying a device” or “different colors of a device” , it also misses other topics such as “remote control”, and “losing a remote control” which are also representative of the fragment. On the contrary, our novel keyword extraction technique maximizes the coverage of topics and reduces the number of irrelevant words by rewarding at the same time topical similarity and topical diversity.

This chapter is organized as follows. In Section 4.2 we describe the novel topic-aware diverse keyword extraction algorithm intended for just-in-time retrieval systems. Section 4.3 presents the data and the definition of the evaluation protocol as a set of micro-tasks (comparing sets of keywords), which are crowdsourced using Amazon’s Mechanical Turk. In Section 4.4, we provide and discuss the comparative results of two existing methods and our own one, which outperforms both of them, in terms of diversity and overall representativeness of the conversation over both the manual and the ASR transcripts of two conversational corpora. We also exemplify the results on one conversation fragment given in the Figure 4.7 of the thesis. Finally, in Section 4.5, we illustrate another application of the method, to the ASR transcripts of the video lectures instead of those of conversations, within the MUST-VIS multimedia recommender system.

4.2 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, following an approach that was inspired to us by recent summarization methods (Lin and Bilmes, 2011; Li et al., 2012). 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, which leads to the selection of a smaller number of noisy keywords (compared to the algorithms which ignore diversity) when words which are in reality ASR noise are associated to important topics in a fragment.

The proposed method for diverse keyword extraction proceeds in three steps, represented schematically in Figure 4.1. First, a topic model is used to represent the distribution of the abstract topic z for each word w, noted $p(z|w)$, as shown in Figure 4.1. The abstract topics are not pre-defined manually but are represented by latent variables using a generative topic modeling technique. These topics are inferred from 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, noted β_z, as shown in the second step of Figure 4.1. Finally, the keyword list $C = \{c_1, ..., c_k\}$ which covers a maximum number of the most important topics is selected by rewarding relevance and diversity, using an original algorithm introduced in this section.

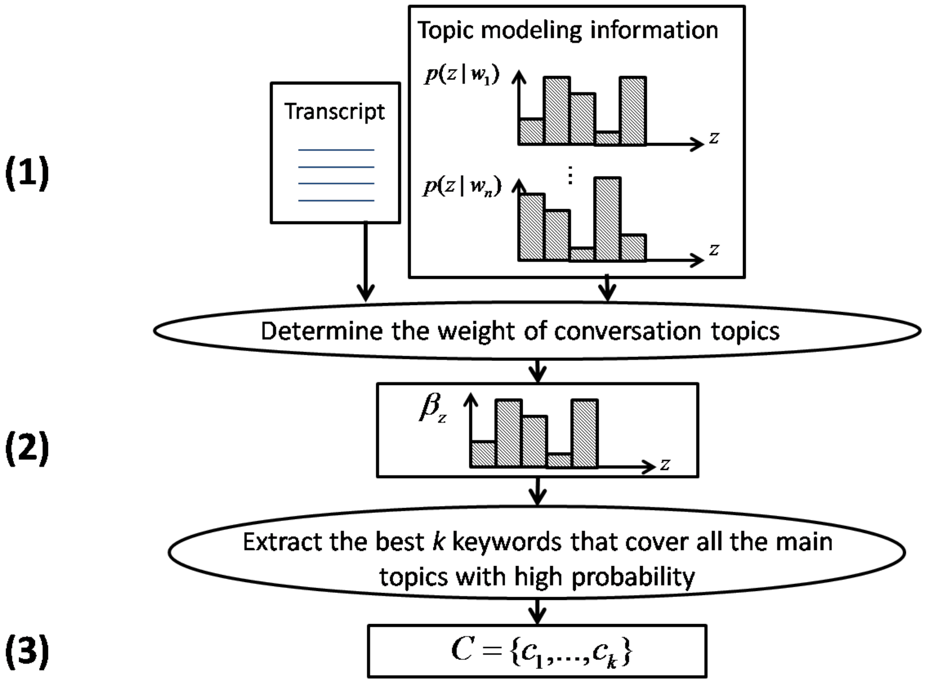

(1)
Transcript
Topic modeling information
$p(z|w_1)$
z
$p(z|w_n)$
z
Determine the weight of conversation topics
(2)
β_z
z
Extract the best k keywords that cover all the main topics with high probability
(3)
$C = \{c_1,...,c_k\}$

Figure 4.1: 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.

4.2.1 Modeling Topics in Conversations

Topic models such as Probabilistic Latent Semantic Analysis (PLSA) (Hofmann, 1999) or Latent Dirichlet Allocation (LDA) (Blei et al., 2003) can be used to determine the distribution over the topic $z \in Z$ of each word w, noted $p(z|w)$, from a large amount of training documents. LDA implemented in the Mallet toolkit (McCallum, 2002) is used here because it does not suffer from the overfitting issue of PLSA, as discussed by Blei et al. (2003). The smoothing parameters of the model are initially set to $\alpha = 50/|Z|$ and $\beta = 0.01$ as these values work well with different corpora (Steyvers and Griffiths, 2007), and they are then updated using the optimization algorithm of the Mallet toolkit.

When a conversation fragment is considered for keyword extraction, its topics are weighted by the β_z values (Equation 4.1 below), which are obtained by averaging over all probabilities $p(z|w)$ of the N words w spoken in the fragment t. The value of $p(z|w)$ is computed as in Equation 4.2.

$$\beta_z = \frac{1}{N}\sum_{w \in t} p(z|w) \tag{4.1}$$

$$p(z|w) = \frac{p(w|z)p(z)}{p(w)} \tag{4.2}$$

In the above equations, $p(w|z) = n_{w,z}/n_z$ and $p(z) = n_z/\sum_{z\in Z} n_z$, where $n_{w,z}$ is the number of times the word w is assigned to topic z, and n_z is the number of times all the words assigned to topic z, which can be computed using the output of the Mallet toolkit. Finally, $p(w)$ is computed as follows:

$$p(w) = \sum_{z\in Z} p(w|z)p(z) \tag{4.3}$$

4.2.2 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 is related to a subset of the topics in Z, then the goal is to find a subset C of k unique words ($C \subseteq t$ and $|C| = k$) 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, 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 (Nemhauser et al., 1978). A function F is submodular if $\forall A \subseteq B \subseteq U \setminus u$, $F(A+u) - F(A) \geq F(B+u) - F(B)$ (diminishing returns) and is monotone nondecreasing if $\forall A \subseteq B$, $F(A) \leq F(B)$.

To achieve our goal, we define the relationship of a topic z with respect to each set of words $C \subseteq t$ 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 C and topic z, to model the contribution of the set C to the topic z. Finally, we select one of the sets $C \subseteq t$ which maximizes the cumulative reward values over all the topics. This procedure is formalized below.

4.2.3 Definition of a Diverse Reward Function

We introduce $r_{C,z}$, the contribution towards topic z of the keyword set C selected from the fragment t:

$$r_{C,z} = \sum_{w \in C} p(z|w) \tag{4.4}$$

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 and λ is a parameter between 0 and 1. This is a submodular function with diminishing returns when $r_{C,z}$ increases, as proved in the next section.

$$f : r_{C,z} \to \beta_z \cdot r_{C,z}^{\lambda} \tag{4.5}$$

Finally, the keyword set $C \subseteq t$ is chosen by maximizing a cumulative reward function $R(C)$ over all the topics, formulated as follows:

$$R(C) = \sum_{z \in Z} \beta_z \cdot r_{C,z}^{\lambda} \tag{4.6}$$

Following this definition, 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. 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.

4.2.4 Proof of the Submodularity of the Reward Function

We will first show that $f(r_{C,z})$ is monotone nondecreasing and then shown that it is submodular (diminishing returns property). These properties are illustrated in Figure 4.2 for various values of λ, and are proven as follows.

To show that $f(r_{C,z})$ 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 $D = B \setminus A$, then:

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

If we substitute $\beta_z \cdot (r_{A,z} + r_{D,z})^{\lambda}$ with its binomial expansion by an infinite series (because λ is

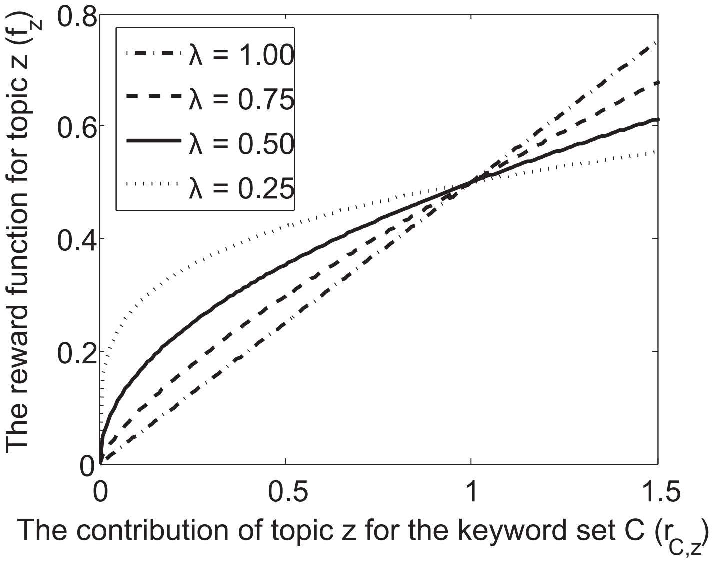


Figure 4.2: The reward value given to the topic z is decreasing when the contribution of this topic is increasing for various λ values (1, 0.75, 0.5, and 0.25) and $\beta_z = 0.5$.

not an integer), then:

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

Since $r_{A,z}$ and $r_{D,z}$ are obtained by summing over positive probability values, $\sum_{k=1}^{\infty} \binom{\lambda}{k} r_{A,z}^{\lambda-k} r_{D,z}^{k}$ 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 $f(r_{C,z})$ 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.

$$f(r_{A',z}) - f(r_{A,z}) = \beta_z \cdot r_{A',z}^{\lambda} - \beta_z \cdot r_{A,z}^{\lambda} = \beta_z \cdot (r_{A,z} + p(z|w))^{\lambda} - \beta_z \cdot r_{A,z}^{\lambda}$$

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

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

Similarly, we can establish that

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

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 $\beta_z \cdot \sum_{k=1}^{\infty} \left(\begin{array}{c} \lambda \\ k \end{array} \right) r^{\lambda-k}_{B,z} p(z|w)^k \leq \beta_z \cdot \sum_{k=1}^{\infty} \left(\begin{array}{c} \lambda \\ k \end{array} \right) r^{\lambda-k}_{A,z} p(z|w)^k$, which concludes the proof for diminishing returns.

Since it is monotone nondecreasing and has diminishing returns, $f(r_{C,z})$ is a monotone submodular function. Moreover, since the class of submodular functions is closed under non-negative linear combinations (Nemhauser et al., 1978), $R(C)$ is also a monotone non-decreasing submodular function.

According to a different perspective, the definition of $R(C)$ in Equation 4.6 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 $r^{\lambda}_{C,z}$ with a scaling exponent λ and identical coefficients over all topics. However, despite what this apparent similarity suggests, the use of cosine similarity for $R(C)$ 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 $R(C)$, for cosine similarity, then we can show that $R(C)$ is no longer monotone submodular, e.g. using the second example in the following subsection.

4.2.5 Examples for the Diverse Reward Function

We will illustrate the motivation for our definition of $R(C)$ 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 $C_1 = \{w_1, w_2\}$ and $C_2 = \{w_1, w_3\}$ the cumulative rewards are respectively $R(C_1) = 0.5 \cdot (1+1)^{\lambda} + 0.5 \cdot 0^{\lambda}$ and $R(C_2) = 0.5 \cdot 1^{\lambda} + 0.5 \cdot 1^{\lambda}$. Since $R(C_1) \leq R(C_2)$ for $0 < \lambda < 1$, the keyword set C_2 which covers two main topics is selected. If $\lambda = 1$ then the cumulative reward for the two sets C_1 and C_2 is equal, which does not guarantee to select the set which covers both topics.

The example above has the desirable values of $R(C)$ regardless of whether the dot product or the cosine similarity (discussed at the end of the previous section) are used for the definition of $R(C)$ in Equation 4.6. However, this is not always the case. In the example shown in Table 4.1 on page 41 (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 a cosine-based definition 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 $R(C)$ would neither have the diminishing returns property, if cosine similarity was used to define it.

4.2.6 Comparison with the Function Used for Summarization

In our definition of a monotone submodular function for keyword extraction, we have been inspired by recent work on extractive summarization methods (Lin and Bilmes, 2011; Li et al., 2012). 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. 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 considerably generalize over the previous function (square root) and the constant coefficient (1) for all concepts.

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 Lin and Bilmes (2011), and Li et al. (2012) 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.

4.2.7 Finding the Optimal Keyword Set

To maximize $R(C)$ in polynomial time under the cardinality constraint of $|C| = k$ we present a greedy algorithm shown as Algorithm 1. In the first step of the algorithm, C is empty. At each step, the algorithm selects one of the unselected words from the conversation fragment $w \in t \setminus C$ 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 C. The coverage is defined as $h(w, C) = \sum_{z \in Z} \beta_z [p(z|w) + r_{C,z}]^{\lambda}$, where $p(z|w)$ is the contribution to topic z by word $w \in t \setminus C$ which is added to the contribution of the topic z in the set C. The algorithm updates the set C by adding one of the words $w \in t \setminus C$ to the set C which maximizes $h(w, C)$. This procedure continues until reaching k keywords from the fragment t.

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

Algorithm 1: Algorithm for diverse keyword extraction.

4.2.8 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 4.1. 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 two different values of λ. For $\lambda = 1$, the algorithm selects words based on their topical similarity to the main topics of the conversation, and for $\lambda = .75$ it considers both topical diversity and similarity for keyword extraction.

Table 4.1: Sample input to the greedy algorithm.

Words	$p(z_1\|\cdot)$	$p(z_2\|\cdot)$	$p(z_3\|\cdot)$	$p(z_4\|\cdot)$
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

Initially C is empty. The reward values, $h(w, C = \emptyset)$, for all words and $\lambda \in \{.75, 1\}$ are shown in Table 4.2. In the first step of the algorithm, w_1 (the best representative of topic z_1) is added to the set C for both values of λ. In the second step, the $h(w, C = \{w_1\})$ values are computed for the remaining unselected words and both values of λ, and are shown in Table 4.2. 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.

Table 4.2: The $h(w, C)$ values calculated using Algorithm 1 to select two keywords out of 5 words for $\lambda = .75$ and 1.

	$\boldsymbol{\lambda = 1}$		$\boldsymbol{\lambda = .75}$	
Words	$h(\cdot, \emptyset)$	$h(\cdot, \{w_1\})$	$h(\cdot, \emptyset)$	$h(\cdot, \{w_1\})$
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**

4.3 Data and Evaluation Methods

The proposed keyword extraction method is tested on two conversational corpora, the Fisher Corpus (Cieri et al., 2004), and the AMI Meeting Corpus (Carletta, 2007). The relevance of the keywords is assessed by designing first a comparison task, and then averaging the judgments obtained by crowdsourcing the task through the Amazon Mechanical Turk (AMT) platform[1]. For the qualification control, we use the PCC-H method defined in Chapter 3. In addition, the α-NDCG measure (Clarke et al., 2008) is used to measure topic diversity in the list of keywords.

4.3.1 Conversational Corpora Used for Experiments

The Fisher Corpus (Cieri et al., 2004) contains about 11,000 topic-labeled telephone conversations, on 40 pre-selected topics (one per conversation). In our experiments, we use the manual reference transcripts available with the corpus. We create a topic model using the Mallet (McCallum, 2002) 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 is used to build 11 artificial conversation fragments (1-2 minutes long) for testing, by concatenating 11 times three fragments about three different topics. Therefore, each test fragment is composed of three parts, each one about a different topic.

The AMI Meeting Corpus (Carletta, 2007) contains conversations on designing remote controls, in series of four scenario-based meetings each, for a total of 138 meetings. Speakers are not constrained to talk about a single topic throughout a meeting, hence these transcripts are multi-topic. The annotation of "episodes" that is provided with the AMI Corpus considers goal-based rather than topic-based episodes, therefore this annotation is not usable here to determine topical changes. Instead, we perform automatic topic analysis with LDA. However, the number of meetings in the AMI Corpus is not large enough for building topic models with LDA, which is why we use a subset of the English Wikipedia with 124,684 articles. Following several previous studies (Boyd-Graber et al., 2009; Hoffman et al., 2010), we fix the number of topics at 100.

We use 8 conversation fragment, each having 1-2 minutes length, from the AMI Meeting Corpus to set the parameters in our experiments. We also select a separate set of conversation fragments for testing which contains 8 conversation fragments, each 1-2 minutes long, from the AMI Corpus. We use both manual and ASR transcripts of these fragments. The ASR transcripts are generated by the AMI real-time ASR system for meetings (Garner et al., 2009), with an average word error rate (WER) of 36%.

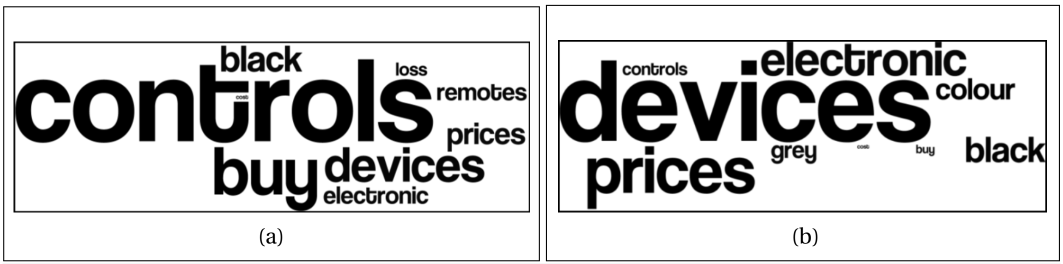
In addition, for experimenting with a variable range of WER values, we simulate ASR noise over the AMI manual transcripts in terms of word deletions, insertions and substitutions. Namely, we randomly delete words, or add new words, or substitute words by other words, in a systematic manner, i.e. all occurrences of a given word type are altered. We randomly select 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%.

[1] Following other researchers who designed methods for extracting keywords from conversational transcripts, such as like Harwath and Hazen (2012), we compare the methods using Amazon Mechanical Turk (AMT). If ground truth data were available, we could also use other evaluation techniques used in previous studies on keyword extraction from texts, but this is not the case here. Generating ground truth keywords for our conversations appeared to require much more effort than the AMT-based evaluation.

The result of this simulation technique is actually more challenging to our application than real ASR errors, because these are not created randomly and tend to spare long content words as they have fewer homophones.

4.3.2 Designing Tasks to Compare the Representativeness of Keyword Lists

We define comparison tasks to evaluate the relevance or the representativeness of extracted keywords with respect to each conversation fragment. Similarly to the four-choice "Human Intelligence Tasks" (HITs) designed in Chapter 3, for each task shown to a human subject, we first display the transcript of the fragment in a web browser, followed by several control questions about its content, and then by two lists of keywords (typically, with nine keywords each in our experiments). To improve readability, the keyword lists are presented using a word cloud representation generated by the Wordle™ tool[2], in which the words ranked higher are emphasized in the cloud. The subjects have to read the conversation transcript, answer the control questions, and then decide which keyword cloud better represents the content of the conversation fragment. The task is exemplified in Figure 4.3, without the control questions. The conversation transcript for this example is given in Figure 4.7.



Please select one of the following options:
1. Image (a) represents the conversation fragment better than (b).
2. Image (b) represents the conversation fragment better than (a).
3. Both (a) and (b) offer a good representation of the conversation.
4. None of (a) and (b) offer a good representation of the conversation.

Figure 4.3: Examples 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 are generated using Wordle™ from the lists produced in this example by: (a) the diverse keyword technique with λ =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.

For the comparisons presented in this section, we use 11 conversation fragments from the Fisher Corpus and 8 conversation fragments from the AMI Meeting Corpus, hence 19 HITs. There are 10 workers per HIT, and each one is paid 0.20 USD per HIT. The average time per assignment is almost 2.5 minutes. For qualification control, we accept workers with greater than 95% approval rate or with more than 1000 approved HITs. We also reject answers from

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

the workers who answered incorrectly the control questions.

The results obtained on the eight manual transcripts of the AMI conversation fragments used for testing (noted A–H) are shown in Table 4.3, for the comparison of the representativeness of the keywords extracted by the proposed keyword extraction (noted D(.75) as explained in Section 4.4) with a method using only topic similarity (noted TS). The ten subjects can choose between the four comparative answers presented in Figure 4.3, which amount to: 'X better than Y', 'Y better than X', 'both good', or 'both bad'. No answers are rejected for these HITs. The total counts for each answer and each HIT from Table 4.3 indicate that subjects agreed strongly on certain answers to some HITs (e.g. C, D, H) but disagreed on others (mainly A).

Table 4.3: 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 (TS) 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

4.3.3 Measure for Evaluating Topical Diversity

To evaluate the diversity of a keyword set $C = \{c_1, \cdots, c_k\}$, we use the α-NDCG measure proposed for information retrieval (Clarke et al., 2008), which rewards a mixture of relevance and diversity. We set the α parameter to 0.5, which rewards both relevance and diversity with equal weights, as used in the TREC 09 Web track (Clarke et al., 2009) and several other studies (Chandar and Carterette, 2011; Yin et al., 2009). 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 Equation 4.7, where $DCG[k']$ is obtained by Equation 4.8 at rank k' and $\text{DCG}_{\text{ideal}}[k']$ is computed by reordering keywords in a way that maximizes $DCG[k']$ values in each rank. In Equation 4.8, Z_t is the set of mono-topic dialogues included in the conversation fragment t, $Nr_{z_t,i-1}$ is the number of relevant keywords to the mono-topic dialogue z_t up to the rank i, and $Jr(c_i, z_t)$ measures the relevance of the keyword c at the rank i to the mono-topic dialogue z_t according to Equation 4.9. We set the relevance of a keyword to a conversation topic at 1 when the keyword belongs to the corresponding fragment, as shown in Equation 4.9 (note that a keyword may thus be relevant to several topics). A higher α-NDCG value indicates that keywords from the set are more uniformly distributed across the three topics.

$$\alpha\text{-NDCG}[k'] = \text{DCG}[k']/\text{DCG}_{\text{ideal}}[k'] \tag{4.7}$$

$$\text{DCG}[k'] = \sum_{i=1}^{k'} \frac{\sum_{z_t \in Z_t} Jr(c_i, z_t)(1-\alpha)^{Nr_{z_t,i-1}}}{\log_2(1+i)} \tag{4.8}$$

$$Jr(c_i, z_t) = \begin{cases} 1 & \text{if } c_i \in z_t \\ 0 & \text{otherwise} \end{cases} \tag{4.9}$$

4.4 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. We compare the following systems:

- two versions of the proposed diverse keyword extraction method (Algorithm 1) noted D(λ) for $\lambda \in \{.5, .75\}$;
- a method using only word frequency (excluding stopwords), noted WF;
- a recent method based on topical similarity but which does not enforce diversity (Harwath and Hazen, 2012), noted TS. This method coincides with one version of the proposed method D(λ) for $\lambda = 1$ as well.

4.4.1 Selection of Configurations

The tested values of λ are motivated as follows. As the relevance of keywords for D(.5) appears to be quite low, we do not test lower values of λ. Similarly, we do not test additional values of λ between .5 and 1, apart from .75, because the resulting word lists are very similar to the tested values. The similarity comparison scores between two keyword lists at rank 15 over 8 conversation fragments used as our development set are shown in Figure 4.4. In this figure, 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, Webber et al. (2010)) as a similarity metric, based on the fraction of keywords overlapping at different ranks.[3]

[3] RBO is computed as follows. Let A and B be two ranked lists, and let a_i be the keyword at the rank i in the set A. The set of the keywords up to the rank j in A is $\{a_i : i \le j\}$, noted as $A_{1:j}$. RBO is calculated as in Equation 4.10, in which k is the size of the ranked lists:

$$RBO(A,B) = \frac{1}{\sum_{j=1}^{k}(\frac{1}{2})^{j-1}} \sum_{j=1}^{k} (\frac{1}{2})^{j-1} \frac{|A_{1:j} \cap B_{1:j}|}{|A_{1:j} \cup B_{1:j}|} \tag{4.10}$$

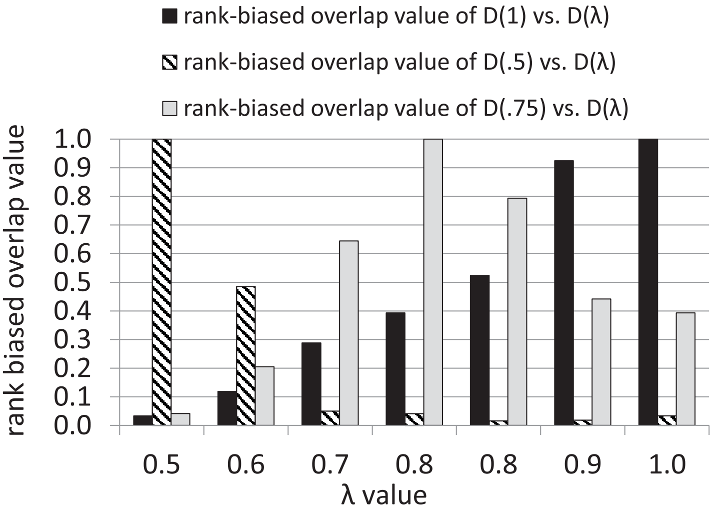


Figure 4.4: Comparison of keyword lists generated by D(λ) when $.5 \leq \lambda \leq 1$ using the rank biased overlap metric (RBO) computed between D(λ) and the three keyword lists generated by D(1), D(.5) and D(.75). The differences in RBO are small enough to allow clustering around the three values of λ.

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 $\lambda = 1$. As mentioned above, TS can be reformulated by D(1), because the diversity factor of $\lambda = 1$ just considers the similarity of topics, but not the diversity of them in the final keyword set. Therefore, in our human evaluations, we will consider only the D(.5), D(.75).

4.4.2 Results for the Topical Diversity of Keywords

First of all, we compare 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 (Equation 4.7) to measure how evenly the extracted keywords are distributed across the three topics of each fragment.

Figure 4.5 shows results averaged over the 11 three-topic conversation fragments of the 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, when 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.

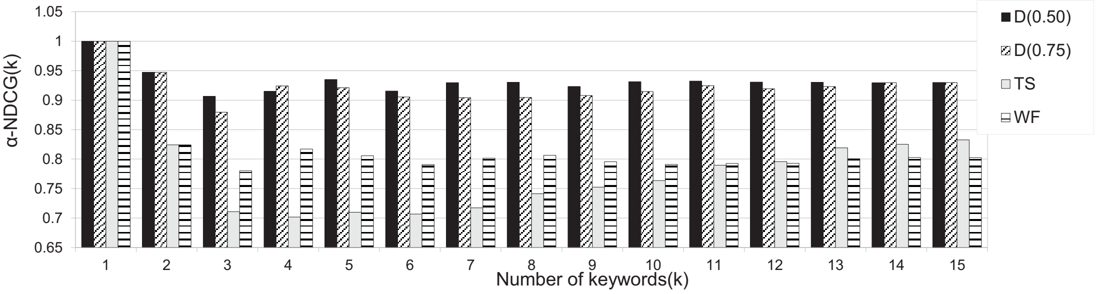


Figure 4.5: Average α-NDCG values over the 11 three-topic conversation fragments of the Fisher Corpus, for a number of keywords varying from 1 to 15. The most difficult task is to extract exactly one keyword from each topic, hence the lowest scores are for three keywords. The best performing methods are always the diversity-preserving ones, D(.5) and D(.75).

4.4.3 Results for Keyword Relevance

We perform binary comparisons between the outputs of each keyword extraction method at rank 9. We only experimented with a fixed rank because we cannot compare the methods at different ranks using crowdsourcing because this would be very time-consuming. Moreover, since the keyword cloud representation which contains too many keywords is not readable enough to be judged by human subjects, we decided to select only 9 keywords by each method for comparison, which is on average one-third of all the words except stopwords of each fragment (Mihalcea and Tarau, 2004; Rose et al., 2010). The length of each conversation fragment is approximately two minutes. The experiments are performed using crowdsourcing, over 11 fragments from the manual transcripts of the Fisher Corpus and 8 fragments from the manual transcripts of the AMI Corpus. We assume that our comparative evaluation method is transitive, as exemplified for instance by the complete set of comparisons in Table 5.1. Therefore, we only report here the binary comparisons that allowed us to determine the ordering of the four methods, and exclude redundant comparisons, which we tried to minimize because of their costs.

We compute comparative relevance values using the PCC-H method using the two approaches described in Section 3.4, noted here PCC-H Score 1 and Score 2 respectively. In addition to PCC-H scores, we also report the raw preference scores for each comparison, i.e. the number of times a system is preferred over the other, although PCC-H was shown to be a more reliable indicator of quality.

Table 4.3 (in Section 4.3.2) shows as an illustration the judgments that are collected when comparing the output of D(.75) with TS on the 8 HITs of the AMI Corpus. Workers tend to disagree about the first two HITs, but then clearly find that the keywords extracted by D(.75) for the subsequent six HITs better represent the conversation compared to TS. For these results, our comparative relevance score (PCC-H) is 78% for D(.75) vs. 22% for TS.

The averaged relevance values for all comparisons needed to rank the four methods are shown

Table 4.4: Comparative relevance scores based on human judgments for four keyword extraction methods over both manual and ASR transcripts. The following ranking can be inferred: D(.75) > TS > WF > D(.5).

Corpus	**Compared methods**	**Relevance (%)**					
	(m_1 vs. m_2)	**PCC-H Score 1**		**PCC-H Score 2**		**Raw Score**	
		m_1	m_2	m_1	m_2	m_1	m_2
Fisher manual transcripts	D(.75) vs. TS	**68**	32	**.72±.17**	.37±.19	**60**	40
	TS vs. WF	**82**	18	**.74±.24**	.20±.20	**71**	29
	WF vs. D(.5)	**95**	5	**.89±.09**	.09±.08	**93**	7
AMI manual transcripts	D(.75) vs. TS	**78**	22	**.84±.08**	.28±.13	**74**	26
	TS vs. WF	**60**	40	**.63±.10**	.43±.11	**59**	41
	WF vs. D(.5)	**78**	22	**.85±.07**	.31±.10	**73**	27
AMI ASR transcripts	D(.75) vs. TS	**79**	21	**.80±.14**	.46±.17	**68**	32
	TS vs. WF	**65**	35	66±.11	.43±.19	**62**	37
	WF vs. D(.5)	**73**	27	**.77±.14**	.37±.20	**70**	30

in Table 4.4, for the manual transcripts of the Fisher and the 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 difference in relevance of D(.75) vs. D(.5) on the Fisher Corpus is very large (96% vs. 4%).

4.4.4 Results for Noise Reduction Power

Turning now to ASR output, we perform binary comparisons between the keyword lists at the rank 9 over the 8 fragments from the 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 in the last three lines of Table 4.4. The differences between comparison values are similar for the ASR transcripts and 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)**. The results confirm that our method is robust to ASR noise[4].

We have counted the number of keywords selected by each method among ASR errors, which are artificially generated as explained in Section 4.3.1, so that these words are precisely known. The average numbers of such erroneous keywords are shown in Figure 4.6 for a noise level

[4] Our results corroborate those obtained in the field of spoken document retrieval: for instance, Garofolo et al. (2000) have shown that their retrieval method is robust to ASR noise.

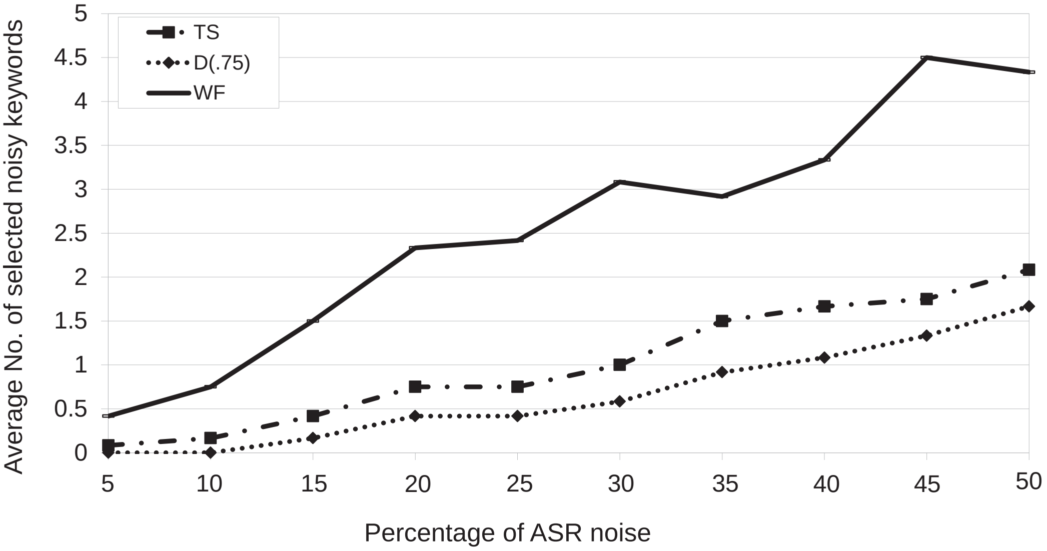


Figure 4.6: Average number of noisy keywords which are chosen by the algorithms over the 8 conversation fragments of the AMI Corpus, for a percentage of artificial ASR noise varying from 5% to 50%. The best performing method is D(.75).

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 topics 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 D(.75) over TS is that D(.75) selects a smaller number of noisy keywords from it.

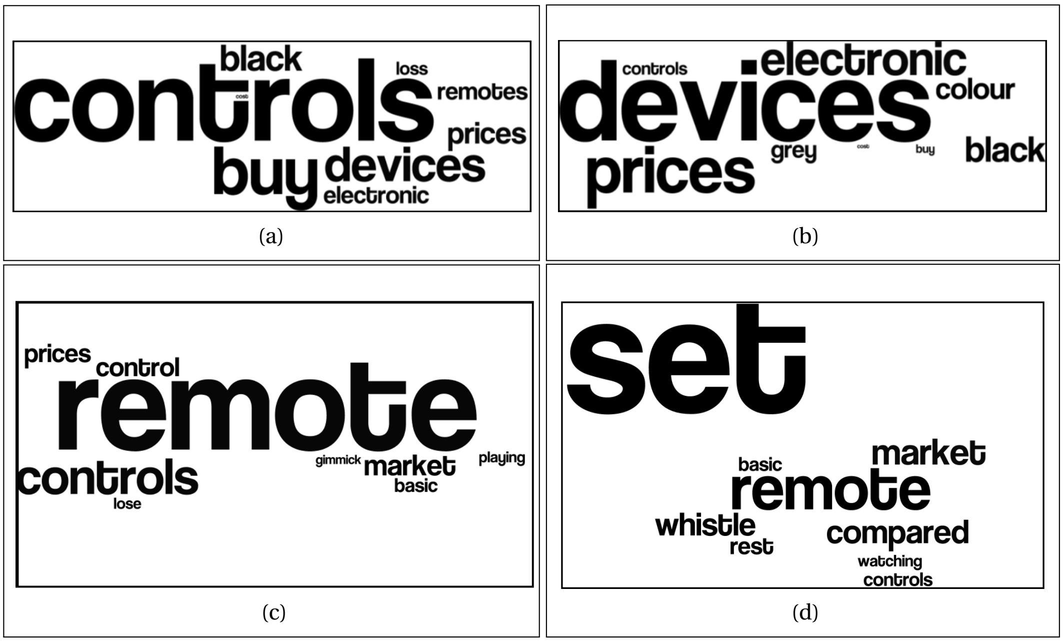
4.4.5 Examples of Keyword Clouds

To illustrate the superiority of D(.75) over the other techniques, we consider an example from one of the conversation fragments of the AMI Meeting Corpus, the transcript of which is given in Figure 4.7. In this fragment, four individuals are discussing the impact on sales of several features of a remote control. The lists of keywords of size $|C| = 9$ extracted from this fragment using the four techniques of D(.75), TS, WF and D(.5) are presented using word clouds in Figure 4.8.

The topical similarity method, TS, over-represents the "color" topic by selecting three words related to it, as well as "electronic devices" by choosing two relevant words. However, TS misses other topics such as "remote control", and "losing a device" which are also representative of the fragment. Although the word frequency method, WF, can recognize the keywords related to the "remote control" and "buying a remote control" topics, the method has difficulties

A: The only the only remote controls I've used usually come with the television, and they're fairly basic. So uh
D: Yeah. Yeah.
C: Mm-hmm.
D: Yeah, I was thinking that as well, I think the the only ones that I've seen that you buy are the sort of one for all type things where they're, yeah.
D: So presumably that might be an idea to
C: Yeah the universal ones. Yeah.
A: Mm. But but to sell it for twenty five you need a lot of neat features. For sure.
D: put into.
B: Slim.
C: Yeah.
D: Yeah, yeah.
D: Uh 'cause I mean, what uh twenty five Euros, that's about I dunno, fifteen Pounds or so?
C: Mm-hmm, it's about that.
D: And that's quite a lot for a remote control.
A: Yeah, yeah.
C: Mm. Um well my first thoughts would be most remote controls are grey or black.
C: As you said they come with the TV so it's normally just your basic grey black remote control functions, so maybe we could think about colour?
C: Make that might make it a bit different from the rest at least. Um, and as you say, we need to have some kind of gimmick, so um I thought maybe something like if you lose it and you can whistle, you know those things?
D: Uh-huh. Mm-hmm. Okay. The the keyrings, yeah yeah. Okay, that's cool.
C: Because we always lose our remote control.
A: Right.
B: Uh yeah uh, being as a Marketing Expert I will like to say like before deciding the cost of this remote control or any other things we must see the market potential for this product like what is the competition in the market?
B: What are the available prices of the other remote controls in the prices? What speciality other remote controls are having and how complicated it is to use these remote controls as compared to other remote controls available in the market.
D: Okay.
B: So before deciding or before finalising this project, we must discuss all these things, like and apart from this, it should be having a good look also, because people really uh like to play with it when they are watching movies or playing with or playing with their CD player, MP three player like any electronic devices. They really want to have something good, having a good design in their hands, so, yes, all this.

Figure 4.7: Transcript of a sample fragment from a four-party conversation (speakers noted A through D) from the AMI Meeting Corpus, which was submitted to our diverse keyword extraction method and to the three baselines, with results shown in Figure 4.8.



(a) (b) (c) (d)

Figure 4.8: Examples of keyword sets (9 words each) obtained by four keyword extraction methods for a fragment of the AMI Corpus discussing the impact of the features of a remote control on its sales. The word clouds are generated using Wordle™ from the lists produced in this example by: (a) diverse keyword extraction with λ =0.75, (b) topical similarity, (c) word frequency, and (d) diverse keyword extraction with λ =0.5 methods.

to extract keywords related to “different colors of a remote control” and “losing a remote control” (low-frequency words which represent a main topic), and also selects keywords that are not relevant to the main topics such as “basic” and “playing”, because it ignores topical information. The diverse keyword extraction method with $\lambda = .5$ chooses keywords relevant to minor topics of the fragment by giving more reward to topical diversity than to topical similarity. The diverse method with $\lambda = .75$ provides a list of keywords which covers the maximum number of main topics by rewarding both topical diversity and relevance. Moreover, this method does not select keywords from irrelevant topics thanks to its appropriate balance between topical diversity and relevance.

4.5 Using Diverse Keyword Extraction for Content Representation

To demonstrate its generality beyond the just-in-time retrieval scenario (ACLD), we apply the proposed diverse keyword extraction technique to a multimedia recommendation system, MUST-VIS. This system was designed by the Idiap NLP group (Bhatt et al., 2013) for the MediaMixer/VideoLectures.NET Temporal Segmentation and Annotation Grand Challenge at ACM Multimedia 2013, and was declared the winner of this challenge.

Users of lecture databases are confronted with the problem of efficient browsing or search, especially for specific pieces of information such as facts, arguments or references. Moreover, understanding the main content of a lecture at a glance without totally watching it is another challenge for lecture browsers. The MUST-VIS system allows users to visualize a lecture as a series of segments which are represented by keyword clouds, with relations to other similar lectures and segments.

Our diverse keyword extraction method is utilized to represent each piece of information (a segment or a lecture) as a keyword set. These keywords are then used by a content-based recommendation system to compute the cosine similarity between segments or lectures. They are also used to visualize the content of segments or lectures as keyword clouds.

4.5.1 Description of the MUST-VIS System

The MUST-VIS system presents to users a graphical user interface as shown in Figure 4.9. When users hover over lecture segments with the mouse, an overview of them is shown to users by magnifying the keyword clouds. The segments of each lecture are arranged in chronological order in a clockwise circular manner around a keyframe of the lecture. The lecture presented at the center of the screen is considered to be in focus and can be played (audio, video, and slides) by clicking on it.

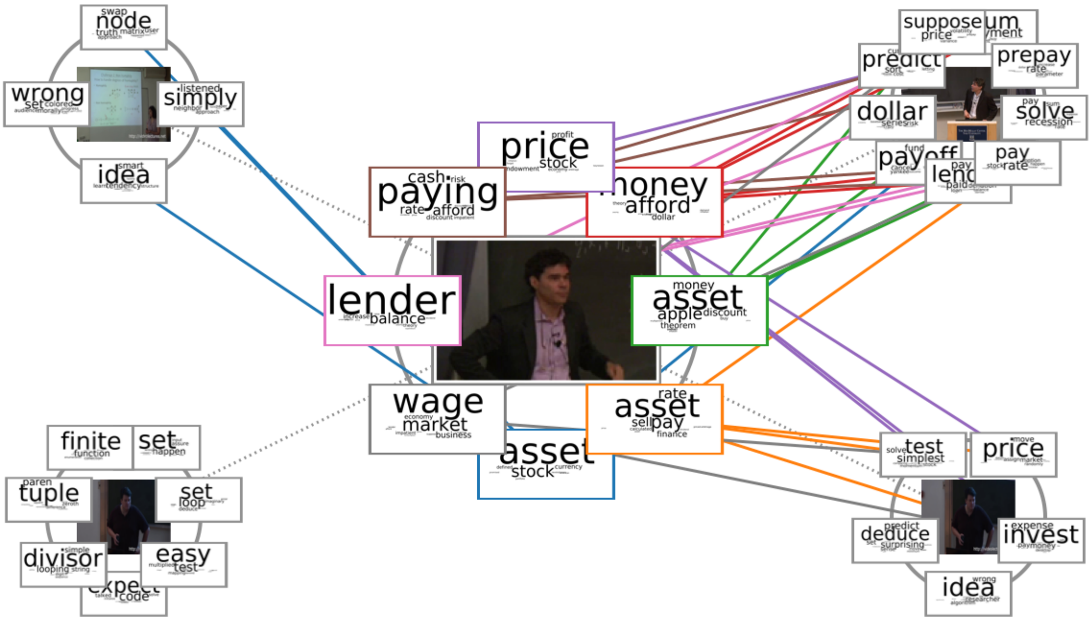


Figure 4.9: Navigation graph of the MUST-VIS system (Bhatt et al., 2013). Each lecture is represented with a keyframe and keyword clouds for each segment around it. The lecture in focus is surrounded by other lectures with related segments.

Using content-based recommendation techniques (Pappas and Popescu-Belis, 2015), each segment and each lecture are related to the similar ones, by a navigation graph. The interface displays segment-to-segment similarity links (color-coded) and lecture-to-lecture ones (dashed). In Figure 4.9, the content of the lecture in focus is highly similar to the ones in the right side according to the information provided by the user interface. However, the ones in the left side have less similarity to the focused lecture, based on the links and word clouds. Overall, the segment-to-segment links and the keyword clouds make the browsing of relevant segments easier and faster for users.

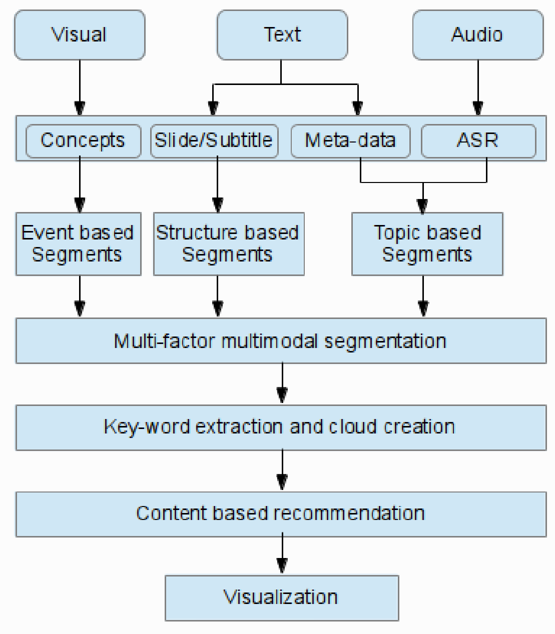


Figure 4.10: Lecture processing in MUST-VIS (Bhatt et al., 2013).

The MUST-VIS system uses state-of-the-art methods for multimodal processing of audios, videos and texts. The components of the system are shown in Figure 4.10. The ASR transcripts or the subtitles of each lecture of the dataset are segmented using the TextTiling algorithm implemented in the NLTK toolkit (Bird, 2006). Then the words in each segment are ranked using the proposed diverse keyword extraction technique, which selects keywords so that they cover the maximum number of topics mentioned in each segment. Word clouds are generated using WordCram[5] for each segment and also for entire lectures. Words ranked higher become graphically emphasized in the word cloud. Similarity across lectures and segments is then computed using a state-of-the-art content-based recommendation algorithm based on cosine similarity and is used to generate the navigation graph.

4.5.2 Evaluation Results

In a pilot experiment, we recruited two experts to assign ground-truth recommendations to all the lectures of the dataset. We compare in Table 4.5 the ground-truth rankings with the automated methods: lecture-to-lecture (LL), segment-to-segment (SS), and a random

[5] http://www.wordcram.org

policy (R),using Mean Reciprocal Rank (MRR) and Mean Average Precision (MAP). The highest agreement is between the two annotators, followed very closely by the agreement between LL and averaged SS similarities. The lecture-level recommendation is slightly closer to the ground-truth than the averaged segment-level one, although both are at considerable distance with respect to inter-coder agreement. Both recommendation techniques benefit from keyword extraction, though in these experiments it was not possible to assess independently the contributions of alternative keyword extraction methods to recommendation relevance.

Table 4.5: Comparison of ground-truth recommendations (A1, A2) with automated ones: lecture-to-lecture (LL), segment-to-segment (SS) and random (R, as baseline, 500 draws), using MAP (a) and MRR (b) over 1–5 top recommendations. LL, using lecture-level keywords, is the closest to the ground truth.

	A1		**A2**		**LL**		**SS**		**R**	
	(a)	(b)	(a)	(b)	(a)	(b)	(a)	(b)	(a)	(b)
A1	1.0	1.0	.91	.54	.33	.14	.28	.10	.13	.03
A2	-	-	1.0	1.0	.31	.15	.22	.10	.10	.02
LL	-	-	-	-	1.0	1.0	.91	.52	.29	.11
SS	-	-	-	-	-	-	1.0	1.0	.30	.11
R	-	-	-	-	-	-	-	-	1.0	1.0

4.6 Conclusion

We compared the diverse keyword extraction technique with existing methods based on word frequency or topical similarity, in terms of the representativeness of the extracted keywords. The comparisons involved both the manual and the ASR transcripts of two conversational corpora. The keywords 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 for topical diversity. Moreover, we have shown that the diverse keyword extraction method selects fewer words from ASR noise compared to the other methods. Therefore, enforcing both relevance and diversity brings an effective improvement to keyword extraction. We also exemplified the use of this method to extract keywords for visualizing and recommending segments of video lectures transcripts.

In the next chapter, we will use keywords to build queries in order to retrieve documents from a repository and recommend them to conversation participants. We will assess the retrieval results of queries made by the proposed technique in comparison to others. We will then divide the keyword list into several topically-separated keyword subsets, perform separate retrieval operations, and merge the results.

5 Formulation of Implicit Queries from Conversations

The focus of this chapter is on the formulation of implicit query for a just-in-time-retrieval system, which is in our case a recommender system designed for conversations. The goal is to use these queries to retrieve, for each short conversation fragment, a small number of potentially relevant documents, which can be recommended to participants.

As people in a conversation often shift from one topic to another, conversation fragments are usually multi-topic. The proposed method relies on the diverse keyword extraction method proposed in Chapter 4 to maximize the coverage of potential users' information needs as well as to reduce the impact of ASR noise. Then, the method clusters the extracted keywords into several topically-separated subsets, and builds from each of them an implicit query.

The presented method is evaluated in terms of the relevance of document results, retrieved using the implicit queries over the English Wikipedia, rated by several human judges. We show that the document results of the queries obtained by the diverse keyword extraction technique surpass those of other techniques. Moreover, we show that modeling users' information needs using multiple topically-separated queries instead of a single query made of the entire keyword set generates more relevant results to recommend. In this chapter, the experiments are carried out over conversation fragments from the ELEA (Emergent LEader Analysis) conversational corpus (Sanchez-Cortes et al., 2012), based on a brainstorming task.

5.1 Introduction

In Chapter 4, we have shown that even a short fragment of a conversation contains a variety of words, which are potentially related to several topics. We have also demonstrated that the diverse keyword extraction technique provides the best representative keyword set for a fragment by rewarding both the relevance and the diversity of topics. Here, we aim to define different policies to construct implicit queries out of these keywords, for each fragment, and then compare different types of queries in terms of the relevance of retrieved documents. For instance, in the example discussed in Section 5.4.4 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?

Previous methods for implicit query formulation used in existing just-in-time retrieval systems rely on the entire keyword list extracted either with word frequency or with TFIDF (Luhn, 1957; Salton and Buckley, 1988). As we have shown in the previous chapter, these techniques do not optimally capture users' information needs. For instance, while a method based on word frequency would retrieve the following Wikipedia pages: "Light", "Lighting", and "Light My Fire" for the fragment discussed in Section 5.4.4 below, users would likely prefer a set such as "Lighter", "Wool" and "Chocolate" which covers more main topics of their conversation. Moreover, using the entire keyword set as a single multi-topic query is not an optimal solution either, because irrelevant topics in a query might cause poor retrieval results (Bhogal et al., 2007; Carpineto and Romano, 2012).

Given the potential multiplicity of topics, reinforced by potential ASR errors or speech disfluencies (such as 'whisk' in the example fragment), 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. Following this goal, we first extract a relevant and diverse set of keywords from each conversation fragment using the method from Chapter 4, and then cluster the keywords into topic-specific queries ranked by importance. Finally we present to users a sample of document results from these queries. In this chapter, we select the best document representative from the retrieval results of each query and rank them based on the importance of the query to be answered, but in the following chapter, we will improve over this strategy and propose a diverse merging method.

In the proposed method for building implicit queries, the topical diversity of keywords increases the chances to cover the main topics discussed in a fragment, and decreases the effect of keywords which are actually ASR noise. In addition, the topic-based clustering decreases the noisy effect of the mixture of topics in queries.

The chapter is organized as follows. In Section 5.2 we describe the proposed technique for implicit query formulation, which relies on the proposed diverse keyword extraction algorithm (Chapter 4) and a topic-aware clustering method (5.2.2). Section 5.3 introduces the data and our method for comparing the relevance of sets of recommended documents using crowdsourcing, while in Section 5.4 we present and discuss the experimental results, including sample results for the conversation fragment given in Figure 5.1.

5.2 Method for the Formulation of Implicit Queries

We propose a two-stage approach to the formulation of implicit queries. We first extract a set of keywords $C = \{c_1, \ldots, c_k\}$ from the transcript of users' conversation using the diverse keyword extraction technique defined in Section 4.2. The diverse set of extracted keywords is hypothesized to represent the information needs of the participants in a conversation, in terms of the notions and topics that they mention. We will split this set into several topically-disjoint subsets, to maintain the diversity of topics and to reduce the noisy effect of each topic on the others (if they appeared in the same query). Each subset corresponds then to an implicit query that can be sent to a document retrieval system.

Our proposal consists of two stages: we first represent the entire conversation fragment and each keyword using topical information, and then we build implicit queries by topically clustering of keywords, as explained in the following subsections.

5.2.1 Keyword Representation Using Topic Models

We extract topic models using the LDA topic modeling technique (Blei et al., 2003) as we did in Chapter 4, Section 4.2.1. The hyper-parameters of the LDA topic model are again optimized using the Gibbs sampling implemented by the Mallet toolkit. We note the distribution over the topic z of each keyword $c_i \in C$ by $p(z|c_i)$. We also represent each conversation fragment t by the set of weights (noted β_z[1]) for each topic of the fragment. β_z is computed by averaging over all probabilities $p(z|w)$ of the N words w spoken in the fragment t, as formulated in Equation 4.1 on page 36.

5.2.2 Topic-aware Keyword Clustering

Clusters of keywords are built by ranking keywords for each main topic of the fragment, as follows. The keywords are ordered for each topic z by the decreasing value of the similarity in the topic space (dot product) between the keyword $c_i \in C$ and the entire conversation fragment:

$$s_{c_i,z} = p(z|c_i) \cdot \beta_z \tag{5.1}$$

For each cluster (representing a main topic), only the keywords with topical similarity values higher than a threshold are kept. Note that a given keyword can appear in more than one cluster. Following this ordering criterion, keywords with higher value of $p(z|c_i)$ (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 higher value of β_z.

Each cluster of keywords represents one implicit query Q_i. The clusters are ranked and weighed by we_{im,Q_i} which are equal to their β_z values. Therefore, a set of topically-separated implicit queries along with their weights is defined in Equation 5.2:

$$Q_{implicit} = \{(Q_1, we_{im,Q_1}), \ldots, (Q_i, we_{im,Q_i}), \ldots, (Q_M, we_{im,Q_M})\} \tag{5.2}$$

[1] This weight is different from the hyperparameter of the LDA topic model β.

where M is the number of implicit queries.

5.3 Data and Evaluation Method

Our proposal is tested on the ELEA Corpus (Emergent LEader Analysis, collected by Sanchez-Cortes et al. (2012)) because there are several relevant documents from the Wikipedia articles to the topics discussed in this corpus. Moreover, it is a transcribed conversational corpus with a brainstorming task, so receiving just-in-time recommendations from a system would be beneficial to their participants.

The quality of implicit queries is assessed by estimating the relevance of the documents that are retrieved when submitting these queries to the Lucene search engine[2] over the English Wikipedia[3] (version dated 2009-06-16) and merging the results as explained below. The comparison scores are obtained by designing a comparison task and averaging several judgments obtained by crowdsourcing this task through the Amazon Mechanical Turk (AMT) platform. For the qualification control, we use the PCC-H method which is described in Chapter 3.

5.3.1 Conversational Corpus Used for Experiments

The ELEA Corpus (Sanchez-Cortes et al., 2012) 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 mountains until they are rescued. In our experiments, we consider 5 conversations in English, of about 15 minutes each, and divide their transcripts into 35 segments, of about two minutes each.

We perform our experiments only over the manual transcripts of the ELEA Corpus, as the study of ASR noise is less central to this chapter. In fact, we showed in the previous chapter that the diverse keyword extraction method was more robust to ASR noise compared to other methods. We also showed experimentally that the proposed method adds fewer than one noisy word out of nine when word error rate is smaller than 30%, which is assumed to be the case here, as the best recognition accuracy reaches around 70% for conversational activities (Hain et al., 2010).

In Chapter 4, we compared the keyword clouds, each contains 9 keywords and, have shown experimentally the ones prepared by D(0.75) method outperform those of the others in terms of relevance and diversity over main topics. Therefore, we decide to extract 9 keywords from each conversation fragment of the ELEA corpus and hope to properly cover three or four main topics conveyed in the fragment within the keyword set. Finally we build implicit queries by

[2] Version (Lucene 4.2.1 API) is available in http://lucene.apache.org. We employed the Standard Analyzer for indexing, and TF-IDF similarity values between documents and queries for ranking document results.

[3] A local copy downloaded from the Freebase Wikipedia Extraction (WEX) dataset from http://dumps.wikimedia.org.

clustering these keywords. We also set the threshold of the clustering to 0.10. This threshold value is tuned using 9 conversation fragments out of 35 in terms of achieving meaningful queries. To obtain these queries, the keywords which have a very small distribution over the topic z are not selected in the implicit query formulated for the topic z. Moreover, having duplicate implicit queries in the list of queries is avoided. Finally, the 26 remaining fragments are utilized to evaluate the policies used for query formulation.

Since the number of meetings in the ELEA Corpus is not large enough for building topic models with LDA, we use a subset of the English Wikipedia with 124,684 articles. Following several previous studies (Boyd-Graber et al., 2009; Hoffman et al., 2010), we fix the number of topics at 100.

5.3.2 From Keywords to Document Recommendations

We submit the implicit queries to the Lucene search engine to retrieve documents from the English Wikipedia articles. We perform our experiments using two types of queries made of keywords extracted from the conversation: (1) a single implicit query per conversation fragment formulated using the entire keyword list; (2) multiple topically-separated implicit queries with the keywords of each cluster as described above.

In the experiments with only one implicit query per conversation fragment, the document results corresponding to each conversation fragment are prepared by selecting the first six documents retrieved for the implicit query.

In the experiments with multiple implicit queries, we prepare a single document list per conversation fragment using a simple merging method. The merging method first takes the first-best document retrieved for each implicit query, and then ranks these documents based on their corresponding query weights. This procedure is then repeated for the second-best results and so on. Similar to our experiments with single implicit queries, we show to users the first six documents obtained by this merging method. This is a baseline algorithm, but we will describe an improved ranking method in Chapter 6.

5.3.3 Designing Tasks to Compare the Relevance of Recommended Documents

We compare the relevance of two lists of recommended documents for the same conversation fragment by designing the four-choice “Human Intelligence Tasks” (HITs) as described in Section 3.3.

For each comparison task we recruit 10 workers. Each worker is paid 0.20 USD per HIT. The average time spent per HIT is about 2.5 minutes. For qualification control, we only accept workers with greater than 95% approval rate and with more than 1000 approved HITs. We only keep answers from the workers who answer correctly our control questions about each HIT. Finally, we compute the comparative relevance scores based on the PCC-H method proposed in Section 3.4. Moreover, we present the averaged raw preference scores for each comparison in addition to PCC-H scores.

5.4 Experimental Results

We compare the two policies 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 (5.2.2) and using each sub-set as one implicit query, merging afterward the results into a unique document set (5.3.2).[4]

We also compare the retrieval results of queries built from keyword lists obtained by the three keyword extraction techniques presented in Section 4.4. Firstly, we build single queries from the keyword sets provided by the D(.75), TS and WF keyword extraction methods, and compare the three resulting document sets. Secondly, we build multiple queries from the same methods and perform similar comparisons between the resulting document sets.

Finally, we compare the best results of multiple queries with the best results of the single queries. This procedure is used because evaluation with human subjects is time-consuming, therefore we attempt to carry out the minimal number of binary comparisons allowing the ordering of the methods.

5.4.1 Comparison across Single Queries

Binary comparisons are performed between the retrieval results from single queries based respectively on D(.75), TS, and WF, over 26 fragments from the ELEA Corpus. The workers compare two document lists in terms of the relevance or utility of suggested documents to the meeting participants at the time of each fragment, represented through its transcript.

The average relevance scores for the comparisons needed to rank the three methods are shown in Table 5.1. These values are computed based on the first approach proposed for calculating comparative scores presented in Section 3.4. 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. The comparison value obtained by the second approach for D(.75) vs. TS is .69±.10 vs. .39±.10. This comparison shows that our method significantly outperforms the TS method.

[4] In the initial version of the system described in Chapter 3, we used the entire conversation fragment, minus the stopwords, as a single (long) query. We also evaluated this approach, comparing its results with those of the keyword lists based on word frequency (WF). We found that WF outperformed the use of the entire fragment, with 87% vs. 13% comparative relevance scores. Moreover, several previous just-in-time retrieval systems such as the Remembrance Agent also used WF rather than all words. Therefore, we will not discuss the approach using the entire conversation fragment any further in this section.

Table 5.1: Comparative relevance scores, as well as raw preference 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 (m_1 **vs.** m_2)	**Relevance (%)**			
	PCC-H Score 1		**Raw Score**	
	m_1	m_2	m_1	m_2
WF vs. TS	**54**	46	**55**	45
D(.75) vs. WF	**58**	42	**53**	47
D(.75) vs. TS	**70**	30	**62**	38

5.4.2 Comparison Across Multiple Queries

Binary comparisons are then performed between the retrieval results of multiple topically-disjoint queries. Multiple queries are 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 26 fragments from the ELEA Corpus show that CD(.75) outperforms CTS, with an averaged relevance value obtained by the first PCC-H approach of 62% vs. 38% as shown in the last line of Table 5.2.

Table 5.2: Comparative relevance scores, as well as raw preference 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 (m_1 **vs.** m_2)	**Relevance (%)**			
	PCC-H Scores 1		**Raw Scores**	
	m_1	m_2	m_1	m_2
CD(.75) vs. D(.75)	**65**	35	**60**	40
CTS vs. TS	**65**	35	**62**	38
CD(.75) vs. CTS	**62**	38	**60**	40

5.4.3 Single Queries Versus Multiple Queries

Finally, we compare single queries with multiple queries derived from the same keyword lists, namely D(.75) and TS, on the 26 fragments from the ELEA Corpus. The averaged relevance values obtained from human judgments, in the first two lines of Table 5.2, reveal that using multiple queries, for both types of keyword extraction techniques, leads to more relevant document results compared to single queries, i.e. **CD(.75)** > **D(.75)** and **CTS** > **TS**. For both comparisons, the averaged relevance scores obtained by the first approach 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. These values also are computed by the PCC-H second approach, in which the mean and the confidence interval of the estimation are calculated for each comparison. The values for CD(.75) vs. D(.75) are .76±.06 vs. .51±.12, and the values for CTS vs. TS are .70±.07 vs. .46±.07, again demonstrating the statistical significance of the difference.

5.4.4 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 Figure 5.1.As described in Section 5.3, for these meetings, 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 5.3. As WF does not consider topical information, the keywords it selects ('lighted', 'light', and 'lighter') all revolve around the same notion.

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 don't 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 don't 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 doesn't
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 didn't have that -
B: but – but – I mean the last two things it doesn't – it doesn't 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?

Figure 5.1: Sample transcript of a four-party conversation (speakers A–D) from the ELEA Corpus which was submitted to the document recommendation system.

Table 5.4 shows the topically-aware implicit queries prepared from the keyword lists provided

Table 5.3: Examples of keyword sets obtained by three keyword extraction methods for a fragment of the ELEA Corpus.

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

Table 5.4: 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 5.5: 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

by the D(.75) and TS keyword extraction methods (Table 5.3), ordered based on their importance in the fragment. The TS 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 D(.75) method which considers also topical diversity, first selects two keywords (‘chocolate’ and ‘cigarette’) to cover the first main topic. Then it selects the third keyword of the first main topic, ‘whiskey’, only after the selection of a keyword shared by the first three main topics. Afterward, it selects the keywords ‘pistol’, ‘wool’, ‘shoe’, and ‘fire’ to cover the second, third and fourth main topics of the fragment.

Finally, Table 5.5 shows the retrieval results (six 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. The single query made by the diverse keyword extraction technique (D(.75)) retrieves 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.

5.5 Conclusion

In this chapter, we focused on modeling the users' information needs by deriving implicit queries from short conversation fragments. These queries were based on sets of keywords extracted from the conversation. We proposed to use the diverse keyword extraction technique from Chapter 4, which covers the maximal number of important topics in a fragment, to extract users' information needs in the form of keywords. We proposed in this chapter a clustering technique to divide the set of keywords into smaller topically-independent subsets constituting implicit queries, to reduce the noisy effect of the mixture of topics in a single query.

We compared the diverse keyword extraction technique with methods based on word frequency or topical similarity, in terms of the relevance of retrieved documents. These were judged by human raters recruited via Amazon's Mechanical Turk. The experiments showed that the diverse keyword extraction method provides the most relevant lists of recommended documents through multiple topically-separated implicit queries. Therefore, enforcing both relevance and diversity brings an effective improvement to retrieved documents.

In the next chapter, the goal is 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. First, we will experimentally show that the use of diverse re-ranking techniques does not improve the retrieval results of a single query made from keywords related to multiple irrelevant topics, because off-topic words can ruin the retrieval results of the initial single query and consequently the re-ranking of the results does not help anymore. This again confirms the usefulness of preparing multiple topic-aware implicit queries instead of a single implicit query. Then we will propose a novel diverse merging method that can be applicable to the several document lists retrieved for multiple topically independent queries.

6 Diverse Merging of Document Lists

In this chapter, we propose a diverse merging technique for lists of documents, to combine the retrieval results of the topic-aware implicit queries prepared from the transcripts of conversation fragments, as explained in the previous chapter. Our goal is to suggest a unique and concise list of documents, every couple of minutes, to the participants in conversation. The recommended result list should cover the maximum number of implicit queries and should be as small as possible, because a long list of results may distract users from their actual conversation. In this chapter, we propose an algorithm for diverse merging of these lists, using a submodular reward function that rewards both topical similarity of documents to the users' information needs and their diversity. The method is evaluated through crowdsourcing in terms of both the relevance and diversity of recommended documents.

6.1 Introduction

We address the problem of merging lists of documents that are retrieved based on implicit queries into a concise, diverse and relevant list. In Chapter 4, we stated that even a short fragment is comprised of a variety of words, which can refer to several topics as people often jump from one topic to another during a conversation. Then, in Chapter 5, we showed that modeling users' information needs in the form of multiple topically-separated implicit queries (accompanied by a weight) instead of a single multi-topic implicit query can reduce the noisy effect of the mixture of topics on the retrieval results.

Here, we propose a new method to combine the lists of recommendations retrieved for implicit queries to build a single, short and comprehensive list of results. For instance, in the example discussed in Section 6.5.4 below, in which four people must select a list of the 12 items that would most help them to survive in the mountains, five implicit queries are derived from a short fragment of 120 seconds (with about 250 words). These queries have different weights: two queries are more important to be answered compared to the others. Therefore, which Wikipedia pages would best answer users needs at the respective moment of their conversation, and how would a system proceed to find them?

The chapter is organized as follows. In Section 6.2 we briefly present the motivation of our proposal, and in Section 6.3, we describe the proposed algorithm for diverse merging of lists of recommendations. Section 6.4 presents the data, the parameters setting, and evaluation tasks for comparing document lists. In Section 6.5, we first demonstrate empirically the benefits, for just-in-time retrieval, of separating users' information needs into multiple topically-separated queries rather than using a unique query. Then, we compare the proposed diverse merging technique with several alternative ones, showing that it outperforms them according to human judgments of relevance, and also exemplify the results on one conversation fragment given in Figure 6.2.

6.2 Motivation

Several diverse re-ranking methods have been previously proposed to create a concise list from the retrieval results of a single query. We will show in this chapter that diversification does not improve the results of such a multi-topic single query produced from a conversation fragment. For instance, a diverse re-ranking method applied on the retrieval results of a single query would suggest (for the example used in this chapter) the following irrelevant Wikipedia pages: "Cold Fire (Koontz novel)" or "Extended Cold Weather Clothing System" which are produced because of the mixture of independent topics in a single query. However, users would be more interested in "Igloo", "Shoe", "Lighter", "Flint Spark Lighter", and "Clothing".

Alternatively, the Round-robin merging method presents the best representative of the documents relevant to each implicit queries in the final list can be more helpful; however, its effectiveness is optimal when implicit queries have the same level of importance (Wu and McClean, 2007). For the example of this chapter, a merging method based on Round-robin would recommend the document "Die Hard" to answer an implicit query with a low level of importance, before suggesting enough documents like "Flint Spark Lighter" related to one of the implicit queries with a higher weight.

To improve over these approaches, we will use inspiration from extractive text summarization (Lin and Bilmes, 2011; Li et al., 2012) and from our diverse keyword extraction method proposed in Chapter 4. The merging method proposed here rewards at the same time topic similarity, to select the most relevant documents to the conversation fragment, and topic diversity, to cover the maximum number of implicit queries in a concise and relevant list of recommendations.

6.3 Definition of Diverse Merging Method

The diverse merging of retrieved document lists is the process of creating a short, diverse and relevant list of recommended documents which covers the maximum number of the main topics of each conversation fragment. The merging algorithm rewards diversity by decreasing the gain of selecting documents from a list of documents (retrieved for an implicit query) as the number of its previously selected documents increases. The method proceeds in two steps. We first represent queries and the corresponding list of candidate documents from the Apache Lucene search engine[1] using topic models, and then rank documents by using topical similarity and rewarding the coverage of different lists of documents. These steps are defined respectively in the following subsections.

6.3.1 Representation of Documents and Queries

We first learn a probability model for observing a word w in a document d through the set of abstract topics $Z = \{z_1, ..., z_{|Z|}\}$, using LDA (Blei et al., 2003) as implemented in the Mallet toolkit (McCallum, 2002). The hyperparameters of the model are initialized with $\alpha = 50/|Z|$, and $\beta = 0.01$ following Steyvers and Griffiths (2007) research and are then optimized using the Gibbs sampling also implemented in Mallet. The topic models can be obtained using PLSA as well, but we selected here LDA because it does not suffer from the over-fitting issue of PLSA (Blei et al., 2003). The topic-word distribution $p(w|z_k)$ and the document-topic distribution $p(z_k|d)$ show the contribution of a word w to the construction of a topic z_k. A distribution of topic z_k in a document d with respect to the other topics is finally inferred, using again Gibbs sampling implemented in Mallet.

We represent each new text or fragment t (e.g. from a conversation or document) by a set of probability distributions over all topics Z, noted as $P(t) = \{p(z_1|t), ..., p(z_k|t), ..., p(z_{|Z|}|t)\}$, where $p(z_k|t)$ is inferred using Gibbs sampling given the topic models previously learned. We associate to each new document d_j and query Q_i a set of topic probabilities according to the above definition noted respectively as $P(d_j) = \{p(z_1|d_j), ..., p(z_k|d_j), ..., p(z_{|Z|}|d_j)\}$ and $P(Q_i) = \{p(z_1|Q_i), ..., p(z_k|Q_i), ..., p(z_{|Z|}|Q_i)\}$.

6.3.2 Diverse Merging Problem

As stated above, our goal is to recommend a short ranked list of documents answering the users' information needs hypothesized in a conversation fragment, which are modeled by multiple topic-aware implicit queries (see Section 5.2). We build the final list of recommended documents by merging the document lists, one from each implicit query, with the objective of the maximum coverage of the main topics of the conversation fragment. Since each document list contains documents found by a search engine in response to an implicit query, which was prepared for one of the main topics of the conversation fragment, we merge the lists by selecting documents from the maximum number of lists in addition to maximizing their topical similarity to the conversation fragment. Our solution is formalized as follows.

We consider a set of implicit queries $Q_{implicit} = \{Q_1, ..., Q_M\}$, and the corresponding set of document lists $L = \{l_1, ..., l_M\}$ resulting from each query. M is the number of implicit queries

[1] Version (Lucene 4.2.1 API) is available in http://lucene.apache.org. We employed the Standard Analyzer for indexing, and TF-IDF similarity values between documents and queries for ranking document results.

of the fragment, and each l_i is a list of documents $\{d_1, ..., d_{N_i}\}$ which are retrieved for the query Q_i. We define the weight we_{l_i} of each list l_i as the weight of the implicit query Q_i (as defined in Section 5.2.2 on page 57) normalized over the sum of the weights of all implicit queries.

Moreover, we define the "collective query" Q which is made of the union of all implicit queries. This query is different from the single query which is used in Section 5.3.2 for preliminary retrieval experiments, because some of the keywords from the keyword set C appear in the single query but are not covered by any of the implicit queries, so they do not appear in the collective query. We associate to Q a set of probabilities over abstract topics, $P(Q) = \{p(z_1|Q), ..., p(z_{|Z|}|Q)\}$ as well.

The problem of diverse merging of lists amounts to finding a ranked subset of documents S from the union of all the documents retrieved for all the implicit queries, $\bigcup_{i=1}^{M} l_i$. The final result list S should contain documents which are the most representative of all the individual result lists l_i, and potentially the most informative with respect to the conversation fragment and the information needs that are implicitly stated.

6.3.3 Defining a Diverse Reward Function

The diverse merging problem is a form of the maximum coverage problem. Although this problem is NP-hard, it has been shown that a greedy algorithm can find an approximate solution guaranteed to be within a factor of $(1 - 1/e) \simeq 0.63$ of the optimal one if the coverage function is submodular and monotone non-decreasing, as first stated in Section 4.2.2 on page 36, where we also provided the definition of a monotone submodular function.

Several monotone submodular functions have been proposed in various domains for a similar underlying problem, such as explicit diverse re-ranking of retrieval results (Agrawal et al., 2009; Santos et al., 2010; Vargas et al., 2012), extractive summarization of a text (Lin and Bilmes, 2011; Li et al., 2012), or our own model of diverse keyword extraction from a conversation fragment presented in Chapter 4 of this thesis.

We aim to define a monotone submodular function for diverse merging of document lists inspired by our diverse keyword extraction method, in which we proposed a power function with a scaling exponent between 0 and 1 for diverse selection of keywords covering the maximum number of main topics of a text with a fixed number of items. Each text was represented by a set of abstract topics inferred using LDA. However, this method is not directly applicable for the problem of diverse merging of result lists, for the following reason. Suppose that there are two implicit queries and for each implicit query two document results are retrieved. The goal is to show two documents out of four documents to users. Suppose that the two documents which are retrieved for the first implicit query are highly related to two abstract topics but they have only one topic in common which is discussed in the conversation. If we apply diversity on the abstract topics which represent documents (like in the algorithm used for the diverse keyword extraction method), both of them can be selected by the algorithm to address topical diversity. The two documents are related to two main abstract topics but they answer only one of the main topics discussed in the conversation fragment, and therefore, the second main topic is not covered by the final document list.

To adapt this method to the problem of diverse merging, from the perspective of capturing users' information needs in the set of recommended documents, we define here a reward function enforcing the merging process to select documents from a diverse range of document lists. The proposed method covers the maximum number of document lists instead of the abstract topics which are used to represent documents and queries.

We start by estimating the cosine similarity of a subset of the documents in the final list S which are selected from the list l_i to the collective query Q (see Subsection 6.3.2) in topic space (Guo and Diab, 2012) as $r_{S,i}$:

$$r_{S,i} = \sum_{d \in S} 1_{(d \cap l_i)} \frac{\sum_{z_k \in Z} \{p(z_k|d) \cdot p(z_k|Q)\}}{\sqrt{\sum_{z_k \in Z} \{p(z_k|d) \cdot p(z_k|d)\}} \cdot \sqrt{\sum_{z_k \in Z} \{p(z_k|Q) \cdot p(z_k|Q)\}}} \tag{6.1}$$

In Equation 6.1, if the document $d \in S$ is chosen from the list l_i then it is counted in the summation otherwise it is not.

We then propose the following reward function f for any subset of the list S with relevant documents selected from l_i (results of implicit query Q_i), where we_{l_i} is the weight of the list l_i, and λ is a parameter between 0 and 1. This reward function is submodular because it has the diminishing returns property when $r_{S,i}$ increases.

$$f : r_{S,i} \to we_{l_i} \cdot r_{S,i}^{\lambda} \tag{6.2}$$

The set S is ultimately ranked by maximizing the cumulative reward function $R(S)$ over all the lists from all implicit queries, defined as follows:

$$R(S) = \sum_{i=1}^{M} we_{l_i} \cdot r_{S,i}^{\lambda} \tag{6.3}$$

The probability of selecting documents from the list of results for Q_i thus depends on we_{l_i}, the topical similarity of the query to the conversation fragment. This is in contrast to choosing the best representative document from the list of documents of each query, as in the Watson system, which does not select more documents for queries with higher weight before considering lower weight ones. Moreover, our model rewards diversity to increase the chance of choosing documents from all the document lists retrieved for implicit queries.

6.3.4 Finding the Optimal Document List

If $\lambda = 1$, the reward function ignores the diversity constraint, because it does not penalize multiple selections from the same list l_i and ranks documents only depending on their similarity to the collective query and the weights of implicit queries. However, when 0< λ <1, as soon as a document is selected from the list of results of an implicit query, other documents from the same list start having diminishing returns as competitors for selection. Decreasing the value of λ increases the impact of the diversity constraint on ranking documents, which augments the chance of recommending documents from other document lists.

Since $R(S)$ is a monotone submodular function, we propose the greedy algorithm shown in Algorithm 2 to maximize $R(S)$. In contrast to the greedy algorithm proposed in Section 4.2.7 on page 40 for the diverse keyword extraction problem in which keywords are distributed in all topics with a probability value, in this problem the documents in the lists of results are represented by a binary value (i.e. if they are included in the list the value is 1 otherwise it is 0). In the first step of the algorithm, S is empty. At each step, the algorithm selects a document d among the unselected documents from the union of all the result lists (i.e. $d \in (\bigcup_{i=1}^{M} l_i \setminus S)$), so that d has the maximum similarity to the collective query and also maximizes the coverage of the main topics of the conversation with respect to the previously selected documents in S. This coverage is defined as:

$$g(d,S) = \sum_{i=1}^{M} we_{l_i} \left(r_{S,i} + 1_{(d\cap l_i)} \frac{\sum_{z_k \in Z} [p(z_k|d) \cdot p(z_k|Q)]}{\sqrt{\sum_{z_k \in Z} [p(z_k|d) \cdot p(z_k|d)]} \cdot \sqrt{\sum_{z_k \in Z} [p(z_k|Q) \cdot p(z_k|Q)]}} \right)^{\lambda} \tag{6.4}$$

where the second term of the sum (starting with $1_{(d\cap l_i)}$) is the topical similarity of the document d included in the lists l_i to the collective query Q. The algorithm updates the set S by adding one of the documents $d \in (\cup_{i=1}^{M} l_i \setminus S)$ which maximizes $g(d,S)$. This procedure continues until reaching K documents from $(\cup_{i=1}^{M} l_i)$.

```
Input  : collective query $Q$, query set $Q_{implicit}$ of size $M$ with probabilities, set of
         weights $We$, set of lists of document results $L$ with probabilities, number of
         recommended documents $K$
Output : set of recommended documents $S$
$S \leftarrow \emptyset$;
while $|S| \leq K$ do
    $S \leftarrow S \cup argmax_{d\in((\cup_{i=1}^{M} l_i)\setminus S)} g(d,S)$
    where $g(d,S) = \sum_{i=1}^{M} we_{l_i} \cdot \{r_{S,i} + 1_{(d\cap l_i)} \frac{\sum_{z_k \in Z}[p(z_k|d)\cdot p(z_k|Q)]}{\sqrt{\sum_{z_k \in Z}[p(z_k|d)\cdot p(z_k|d)]}\cdot\sqrt{\sum_{z_k \in Z}[p(z_k|Q)\cdot p(z_k|Q)]}}\}^{\lambda}$;
end
return $S$;
```

Algorithm 2: Diverse merging of document results for recommendation.

6.3.5 A Toy Example

We provide a toy example to illustrate the above greedy algorithm. Suppose that for a conversation fragment we build two implicit queries, resulting in the corresponding document lists $l_1 = \{d_{11}, d_{12}\}$ and $l_2 = \{d_{21}, d_{22}\}$. Let us assume that the documents and queries are represented by three topics z_1, z_2, and z_3, and the topics are weighted as follows: $\beta_{z_1} = 0.4$, $\beta_{z_2} = 0.4$, and $\beta_{z_3} = 0.2$. Then the weights of lists can be inferred as follows: $we_{l_1} = 0.5$, and $we_{l_2} = 0.5$. A sample distribution of the topic weights for each document and the collective query is also shown in Table 6.1.

Table 6.1: Sample input to the greedy algorithm.

Texts	$p(z_1\|\cdot)$	$p(z_2\|\cdot)$	$p(z_3\|\cdot)$
Q	0.40	0.40	0.20
d_{11}	0.60	0.00	0.40
d_{12}	0.60	0.20	0.20
d_{21}	0.40	0.50	0.10
d_{22}	0.35	0.60	0.05

Table 6.2: The reward values $h(d,S)$ for $\lambda = .75$ and $g(d,S)$ for $\lambda = .75$ and $\lambda = 1$ calculated respectively using Algorithms 1 (on page 40) and 2 (on page 70) to select two documents out of four, from two lists of documents retrieved for two topically-separated implicit queries.

	$\lambda = .75$		$\lambda = 1$		$\lambda = .75$	
Documents	$h(\cdot,\emptyset)$	$h(\cdot,\{d_{22}\})$	$g(\cdot,\emptyset)$	$g(\cdot,\{d_{21}\})$	$g(\cdot,\emptyset)$	$g(\cdot,\{d_{21}\})$
d_{11}	0.373	0.767	0.370	0.858	0.399	0.890
d_{12}	0.452	0.794	0.452	0.940	0.463	**0.954**
d_{21}	0.474	**0.800**	**0.488**	–	**0.491**	–
d_{22}	**0.476**	–	0.466	**0.954**	0.475	0.812

For comparison purposes, we will run the algorithm for diverse keyword extraction with $\lambda = .75$ by substituting words with documents, and also run the algorithm proposed here for $\lambda = .75$ and $\lambda = 1$. The goal is to compare the selection of two documents out of four in terms of the coverage of the main topics of the conversation.

Initially S is empty. In Table 6.2, we provide the $h(d,S)$ values which are generated by the diverse keyword extraction method and by the method proposed here for all documents. The first algorithm selects d_{22} as the first document from l_2 and then selects d_{21} from the same list. Although these documents are covering the two main topics, they do not cover well the first implicit query, which is mostly about the first topic and not the mixture of them. In contrast, the merging algorithm (with both values of λ) adds d_{21} from l_2 as the first document in the final list S. Then the algorithm with $\lambda = 1$, which only considers the relevance of the documents to the collective query, selects d_{22} again from l_2. However, the algorithm with $\lambda = .75$, which considers the diversity of implicit queries, selects d_{12} from l_1 as the second document to show in the list S to users.

6.4 Data, Settings and Evaluation Method

The experiments were performed on conversational data from the ELEA Corpus (Emergent LEader Analysis, (Sanchez-Cortes et al., 2012)). Implicit queries are formulated as presented in Chapter 5, using the keywords extracted from each conversation fragment by the method proposed in Chapter 4. The lists of document results for each implicit query are obtained by submitting the query to the Apache Lucene search engine[2] over the English Wikipedia[3]. The initial lists of results are merged and/or re-ranked into a final list of recommended documents using several baseline methods, as well as the diverse merging method proposed in Section 6.3.4 above. This section presents the data, the framework of the system and parameters of the method, and also the evaluation techniques used in our experiments.

6.4.1 Conversational Corpus

The ELEA Corpus was introduced above in Section 5.3.1 on page 58. As we already evaluated the keyword extraction method over the ASR transcripts of the AMI Meeting Corpus and have shown that it is robust to ASR noise (Chapter 4), we only performed our experiments over manual transcripts of the ELEA Corpus here.

We use from the ELEA Corpus 5 conversations of around fifteen minutes each, which have been manually transcribed. Our data comprises 35 two-minute segments, each of them ending at a speaker change, like the data used in Chapter 5. We first use 9 conversation fragments out of 35 to set the parameters of our experiments. On average, these fragments contain 278 words including stop words. Once topic modeling is applied, the average number of main topics per fragment is 5, with an observed minimum of 3 and a maximum of 9. The remaining 26 fragments are used as a test set to evaluate the methods listed below.

6.4.2 Methods Compared in the Evaluation

The recommendation process starts by extracting a set of keywords, C, from the words recognized by the ASR system from the users' conversations as depicted in Figure 6.1. The keywords are extracted using the diverse keyword extraction technique that we proposed in Chapter 4. Then, implicit queries are formulated using this keyword set, following the two alternative approaches depicted in step 2 of Figure 6.1. In a simple model (right side of the figure), a single query is built for the conversation fragment using the entire keyword list as an implicit query. Conversely, in the approach we proposed in Chapter 5 (step 2, left side of the figure), multiple implicit queries are produced for the conversation fragment by clustering the keyword set into several topically-separated subsets, weighted based on the strength of the topic in the conversation fragment.

In step 3, we separately submit each implicit query to the Apache Lucene search engine over the English Wikipedia and obtain several lists of relevant articles. Finally, in step 4, we merge and re-rank these lists before recommendation. One baseline is the explicit diverse re-ranking technique proposed by Santos et al. (2010) for diversifying the primary search results retrieved

[2] Version (Lucene 4.2.1 API) available in http://lucene.apache.org.

[3] A local copy obtained using the Freebase Wikipedia Extraction(WEX) dataset Metaweb Technologies (version dated 2009-06-16) was downloaded from http://download.freebase.com/wex.

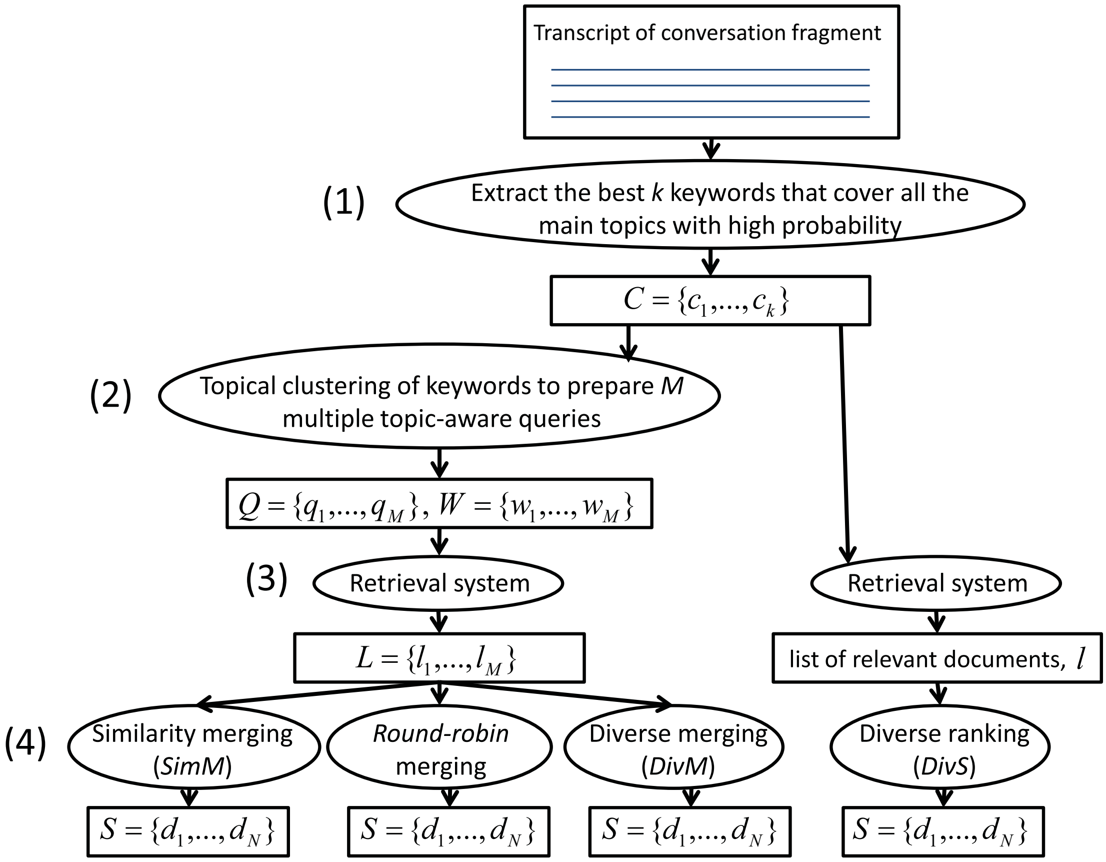

Transcript of conversation fragment
(1)
Extract the best k keywords that cover all the main topics with high probability
$C = \{c_1, ..., c_k\}$
(2)
Topical clustering of keywords to prepare M multiple topic-aware queries
$Q = \{q_1, ..., q_M\}, W = \{w_1, ..., w_M\}$
(3)
Retrieval system
Retrieval system
$L = \{l_1, ..., l_M\}$
list of relevant documents, l
(4)
Similarity merging (SimM)
Round-robin merging
Diverse merging (DivM)
Diverse ranking (DivS)
$S = \{d_1, ..., d_N\}$
$S = \{d_1, ..., d_N\}$
$S = \{d_1, ..., d_N\}$
$S = \{d_1, ..., d_N\}$

Figure 6.1: The four stages of our document recommendation approach (shown vertically as (1)–(4)) and the four options considered for comparative evaluation (shown horizontally at the bottom as *SimM*, *Round-robin*, *DivM*, and *DivS*).

for a single query, shown on the right side of Figure 6.1. To compare the methods, we adapted it to our system, when a single implicit query is built for a conversation fragment, by defining query aspects using the abstract topics employed for query and document representation. The method is noted *DivS* as it diversifies documents from a single list.

The method proposed in this chapter, noted *DivM*, appears at step 4. As shown on the left side of Figure 6.1, we merge the lists of documents retrieved for multiple implicit queries. We compare *DivM* with two other techniques. The first one, noted *SimM*, ignores the diversity of topics in the list of results and ranks documents only by considering their topic similarity to the conversation fragment. The second one is the *Round-robin* merging technique, used e.g. in the Watson just-in-time information retrieval system (Budzik and Hammond, 2000).

6.4.3 Parameter Settings for Experimentation

Since document search is performed over the English Wikipedia, we train our topic models on this corpus as well. We use only a subset of it for tractability reasons, i.e. about 125,000 articles as in other studies (Hoffman et al., 2010). The subset is randomly selected from the entire English Wikipedia. Like previous studies, we fix the number of abstract topics at 100 (Boyd-Graber et al., 2009; Hoffman et al., 2010).

The exponent of the submodular function is set to $\lambda = .75$, as in our diverse keyword extraction method (Chapter 4). This is found to be the best value for diverse merging of lists of results, as it leads to a reasonable balance between relevance and diversity in the aggregated list. Of course, if sufficient training data were available, it could be used to optimize λ.

The number of recommended documents is fixed at five. This is the value of the average number of topics in a conversation fragment, which allows the system to cover on average one result per topic. Experiments with other values were not carried out due to the cost of evaluation.

6.4.4 Method for Comparing Recommended Documents

We design four-choice "Human Intelligence Tasks" (HITs) that measure the relevance of recommended document lists for each of the test conversation fragment, following the approach described in Section 3.3 on page 23. The tasks require subjects to compare two lists obtained by two different methods. Here, we ask human judges to compare two lists of recommended documents in terms of their comprehensiveness, in addition to their relevance to the transcript of the conversation fragment. So we asked workers to select one of the ranking lists which contains relevant documents covering more main topics of the conversation fragment.

The 26 comparison tasks were crowdsourced via Amazon's Mechanical Turk. For each HIT we recruited 10 workers, only accepting those with greater than 95% approval rate and more than 1000 previously approved HITs. We only consider judgements from the workers who answered correctly our control questions about each HIT. Each worker was paid 0.20 USD per HIT. We observed that the average time spent per HIT was around 90 seconds.

To compute the *comparative relevance scores* over a large number of subjects and conversation fragments, we use the PCC-H qualification control method which was defined and validated in Section 3.4 on page 24. This method provides two scores, one for each document list that are compared, summing up to 100%; a higher value indicates a better list. In addition to PCC-H scores, we also provide the raw preference scores for each comparison, i.e. simply the number of times a system is preferred over another one in the comparison.

6.5 Experimental Results

We merge and re-rank the document lists – intended to be recommended during a conversation – generated by the four methods presented above in Section 6.4.2 and Figure 6.1. Three methods merge lists of results from topically-separated queries: *SimM* only considers their similarity with the fragment; *Round-robin* picks the best document in each list; and our proposal, *DivM*, considers the diversity and importance of topics. A fourth method, *DivS*, uses one query made of all keywords extracted from the conversation fragment, and ranks the documents using the diverse re-ranking technique proposed by Santos et al. (2010).

Binary comparisons were performed between pairs of techniques, using crowdsourcing over 26 conversation fragments of the ELEA Corpus. We aimed to minimize the number of binary comparisons while still ordering completely the methods according to their quality.

6.5.1 Diverse Re-ranking vs. Similarity Merging

We first perform a comparison between *DivS* and *SimM*. The PCC-H relevance score is 75% for *SimM* vs. 25% for *DivS*, as shown in Table 6.3. These values indicate the superiority of *SimM* over *DivS*. In other words, separating the mixture of topics of a fragment into multiple topically-separated queries mitigates the negative effect of the mixture of topics on the suggestions.

6.5.2 Comparison across Merging Techniques

Binary comparisons are then performed between pairs of merging techniques including *SimM*, *Round-robin*, and *DivM*. The PCC-H scores computed by the first approach are: 62% for *DivM* vs. 38% for *Round-robin*; 59% for *DivM* vs. 41% for *SimM*; and 56% for *Round-robin* vs. 44% for *SimM*, as shown in Table 6.3. The differences are confirmed by the PCC-H scores obtained by the second approach, according to the results in Table 6.3. The *SimM* method does not outperform the *Round-robin* method significantly, so both techniques are state-of-the-art ones. The reason is related to the dependency of these methods on the proportion of the number of recommended documents with respect to the number of implicit queries, which is discussed in more detail in the next section. The scores show that the diverse merging of lists of documents improves recommendations, and indicate the following high to low ranking: *DivM* > *Round-robin* > *SimM*.

SimM ranks lowest in this ordering, likely because of the ignorance of diversity in the list of results. *Round-robin* is second, likely because it disregards the major differences of importance among implicit queries in a conversation fragment. The results of the comparisons confirm that the *DivM* technique is the most satisfying to the majority of human subjects.

6.5.3 Impact of the Topical Diversity of Fragments

To further examine the benefits of the proposed method, *DivM*, we study its sensitivity to the number of topics in the conversation fragments. For this purpose, we divide the set of test fragments into two subsets. The first one (noted 'A' in Table 6.4) gathers the fragments for which fewer than or exactly five main topics (and therefore implicit queries) have been computed. The other fragments, with more than five main topics, form the second subset (noted 'B'). The value of five corresponds to the average number of main topics per fragment

Table 6.3: Comparative scores of the recommended document lists from four methods: *DivS*, *SimM*, *Round-robin*, and *DivM*, evaluated by human judges over the 26 fragments of the ELEA Corpus. The results imply the following ranking: *DivM* > *Round-robin* > *SimM* > *DivS*.

Compared methods	**Relevance (%)**					
(m_1 vs. m_2)	**PCC-H Score 1**		**PCC-H Score 2**		**Raw Score**	
	m_1	m_2	m_1	m_2	m_1	m_2
SimM vs. *DivS*	**75**	25	**.70±.08**	.32±.06	**70**	30
Round-robin vs. *SimM*	**56**	44	.48±.20	**.57±.21**	**52**	48
DivM vs. *Round-robin*	**62**	38	**.66±.07**	.46±.09	**58**	42
DivM vs. *SimM*	**59**	41	**.75±.07**	.47±.13	**58**	42

as well as to the number of allowed recommended documents in our experiments.

Table 6.4: Comparative scores of the recommended document lists from four methods: *DivS*, *SimM*, *Round-robin*, and *DivM*, evaluated by averaging human judges over two separate subsets of the ELEA Corpus. Subset A gathers fragments with fewer than or exactly five topics, while subset B gathers all the other fragments.

Compared methods	**PCC-H relevance score 1 (%)**			
	A		**B**	
(m_1 vs. m_2)	m_1	m_2	m_1	m_2
SimM vs. *DivS*	**80**	20	**70**	30
Round-robin vs. *SimM*	33	**67**	**68**	32
DivM vs. *Round-robin*	**64**	36	**60**	40
DivM vs. *SimM*	**54**	46	**60**	40

As shown in Table 6.4, although there is an improvement in the comparison scores of *DivS* over *SimM* when the number of conveyed topics in the fragments is higher than the number of allowed recommended documents (subset B vs. subset A), the comparison scores indicate the superiority of *SimM* over *DivS* in both cases, and confirm the benefit of the diverse merging techniques.

When comparing *Round-robin* versus *SimM*, the scores show the superiority of the former method when the number of conveyed topics in fragments is higher than the number of recommended documents, because it provides a diverse lists of documents in which documents relevant to less important topics are not displayed.

However, when the number of topics is smaller than the number of recommendations, *SimM* provides better results. The reason of the decrease in the scores of *Round-robin* is likely the ignorance of the actual importance of the main topics when ranking documents. Overall, as shown in Table 6.4, regardless of the number of topics conveyed in the fragments, *DivM* always outperforms *Round-robin* and *SimM*.

6.5.4 Example of Document Results

To illustrate how *DivM* surpasses the other techniques, we consider an example from one of the conversation fragments of the ELEA Corpus. The manual transcript of this conversation fragment is given in the Figure 6.2 on page 77.

A: okay I start.

B: how – how do you want to proceed?

A: I guess -

C: yes what is the most important?

A: I guess fire light.

B: fire lighter?

A: fire, yes. I would say if we had something we can fire with – I guess that the lighter is useful in getting some sparks.

B: hopefully.

A: so we can use either newspaper or – something like that.

C: but again - first it is more important to have enough err clothes.

A: and for me, more important to know where to go. I would say that the compass.

C: I mean – if you don't have enough clothes so – at one point you can –

B: you can die.

C: yes you can – you will die. so first issue, try to keep yourself alive and then you can –

A: but – but you already have some –

B: basics. you everything. you have enormous which is and so is no shoes here.

C: okay that we have shoes so – okay.

B: because seventy kilometers will take you how many days? err in the snow – what do you think?

A: two or three.

B: it can be two or three days?

C: yes, but okay you cannot always have fire with you – but you need always have clothes with you. I mean it is the only thing that protects you when you are walking.

B: oh yes. and erm you can make an igloo during the evening. not that cold. only about five degrees. so lighting a fire is not so important.

C: I guess fire is an extra. I mean it is important but err for me first it is important that when you keep walking you should be protected.

Figure 6.2: Sample transcript of a conversation fragment (speakers noted A through C) from the ELEA Corpus which was submitted to the document recommender system.

As described above, the conversation participants had to select a list of 12 items vital to survive in winter while waiting to be rescued. The keyword set extracted from the manual transcript of this fragment by the proposed diverse keyword extraction method (see Chapter 4) is C = {*fire, lighter, cloth, shoe, cold, die, igloo, walking*}. As our keyword extraction method was shown to be robust to ASR noise, we only use here the reference transcripts.

We display the topically-aware implicit queries prepared by our method from this keyword list along with the weights in Table 6.5. Each implicit query corresponds to one of the main topics of the fragment with a specific weight. In this example, the main topics spoken in the fragment are about “making an igloo”, “lightening a fire”, “having warm clothes”, and “suitable shoes for walking”.

Table 6.5: Examples of implicit queries built from the keyword list extracted from a sample fragment of the ELEA Corpus given in Figure 6.2. Each query covers one of the main topics of the fragment and has a different weight.

Implicit queries	**Weights**
Q_1 = {fire, cold, igloo, lighter}	$we_{l_1} = 0.332$
Q_2 = {shoe, lighter, walking}	$we_{l_2} = 0.293$
Q_3 = {cloth}	$we_{l_3} = 0.175$
Q_4 = {die}	$we_{l_4} = 0.120$
Q_5 = {igloo}	$we_{l_5} = 0.078$

Table 6.6: Examples of retrieved Wikipedia pages from the four different methods tested here. Results of diverse merging (*DivM*) appear to cover more topics relevant to the conversation fragment than other methods. The average ranking (*DivM* > *Round-robin* > *SimM* > *DivS*) is also observed in this example.

DivS	***SimM***	***Round-robin***	***DivM***
Flint spark lighter	Igloo	Igloo	Igloo
Extended Cold Weather Clothing System	Flint spark lighter	Shoe	Shoe
Cold Fire (Koontz novel)	Lighter	Jersey (clothing)	Flint spark lighter
Igloo	Lighter (barge)	Die Hard	Jersey (clothing)
Walking	Worcester Cold Storage Warehouse fire	Flint spark lighter	Lighter

In Table 6.6 we show the retrieval results (five highest-ranked Wikipedia pages) obtained by the four methods using the reference transcript of this fragment. *DivS* provides two irrelevant documents (Wikipedia pages) such as “Cold Fire (Koontz novel)”, likely because the single query does not separate the mixture of topics in the conversation fragment. *SimM* slightly improves the results by separating the discussed topics of the conversation fragment into multiple queries. However, it does not cover all the topics mentioned in the fragment due to mostly focusing on the single topic represented by Q_1. *Round-robin* further enhances the results by adding diversity, but as it gives the same level of importance to all topics, it provides a poor result like “Die Hard” from a topic of the conversation fragment with a small weight. The results of *DivM* appear to be the most useful ones, as they include other articles relevant to Q_1, Q_2, and Q_3 before showing results relevant to the low weight queries Q_4 and Q_5. Therefore, in this example, *DivM* provides better ranking of documents by covering the largest number of main topics mentioned in the fragment.

6.6 Conclusion

We proposed a diverse merging technique for combining lists of documents from multiple topically-separated implicit queries, prepared using keyword lists obtained from the transcripts of conversation fragments. Our diverse merging method *DivM* provides a short, diverse, and relevant list of recommendations, which avoids distracting participants that would consider it during the conversation. We also compared *DivM* to existing merging techniques, in terms of comprehensiveness and relevance of the final recommended list of documents to the conversation fragment. The human judgments collected via Amazon Mechanical Turk showed that *DivM* outperforms all other methods.

Moreover, these results emphasized the benefit of splitting the keyword set into multiple topically-separated queries: the suggested lists of documents from *DivS* (which accounts for the diversity of results by re-ranking the documents of a single list) were indeed found less relevant than those from *SimM* and the other two methods, which merged results from multiple queries.

In the following chapter, we will enable our system to answer explicit queries asked by users, considering contextual factors to improve the relevance of the answers, which will complement the recommendation functionality based on implicit queries.

7 Refinement of Explicit Queries

This chapter introduces a query refinement method applied to queries asked explicitly by users during a meeting or a conversation. The method thus adds a new functionality to a just-in-time retrieval system, building upon the methods presented in the previous chapters, and answering a set of attested user needs (Popescu-Belis et al., 2011). The method proposed for answering explicit queries does not require further clarifications from users, to avoid distracting them from their conversation, but leverages instead the local context of the conversation. The method first represents the local context by the keywords which are extracted from the transcript of the conversation using the diverse keyword extraction method proposed in Chapter 4 of this thesis. 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.

To evaluate our proposal, we built over the AMI Corpus a new dataset called AREX with sample queries accompanied by relevance judgments collected in a crowdsourcing experiment. We compare our query expansion approach with other methods, over the queries from the AREX dataset, showing the superiority of our method when either manual or ASR transcripts of the AMI Meeting Corpus are used.

7.1 Introduction

We specify in this chapter a query refinement technique for explicit queries addressed by users to a system during a conversation. Retrieval based on these queries can be erroneous, due to their inherent ambiguity. The proposed technique uses the local context of the conversation to properly answer the users' information needs, without the need for explicit query refinement. For instance, in the example discussed in Section 7.4.4, people are talking about the design of a remote control (see Figure 7.6 for the full transcript), and a participant expresses the need for more information about the acronym“LCD”. Our goal is to find the most helpful Wikipedia pages to answer this need in the context of designing a remote control.

Previous query refinement techniques – reviewed in Chapter 2, Section 2.5 on page 19 – enrich queries either interactively, or automatically, by adding relevant specifiers obtained 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. For example, the acronym "LCD" can be interpreted as "lowest common denominator" or "Lesotho Congress for Democracy", in addition to "liquid-crystal display", which is the correct interpretation in the context of our example. To address this issue, there are several techniques which have attempted to use the local context of users' activities, without requiring user interaction (Alidin and Crestani, 2013; Budzik and Hammond, 2000). However, as we will show, these are less suitable for a conversational environment, because of the nature of the vocabulary and the errors introduced by the ASR, such as 'recap' in the example below.

In this chapter, the local context of an explicit query is represented by a set of keywords automatically obtained from the conversation fragment preceding each query using the diverse keyword extraction technique proposed in Chapter 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. In order to evaluate the improvement brought by this method, we have constructed the AREX dataset (AMI Requests for Explanations and Relevance Judgments for their Answers). This dataset embodies a set of explicit queries inserted in several locations of the conversations from the AMI Meeting Corpus (Carletta, 2007), along with a set of human relevance judgments gathered over sample retrieval results from Wikipedia for each query, and an automatic evaluation metric based on Mean Average Precision (MAP).

The chapter is organized as follows. In Section 7.2, we describe the proposed query refinement method. Section 7.3 explains how the AREX dataset was constructed and specifies the evaluation metric. Section 7.4 presents and discusses the experimental results obtained both with ASR output and human-made transcripts of the AMI Meeting Corpus, showing the superiority of our technique over previous ones and its robustness with respect to unrelated keywords or ASR noise.

7.2 Content-based Query Refinement

Our goal in this chapter is to enable the just-in-time retrieval system proposed in this thesis to answer explicit queries formulated by users, in addition to spontaneous recommendations resulting from the implicit queries prepared by the system. The users can simply address the system by using a pre-defined unambiguous name, which is robustly recognized by the real-time ASR component (Garner et al., 2009). More sophisticated strategies for addressing a system in a multi-party dialogue context have been studied (Bohus and Horvitz, 2009; Wang et al., 2013), but they are beyond the scope of this chapter, which is concerned with the processing of the query itself by the system. As for the query results, once they are generated by the system, they can be displayed on a shared projection screen or on each user's device.

To answer an explicit query $Q_{explicit}$, the process of query refinement starts by modeling the local context using the transcript of the conversation fragment preceding the query. Conversation fragments have a fixed length for all queries. From each fragment, we extract a set of keywords $C = \{c_1, \cdots, c_k\}$ as the expansion terms, where k is the number of extracted keywords. The keywords are extracted using the diverse keyword extraction technique proposed in Chapter 4, because the diverse keyword extraction method maximizes the coverage of the main topics discussed in the conversation fragment while reducing the effect of ASR noise by the set of keywords. As the keyword set is related to several main topics, we weigh each expansion term by using a filter that assigns a weight we_{c_i}, with $0 \le we_{c_i} < 1$, to the term $c_i \in C \setminus Q$ based on its similarity to the explicit query.

To compute the weight for each expansion term, we first represent both query and the terms using topical information. $p(z|w)$ is the distribution of the topic z given an arbitrary word w from the dictionary. These topic distributions are created using the LDA topic modeling technique (Blei et al., 2003), implemented in the Mallet toolkit (McCallum, 2002) as we explained in Chapter 4, Section 4.2.1 on page 35. The topic models are learned over a large subset of the English Wikipedia with around 125,000 randomly sampled documents (Hoffman et al., 2010) as we did in the previous chapters. Moreover, we again fixed the number of topics at 100 (Boyd-Graber et al., 2009; Hoffman et al., 2010).

We then calculate the weight of each keyword w_{c_i} based on its normalized topical similarity to the explicit query, as formulated in the following equation:

$$we_{c_i} = \frac{\sum_{z\in Z} p(z|Q_{explicit})p(z|c_i)}{\sqrt{\sum_{z\in Z} p(z|c_i)^2}\sqrt{\sum_{z\in Z} p(z|Q_{explicit})^2}} \tag{7.1}$$

where Z is the set of abstract topics which correspond to latent variables, and $p(z|c_i)$ is the distribution of topic z in relation to the keyword c_i. Similarly, the averaged distribution of topic z in relation to the query $Q_{explicit}$ made of words $w_{ex,i}$ is calculated as follows:

$$p(z|Q_{explicit}) = \frac{1}{|Q_{explicit}|} \sum_{w_{ex,i}\in Q_{explicit}} p(z|w_{ex,i}) \tag{7.2}$$

Each query $Q_{explicit}$ is 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 may be relevant to one of the query aspects but not to the main topics of the fragment. We obtain a parametrized refined query $RQ(\gamma)$ (γ represents an exponent on the topical similarity values for each weight) which is defined as a set of weighted keywords, i.e. pairs of (word, weight), as shown below:

$$RQ(\gamma) = \{(w_{ex,1}, 1), \ldots, (w_{ex,|Q_{explicit}|}, 1), (c_1, we_{c_1}^{\gamma}), \ldots, (c_k, we_{c_k}^{\gamma})\} \tag{7.3}$$

In other words, the refined query contains the words from the explicit query with weight 1, plus the expansion keywords with a weight proportional to their topic similarity to the query.

The γ parameter has the following role. If $\gamma = \infty$, the refined query is the same as the initial explicit query (with no refinement) because the weights become zero. By setting γ to 0, the query is like the one used in the Watson system (Budzik and Hammond, 2000), 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 $\gamma = 1$ in our experiments, to weigh each keyword based on its relevance to the topics of the query. The value of γ could be optimized if more training data were available.

7.3 Dataset and Evaluation Method

Our experiments are conducted on the AREX dataset[1]. AREX stands for "AMI Requests for Explanations and Relevance Judgments for their Answers". The dataset contains a set of explicit queries, inserted at various locations of the conversations from the AMI Meeting Corpus (Carletta, 2007), as we will explain in Section 7.3.1. The dataset also includes relevance judgments gathered for each query using a crowdsourcing platform, over a set of documents retrieved for the four different methods described in Sections 7.3.2. These judgments will then be used as ground truth to evaluate a retrieval system automatically.

7.3.1 Explicit Queries in the AREX Dataset

The AMI Meeting Corpus contains 138 meetings on designing remote controls. Our dataset is made of a set of 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 generate a set of queries using the following procedure.

First, utterances containing an acronym *X* are automatically detected, because acronyms are one of the typical items which are likely to require explanations due to their potential ambiguity. Moreover, such utterances include the queries which appear naturally in the AMI Corpus. Of course, our query expansion technique is in fact applicable to any explicit query.

We formulate an explicit query such as "I need more information about *X*", and add it after these utterances. Seven acronyms, all-but-one broadly 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 scenario-based meetings of the AMI Corpus.

[1] See www.idiap.ch/dataset/arex

We use both manual and ASR transcripts of the fragments from the AMI Corpus in our experiments. The ASR transcripts are generated by the AMI real-time ASR system for meetings (Garner et al., 2009), which has an average word error rate (WER) of 36%. In addition, for experimenting with a variable range of WER values, we have simulated speech recognition mistakes as stated in Section 4.3.1 on page 42, by applying to the manual transcripts three different types of the ASR noise: deletion, insertion and substitution. The percentage of simulated ASR noise varies from 10% to 30%, as the best recognition accuracy reaches around 70% in conversational environments (Hain et al., 2010). However, noise is never applied to the explicit query itself.

7.3.2 Ground Truth Relevance Judgments

Following a classical approach for evaluating information retrieval (Voorhees and Harman, 2005), we build a reference set of retrieval results by merging the lists of the top 10 results from four different methods used to answer users' explicit queries. The retrieval results are obtained by the Apache Lucene search engine[2] over the English Wikipedia[3]. Three of the methods are listed in Sections 7.2. One is the method proposed in this paper, and the other two methods are baseline methods, one is the original query and the other one is implemented by the Watson just-in-time retrieval system. The fourth one builds a query which consists of only the keywords extracted from conversation fragments, with no words from the queries. We found that each explicit query (over 74) has at least 31 different results in the merged list, and therefore we decided to limit the reference set to 31 documents for each query.

Each fragment is about 400 words long, and this value was selected based on the following observation. We computed the sum of the weights assigned to the keywords extracted from each fragment by the $RQ(1)$ method, which weighs keywords based on their relevance to the query topics. Then we averaged them over 25 queries, randomly selected from AREX to serve as a development set for parameter tuning. The average weight 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 in these values for the last three values, we set fragment size to 400 words. We have also limited the weighting to the first 10 keywords extracted from each fragment following several previous studies (Carpineto and Romano, 2012) in which the typical number of expansion terms are in the rang of 10-30, thus speeding up the query processing.

We designed a set of tasks to gather relevance judgments from human subjects over the document set for each query. We showed to the subjects the transcript of the 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 documents from the reference document set.

[2] The used version (Lucene 4.2.1 API) is in http://lucene.apache.org.

[3] Version dated 2009-06-16 is available in http://dumps.wikimedia.org.

The subjects had to decide on the relevance value of each document by selecting one of the three options among 'irrelevant', 'somewhat relevant' and 'relevant' (these will be noted in the formulas below as $A = \{a_0, a_1, a_2\}$).

We collected judgments for the 74 queries of our dataset from 10 subjects per query. The tasks were crowdsourced via Amazon's Mechanical Turk, each judgment becoming a "Human Intelligence Task" (HIT). The average time spend per HIT was around 2 minutes. 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, in order to reduce the impact of "undecided" cases, inferred from the low agreement of the subjects. We computed the entropy of the judgment distribution H_{tj} as the measure of the uncertainty of subjects regarding the relevance of the document j to the query and the conversation fragment t (similar to the approach described in Section 3.4.2 on page 25) as follows:

$$H_{tj} = -\sum_{a \in A} \frac{s_{tj}(a) \ln(s_{tj}(a))}{\ln |A|} \tag{7.4}$$

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 was computed as $s'_{tj}(a) = s_{tj}(a) \cdot (1 - H_{tj})$, i.e. the raw score weighted by the subjects' uncertainty. These values are part of the distributed AREX dataset, along with the queries and pointers to fragments of the AMI Corpus and to Wikipedia pages.

7.3.3 Evaluation Using Ground Truth

Using the ground truth relevance of each document in the reference set, weighted by the subjects' uncertainty, we will measure the MAP score at rank k' of a candidate document result list. Then we will compare two lists of results using their corresponding MAP scores.

Scoring a List of Documents. 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).

$$gr_{tj} = \frac{s'_{tj}(a_1) + 2s'_{tj}(a_2)}{s'_{tj}(a_0) + s'_{tj}(a_1) + 2s'_{tj}(a_2)} \tag{7.5}$$

Then we compute the MAP score at the rank k'. The procedure starts by calculating $AveP_{tO}(k')$, the Average Precision at the rank k' for the conversation fragment t and the candidate list of results (output) of a system O, as follows:

$$AveP_{tO}(k') = \sum_{i=1}^{k'} P_{tO}(i) \triangle r_{tO}(i) \tag{7.6}$$

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

Finally, we compute $MAP_O(k')$, the MAP score at rank k' for a system O by averaging the Average Precision of all the queries at the rank k' as follows, where $|T|$ is the number of queries or fragments.

$$MAP_O(k') = \sum_{t=1}^{|T|} \frac{AveP_{t,O}(k')}{|T|} \tag{7.7}$$

Comparing two Lists of Documents. We compare two lists of documents obtained by two systems O_1 and O_2 through the percentage of the relative MAP improvement at the rank k', defined as follows:

$$\%RS_{O_1,O_2}(k') = \frac{MAP_{O_1}(k') - MAP_{O_2}(k')}{MAP_{O_2}(k')} \times 100. \tag{7.8}$$

This method emphasizes on the magnitude of the MAP score change at rank k' presented by percentage value. It measures the difference between the MAP score obtained by the proposed method O_1 and that of the baseline method O_2 normalized by the MAP score of the baseline system O_2. $\%RS_{O_1,O_2}$ is meaningful for non-zero denominators (i.e. $MAP_{O_2}(k') \neq 0$).

7.4 Experimental Results

We defined in Section 7.2 three methods for expanding queries based on the values of γ in Equation 7.3. The first method has $\gamma = \infty$ and is therefore noted $RQ(\infty)$ – it only uses explicit query keywords, with no refinement. The second one refines explicit queries using the method of the Watson system (Budzik and Hammond, 2000), with $\gamma = 0$, hence noted $RQ(0)$. The third method has $\gamma = 1$ and is noted $RQ(1)$ – this is the novel method proposed here, which expands the query with keywords from the conversation fragment based on their topical similarity to the query. Comparisons are performed over the human-made transcripts and the ASR output, using as a test set the remaining 49 queries not used for development.

7.4.1 Variation of Fragment Length

We study first the effect of the length of the conversation fragment on the retrieval results of the three methods, $RQ(1)$, $RQ(\infty)$, and $RQ(0)$. Keyword sets used for expansion are extracted here from the manual transcript of the conversation fragments preceding each query, and have a fixed-length per experiment. Although for the AREX dataset we considered 400-word fragments when building the reference document set, we vary the length below between 100 and 500 words.

The relative MAP scores of $RQ(1)$ over $RQ(\infty)$ for different ranks k' from 1 to 4 are provided in Figure 7.1, showing that although $RQ(\infty)$ is superior at $k' = 1$, $RQ(1)$ surpasses it for ranks 2, 3 and 4. The improvement over $RQ(\infty)$ slightly decreases when increasing the length of the conversation fragment, likely because of the topic drift in longer fragments. Indeed, when increasing the fragment length, the proposed method $RQ(1)$ behaves more similarly to $RQ(\infty)$ by assigning small weight values (close to zero) to the candidate expansion keywords.

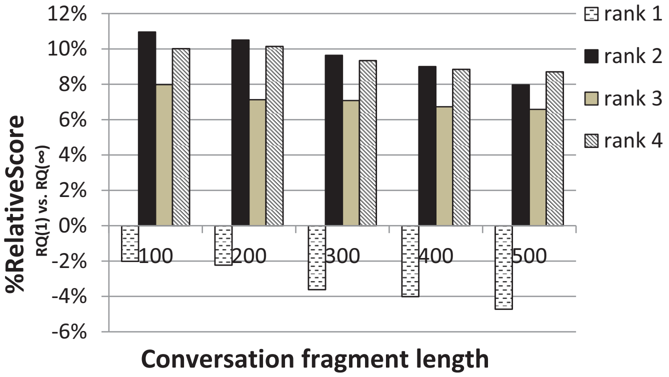


Figure 7.1: Relative MAP scores of $RQ(1)$ over $RQ(\infty)$ up to rank 4 obtained using manual transcripts with fragment lengths of 100, 200, 300, 400 and 500 words. $RQ(1)$ outperforms the $RQ(\infty)$ methods, except at rank $k' = 1$.

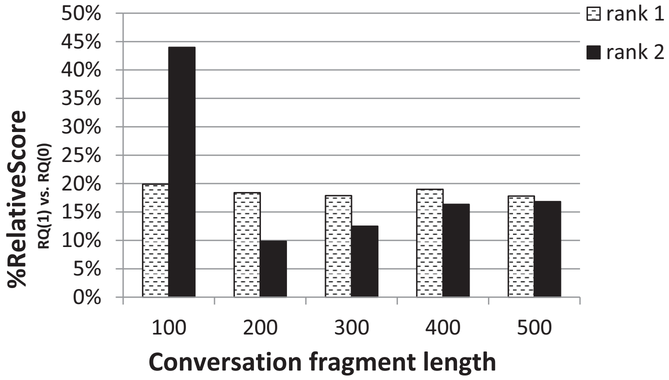


Figure 7.2: Relative MAP scores of $RQ(1)$ over $RQ(0)$ up to rank 2, obtained using manual transcripts with fragment lengths of 100, 200, 300, 400 and 500 words. $RQ(1)$ outperforms the $RQ(0)$ method.

The relative MAP scores of $RQ(1)$ over $RQ(0)$ are reported at ranks $k' = 1$ and $k' = 2$ in Figure 7.2. We do not report values for higher ranks, because of the lack of enough judgments for the retrieval results of RQ(0) among the reference set. The improvements over $RQ(0)$ at rank $k' = 1$ are approximately the same for different fragment lengths. They, nevertheless, vary a lot with the length of fragments when looking at rank $k' = 2$. The improvement is minimum at length 200 words, likely due to more relevant candidate expansion keywords at this length compared to the others. As shown above, the average sum of the weights of the expansion keywords is maximized by our method, $RQ(1)$, at length 200 words, so there are more keywords with topical similarity value closer to 1, an exponent of 0 or 1 over them makes no significant change. The improvement over $RQ(0)$ is increased by decreasing or increasing the length from 200 words at rank $k' = 2$. Probably because, the query topics are not completely covered, or the topics are changed , when the length increased or decreased from 200 words respectively. Thus the results show that $RQ(1)$ is more robust to out-of-topic keywords than $RQ(0)$.

7.4.2 Comparisons on Manual Transcripts

We now compare the proposed method $RQ(1)$ with two methods, $RQ(0)$ and $RQ(\infty)$ over the manual transcripts of the 49 conversation fragments with 400 words length preceding each query. We first provide the comparative relevance scores computed with the direct evaluation of list of document results by humans as proposed in Chapter 3. We design the comparative tasks as explained in Section 3.3. The subjects have to read the conversation transcript and the corresponding explicit query, answer a control question about its content, and then decide which of the two result lists (top 6 retrieved documents) contains more relevant documents, with the following options: the first list is better than the second one; the second is better than the first; both are equally relevant; or both are equally irrelevant.

Table 7.1: Comparative relevance scores at rank 6 of $RQ(1)$ vs. $RQ(0)$ (first line) and vs. $RQ(\infty)$ (second line), obtained over manual transcripts. The values show $RQ(1)$ surpasses the others.

Compared methods	**Relevance (%)**
$RQ(1)$ vs. $RQ(\infty)$	**58** vs. 42
$RQ(1)$ vs. $RQ(0)$	**59** vs. 41

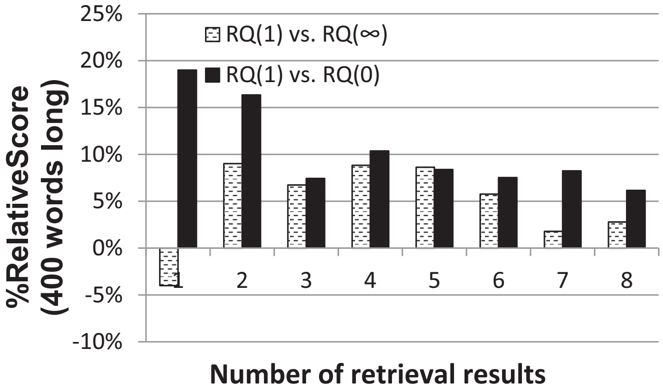


Figure 7.3: Relative MAP scores of $RQ(1)$ over the two methods $RQ(\infty)$ and $RQ(0)$ up to rank 8, obtained over the manual transcript of the 49 fragments with 400 words from the AMI Corpus. $RQ(1)$ surpasses both methods for ranks 2 to 8.

For these experiments, we recruited 10 human subjects via Amazon's Mechanical Turk crowdsourcing platform to perform each HIT, i.e. perform binary comparisons between the lists of top six documents generated by two pairs of systems. The scores obtained manually by this comparison method are shown in Table 7.1. The results confirm the superiority of $RQ(1)$ compared to the other two methods.

Then we compare the methods using the AREX dataset, for ranks k' from $k' = 1$ to $k' = 8$. The improvements obtained by $RQ(1)$ over the two others are represented in Figure 7.3 (the results for 400 words from Figures 7.1 and 7.2 are reused in this figure).

The relative MAP scores of $RQ(1)$ over $RQ(\infty)$, except at rank $k' = 1$, demonstrate the significant superiority of $RQ(1)$ over $RQ(\infty)$ (between 7% to 11%) up to rank $k' = 6$ on average. There are also on average small improvements around 2% over $RQ(\infty)$ at ranks $k' = 7$ 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 $k' = 1, 7$ and 8.

The relative MAP scores of $RQ(1)$ over $RQ(0)$ show significant improvements of more than 15% for ranks $k' = 1$ and $k' = 2$. Although the improvements decrease from rank 2, they remain considerably high at around 7%.

7.4.3 Comparisons on ASR Transcripts

We compare $RQ(1)$ with $RQ(\infty)$ and $RQ(0)$ over the ASR transcripts of the conversations, in order to consider the effect of ASR noise on the retrieval results of the expanded queries. We experiment 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%, as explained above, at the end of Section 7.3.1. We compute the average of the scores over five repetitions of the experiment with simulated ASR transcripts, which are randomly generated, and we 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 AREX dataset, because they do not overlap with the 31 documents obtained by pooling four methods.

First we compare 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. 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 EC), normalized by the sum of the weight value of all keywords used for the refinement of a query, as follows:

$$pn = \frac{\sum_{c_i \in (C \cap EC)} we_{c_i}^{\gamma}}{\sum_{c_i \in C} we_{c_i}^{\gamma}} \times 100\% \qquad (7.9)$$

We average these values over the 49 explicit queries and the five experimental runs with different random ASR errors. The results shown in Table 7.2 reveal that the proposed method, $RQ(1)$, is more robust to the ASR noise than $RQ(0)$. We also represent the relative scores of $RQ(1)$ over $RQ(0)$ in Figure 7.4. The improvement over $RQ(0)$ increases when the percentage of noise added to the fragments increases, and shows that our method exceeds *RQ*(0) considerably.

Table 7.2: Effect of ASR noise on the two query refinement methods *RQ*(1) and *RQ*(0) over the 49 explicit queries from our dataset, for a noise level varying from 10% to 30%. *RQ*(1) is clearly more robust to noise than *RQ*(0).

Average Percentage of the ASR noise added to queries (%)			
ASR noise	10%	20%	30%
RQ(1)	0.78	1.30	2.27
RQ(0)	5.64	12.07	21.07

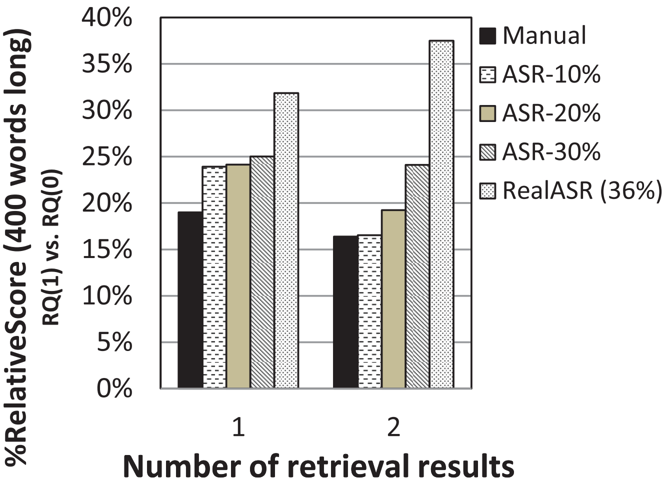


Figure 7.4: Relative MAP scores of *RQ*(1) vs. *RQ*(0) up to rank 2 obtained over the real and simulated ASR transcripts of the AMI Meeting Corpus. The results show the superiority of *RQ*(1) over *RQ*(0).

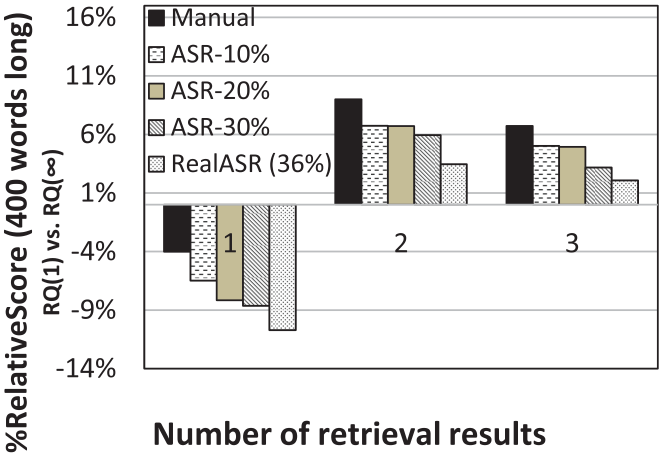


Figure 7.5: Relative MAP scores of *RQ*(1) vs. *RQ*(∞) up to rank 3 obtained over the real and simulated ASR transcripts of the AMI Meeting Corpus. The results show that *RQ*(1) outperforms the *RQ*(∞) method.

Moreover, we compare the retrieval results of $RQ(1)$ and $RQ(\infty)$ (which does not consider context) in noisy conditions, in Figure 7.5. 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.

7.4.4 Examples of Expanded Queries and Retrieval Results

To illustrate how $RQ(1)$ surpasses the other techniques, we consider an example from one of the queries of our dataset, using the ASR transcript of the conversation fragment given in Figure 7.6.The query is: "I need more information about LCD", in other words, it bears on the acronym "LCD". The list of keywords extracted for this fragment is the following one: C = {'interface', 'design', 'decision', 'recap', 'user', 'control', 'final', 'remote', 'discuss', 'sleek', 'snowman'}. Among these keywords, three of them ('recap', 'sleek', and 'snowman') correspond in fact to ASR noise.

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.
QUERY: I need more information about LCD.

Figure 7.6: A 150-word conversation fragment from the ASR transcripts of the AMI Meeting Corpus (segmented by the ASR into utterances) about designing a remote control. A query is inserted at the end of the fragment for the AREX dataset.

In this example, the proposed method, $RQ(1)$, assigns a weight of zero to keywords from ASR noise and to those unrelated to the conversation topics. Its corresponding expanded query is: $RQ(1)$ = {(lcd,1.0), (control,0.7), (remote,0.4), (design,0.1), (interface,0.1), (user,0.1)}.

$RQ(0)$ 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 more irrelevant words than the query for $RQ(1)$. Finally, $RQ(\infty)$ does not expand the query so it considers only 'lcd'.

The retrieval results up to rank 8 obtained for the three methods are displayed in Table 7.3. 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 mathematical function), 'LCD Soundsystem' (an American dance band), and 'Pakalitha

Table 7.3: Examples of retrieved Wikipedia pages (ranked lists) for a query on "LCD" using the three methods, with the conversational context shown in Figure 7.6. Results of *RQ*(1) are more relevant both to the query and to the 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

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 of conversation unrelated to the query or from ASR noise.

7.5 Conclusion

In this chapter, we proposed a query refinement technique which is applicable to explicit queries asked during a conversation. The method expanded queries based on the context of the conversation. We experimentally showed that the best method for contextual query refinement appears to be the proposed method *RQ*(1) over both manual and ASR transcripts. Although, *RQ*(∞) outperforms *RQ*(1) at rank $k' = 1$, the scores of *RQ*(1) show a significant improvement up to rank 8 over manual transcripts and up to rank 3 over ASR ones. Moreover, *RQ*(1) outperforms *RQ*(0) on both manually-made transcripts up to rank 8 and ASR transcripts up to rank 2. The scores also demonstrate that the proposed method *RQ*(1) is robust to various ASR noise levels and to the length of the conversation fragment used for expansion.

In addition, we built the AREX dataset accompanying these experiments which includes explicit queries added to the AMI Meeting Corpus with relevance judgments for the documents retrieved for these queries. The dataset can be used for future comparisons of conversational query-based retrieval systems.

8 System Integration and Scenario for User-based Evaluation

In this chapter, we present the implementation of the document recommender system based on the ideas proposed in Chapters 4, 5 and 6. We first describe the system architecture in Section 8.1 and state the programming techniques used for the implementation of the different modules of the system. In Section 8.2, we define a subjective evaluation task which we designed to measure the usability of the system. Finally, we present in Section 8.3 a pilot experiment that we ran to verify that the implemented system and the defined task appear to be suitable for user-centric evaluation. However, full-fledged user experiments are beyond the scope of the thesis, as we discuss in the final section of the chapter.

8.1 Overall Architecture

The document recommender system for conversations is made of several components that interact through a central module called the Connector Module (CM). The main components are the Speech-to-Text Module (ST), the Query Processor Module (PM), and the User Interface Module (UI). There are as many instances of the UI as participants to the conversation, so that each user can consult the recommendations on their own laptop. The architecture of the system that was implemented is represented in Figure 8.1. We present the modules in more detail below.

Connector Module (CM). The CM reads the file contains the transcript of users' conversation (obtained by the ST module described below) every minute, and sends it to Query Processor Module (PM). Once the execution of the PM is finished, the CM calls the UI to read the results of PM from files and then present them to users. CM also keeps track of all the previous conversation fragments and their results, so that users can go back in time and consult earlier results (possibly even after the end of the meeting).

Speech-to-Text Module (ST). The ST module provides the transcript of users' conversation as a text file. This module contains two components work sequentially: the SDK of the Microcone device, and the Automatic Speech Recognition (ASR) system. The speech signal is captured

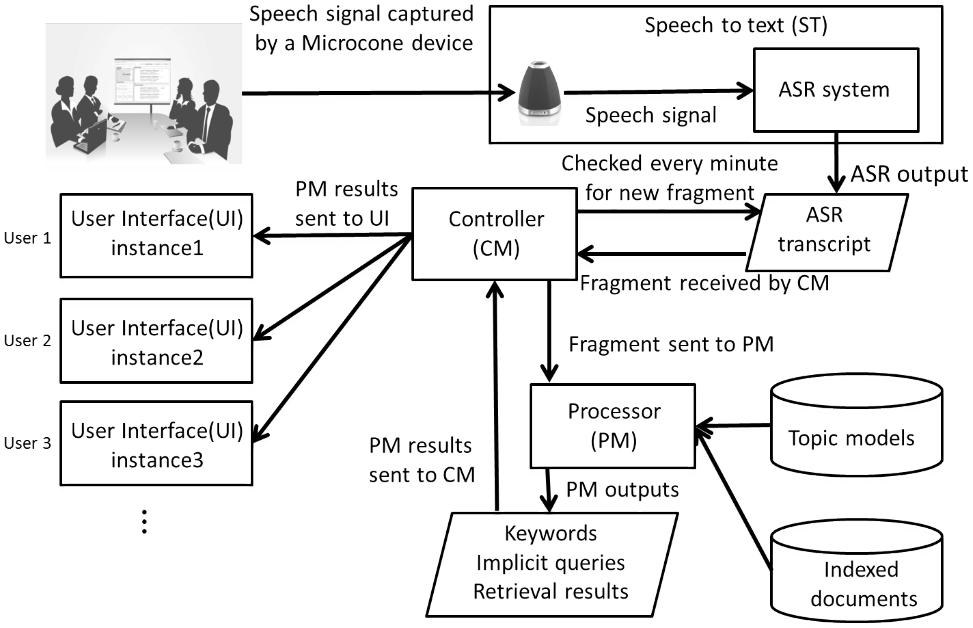


Figure 8.1: Architecture of the document recommender system.

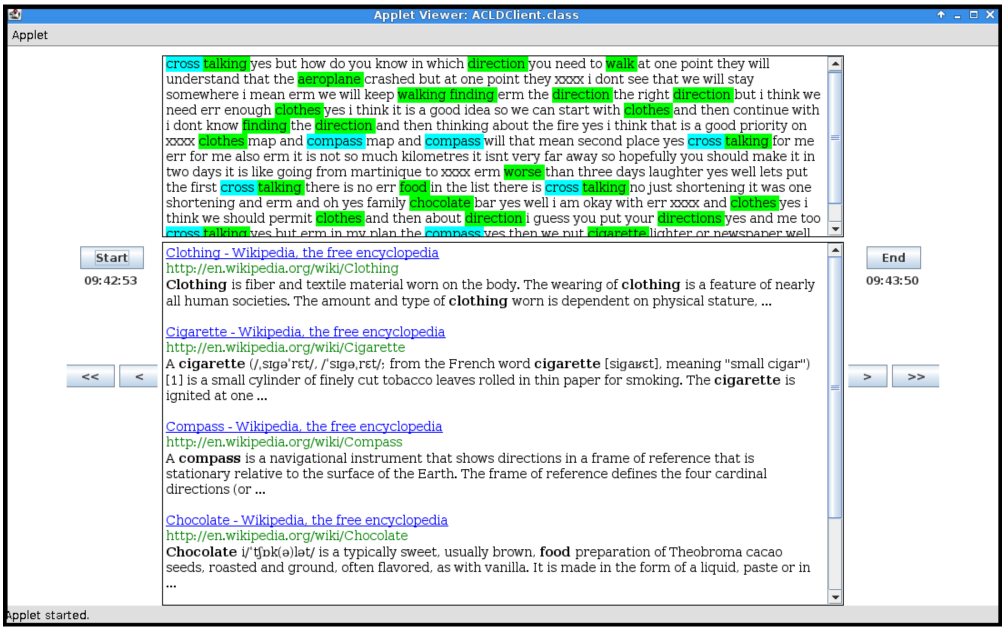
using a Microcone device, i.e. a microphone array with seven microphones (McCowan, 2012)[1]. It can capture and enhance speech signal using beamforming, and identify the direction of the speaker, which is mainly useful for recording meetings. The audio signals enhanced by the SDK of the Microcone device are sent to a real-time ASR system designed for meetings (Garner et al., 2009). The system processes the speech signal in batches using an integrated speech segmenter which relies on silences or minimal of the speech signal. The output of the ASR system is simply written to a file, which is read by the Connector Module and sent to the Processor Module.

Query Processor Module (PM). The PM receives the transcript of users' conversation at regular time intervals from the CM. The PM first extracts keywords from this transcript using the diverse keyword extraction technique proposed in Chapter 4, and writes this list of keywords to a file. Then it prepares implicit queries based on the approach proposed in Chapter 5 and submits each implicit query to the Lucene search engine over the English Wikipedia pages. The PM stores queries along with their weights and their retrieval results in another set of files. Finally, the PM merges the document results based on the method introduced in Chapter 6, and writes the final list to a file, which is used by the instances of the User Interface. The source code of the module is available at https://github.com/idiap/DocRec.

User Interface Module (UI). The UI is run on each participant's laptop, and communicates with the CM via the network. Thus, several instances of the UI can be created, so that each user in a meeting can have their own UI displayed on their laptop, and consult the recommen-

[1] See www.dev-audio.com. This product was developed from original Idiap research in the IM2 NCCR.

dations as they see fit. A snapshot of the UI within a meeting is depicted in Figure 8.2. Each UI displays the ASR transcript of the conversation fragment (to increase the understandability of results) as well as the extracted keywords, highlighted in green. Mainly, the list of recommended documents is displayed below the fragment, with hyperlinks to the full documents (Wikipedia pages) which can be opened in a browser by clicking on the link. Moreover, the first sentence of each document is displayed, marked with the keywords found in the transcript. When the user hovers the mouse over the link to the document, the UI shows the keywords relevant to each document in the transcript, highlighted in cyan. The UI allows users to launch or stop the recommendations, and to move forward or backward through the results by pressing specific buttons: the single arrows on the left and right sides allow respectively to move one conversation fragment backward or forward in time, while the left/right double arrows move to the beginning of the meeting, and, respectively, the “present” (latest fragment).



cross talking yes but how do you know in which direction you need to walk at one point they will understand that the aeroplane crashed but at one point they xxxx i dont see that we will stay somewhere i mean erm we will keep walking finding erm the direction the right direction but i think we need err enough clothes yes i think it is a good idea so we can start with clothes and then continue with i dont know finding the direction and then thinking about the fire yes i think that is a good priority on xxxx clothes map and compass map and compass will that mean second place yes cross talking for me err for me also erm it is not so much kilometres it isnt very far away so hopefully you should make it in two days it is like going from martinique to xxxx erm worse than three days laughter yes well lets put the first cross talking there is no err food in the list there is cross talking no just shortening it was one shortening and erm and oh yes family chocolate bar yes well i am okay with err xxxx and clothes yes i think we should permit clothes and then about direction i guess you put your directions yes and me too cross talking yes but erm in my plan the compass yes then we put cigarette lighter or newspaper well

Clothing - Wikipedia, the free encyclopedia
http://en.wikipedia.org/wiki/Clothing
Clothing is fiber and textile material worn on the body. The wearing of **clothing** is a feature of nearly all human societies. The amount and type of **clothing** worn is dependent on physical stature, ...

Cigarette - Wikipedia, the free encyclopedia
http://en.wikipedia.org/wiki/Cigarette
A **cigarette** (/ˌsɪɡəˈrɛt/, /ˈsɪɡəˌrɛt/; from the French word **cigarette** [siɡaʁɛt], meaning "small cigar") [1] is a small cylinder of finely cut tobacco leaves rolled in thin paper for smoking. The **cigarette** is ignited at one ...

Compass - Wikipedia, the free encyclopedia
http://en.wikipedia.org/wiki/Compass
A **compass** is a navigational instrument that shows directions in a frame of reference that is stationary relative to the surface of the Earth. The frame of reference defines the four cardinal directions (or ...

Chocolate - Wikipedia, the free encyclopedia
http://en.wikipedia.org/wiki/Chocolate
Chocolate i/ˈtʃɒk(ə)lət/ is a typically sweet, usually brown, **food** preparation of Theobroma cacao seeds, roasted and ground, often flavored, as with vanilla. It is made in the form of a liquid, paste or in ...

Figure 8.2: Snapshot of the user interface of the document recommender system, as seen by each user on their laptop. The transcript of the current fragment is at the top, and the list of recommended Wikipedia pages at the bottom. Here, the mouse was hovered over the link to the recommended document “Compass”, and as a result the system highlighted the words of the query that the system built for retrieving this document (cyan color).

Implementation. The CM and the UI modules are implemented on the Java language. However, based on the algorithms presented in previous chapters, the PM module is written in the Matlab language. To run all the modules together, we first prepared an executable file from the Matlab code of the PM, and then called this file from Java by designing a shell script. The output of each component is stored in a shared repository, which is made accessible using a URL address over our local computer network. Each instance of the UI module obtains all the necessary information to display from this repository found at the above URL. Thus, each user can run an instance of UI (also written in Java) on their own laptop, to use the results of the system, on condition that their laptop is connected to the local network.

8.2 Task Definition

The four participants are seated at a square table, as shown in Figure 8.3. The Microcone is placed in the middle of the table to capture users' voices, and is connected via USB to the main workstation running the CM, ST, and PM modules. Each participant is connected via the network to the recommender system, running the User Interface module on his/her personal laptop and receiving the recommendations.

Evaluating the document recommender system has major challenges because several factors need to be separated: quality and timeliness of recommendations, usability of the graphical user interface, utility of the recommendations with respect to the participants' task. Therefore, the evaluation will require a large number of subjects and conditions to separate all these factors. Here, we propose a simple scenario that could be used in such a set of tests.

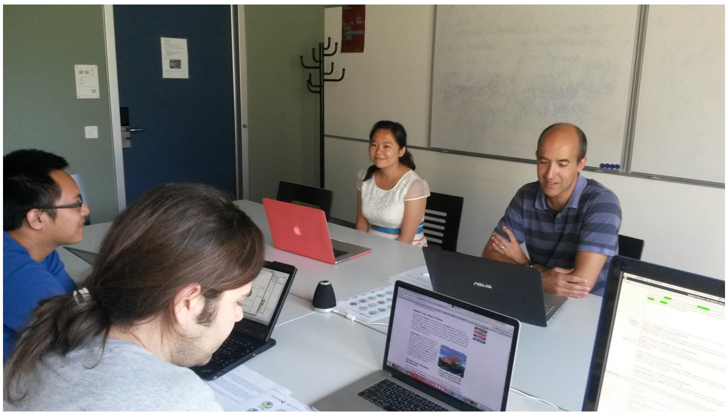


Figure 8.3: Setting of the pilot experiment. Four participants, each with their individual laptop running an instance of the UI, participate in a brainstorming meeting. The Microcone is visible at the center of the table, and the screen of the master workstation (running the CM, ST and PM modules) is partially visible in the lower right corner.

We consider teams of four people, as in the scenario of the AMI Meeting Corpus (Carletta, 2007). Each team is required to organize four lectures introducing scientific topics, for children aged 12–14, at a school festival. The team is given four one-hour slots. In each slot, one of the participants will present (using slides) some of the most attractive topics and innovations of a science. The group is asked to decide on the four scientific branches that they will present, and for each of them write down the title of the lecture and 3-4 bullet points indicating the main ideas they will present in the lecture. Each of the participants will be responsible for one lecture, but the contents should be decided jointly. For example, one would like to present some advances of "Physics", and therefore he/she could choose to talk about "nuclear energy", "solar system", "lasers and optic fibers" or "gravity law". Before actually holding the meeting with this task, we sent the following instructions by email to participants.

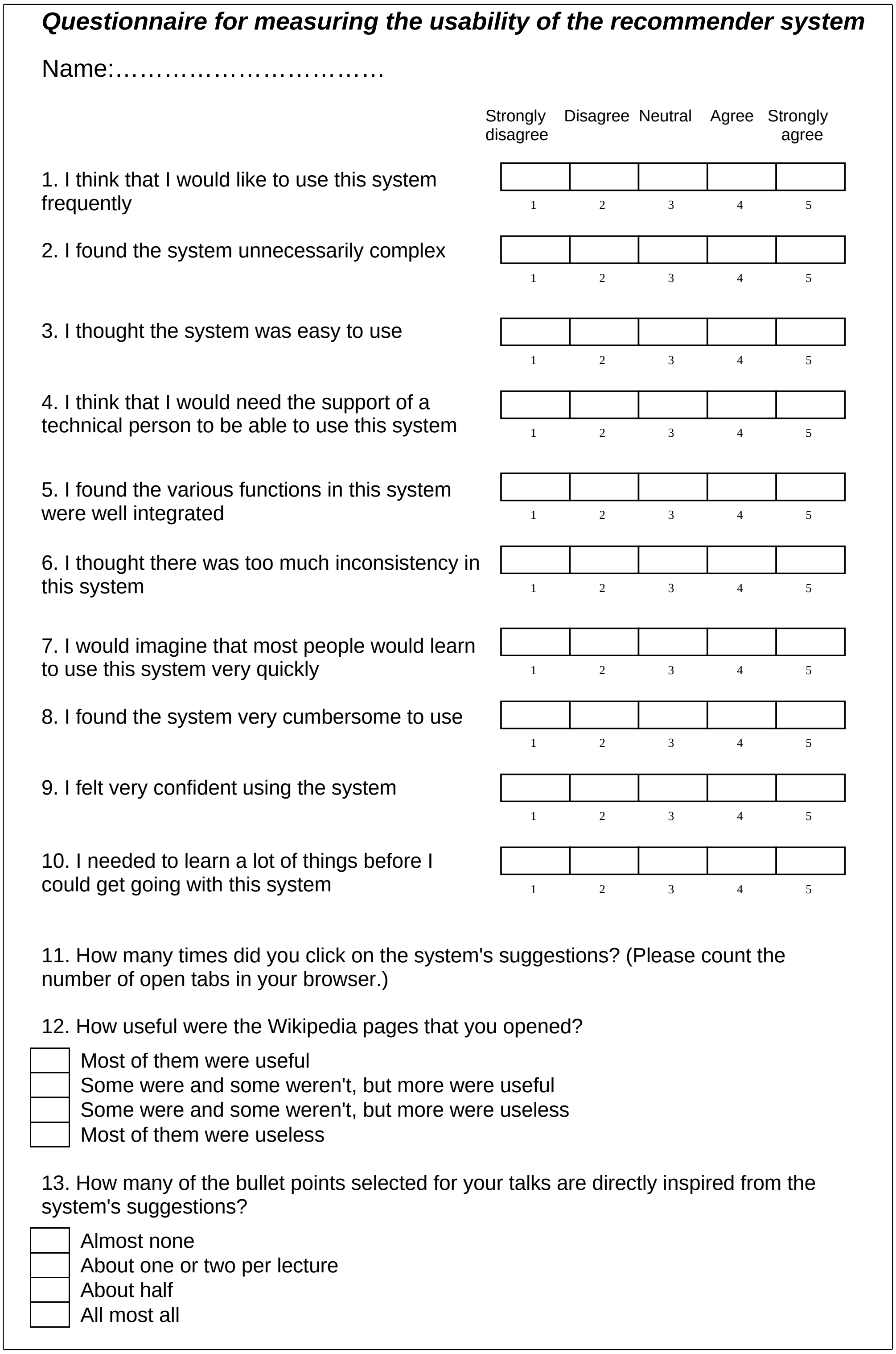
Figure 8.4: Images designating scientific branches, from a website with scientific topics for children (www.sciencekids.co.nz/topics.html). Each participant to the meeting was given a printed version to serve as initial inspiration for the task.

At the meeting, each participant is given two sheets. One is the instruction of the task, following the description above set in more concrete terms. The other one includes a set of images indicating several scientific branches, shown in Figure 8.4, as an inspiration for selection of one of the scientific branches.

At the end of the meeting, the participants are asked to fill in the evaluation questionnaire shown in Figure 8.5. The first ten questions of the questionnaire are intended to measure the usability of the system, and they were reproduced from Brooke (1996), who used them to evaluate helper applications for people with brain injury. The scores can be integrated and assessed using the well-known System Usability Scale (SUS). We added three more questions to measure the usefulness of the recommendations as well.

Regarding the first ten questions, we can extract from them a unique score demonstrating the percentage of the usability of our system. To calculate the unique score, we sum over the score contributions of ten questions. To compute the score contribution of each question, first the answers are quantified by assigning a numeric value to each answer based on their position ranging from 0 (strongly disagree) to 4 (strongly agree). For questions 1,3,5,7,and 9 the score contribution is the value of the position minus one. For questions 2,4,6,8 and 10, the contribution is five minus the value of the position. Then the sum of the scores is multiplied by 2.5 to obtain the overall percentage value of the system usability (Brooke, 1996).

Regarding question 11, we can simply report the average number of the clicked documents by participants. For questions 12 and 13, the numeric value of each answer is based on its position (0: high, 3: low). The contribution of question 12 is five minus the value of the



Questionnaire for measuring the usability of the recommender system

Name:……………………………

	Strongly disagree	Disagree	Neutral	Agree	Strongly agree
1. I think that I would like to use this system frequently	□ 1	□ 2	□ 3	□ 4	□ 5
2. I found the system unnecessarily complex	□ 1	□ 2	□ 3	□ 4	□ 5
3. I thought the system was easy to use	□ 1	□ 2	□ 3	□ 4	□ 5
4. I think that I would need the support of a technical person to be able to use this system	□ 1	□ 2	□ 3	□ 4	□ 5
5. I found the various functions in this system were well integrated	□ 1	□ 2	□ 3	□ 4	□ 5
6. I thought there was too much inconsistency in this system	□ 1	□ 2	□ 3	□ 4	□ 5
7. I would imagine that most people would learn to use this system very quickly	□ 1	□ 2	□ 3	□ 4	□ 5
8. I found the system very cumbersome to use	□ 1	□ 2	□ 3	□ 4	□ 5
9. I felt very confident using the system	□ 1	□ 2	□ 3	□ 4	□ 5
10. I needed to learn a lot of things before I could get going with this system	□ 1	□ 2	□ 3	□ 4	□ 5

11. How many times did you click on the system's suggestions? (Please count the number of open tabs in your browser.)

12. How useful were the Wikipedia pages that you opened?

□ Most of them were useful
□ Some were and some weren't, but more were useful
□ Some were and some weren't, but more were useless
□ Most of them were useless

13. How many of the bullet points selected for your talks are directly inspired from the system's suggestions?

□ Almost none
□ About one or two per lecture
□ About half
□ All most all

Figure 8.5: Evaluation questionnaire for the pilot experiment. The first part (questions 1–10) is the system usability scale, and is inspired from the one used by Brooke (1996), while the second part (questions 11–13) measures the utility of the document recommendations.

position, and that of question 13 is the value of the position plus two. The sum of the scores is multiplied by 10 to obtain again a percentage value of the utility of recommendations.

8.3 Observations from a Pilot Experiment

We performed a pilot experiment using four participants using the implemented system according to the above scenario. The main goal was to test that the proposed setting was functional and ready for large user study, which is beyond the scope of this thesis.

In the pilot experiment, the four users spent the first 20 minutes to determine the scientific branch about which each of them preferred to talk. Then, for each participant, 10 minutes were spent to discuss about the title of his/her lecture and 3–4 main topics of the lecture. The title of the lectures along with the main topics are shown in Table 8.1 for four participants. According to our observations about one third of the topics were obtained from the Wikipedia pages suggested by our system.

Table 8.1: Titles and contents of the talks designed during the pilot experiment by the four participants, using suggestions from our document recommender system.

Participant ID (Scientific branch)	Title of the lecture	Main topics chosen
A (Medicine)	“What doctors study to make you be healthy”	1-Different branches of medicine study different parts of the body 2-How vaccines work: weakened bacteria 3-X-ray / medical imaging
B (Physics)	“Principles of Physics”	1-Classical mechanics (object movement) 2-Solar system 3-Electricity 4-Electromagnetism
C (Math)	“How the math appear in our daily life?”	1-Logic rules 2-Probability rules 3-Mathematical thinking
D (Nature)	“Planting more trees for making a greener planet”	1-Climate harmonization 2-Preventing the global warming 3-Protection from natural disasters

The sample scores of system usability and recommendation utility obtained in this pilot experiment are shown in Table 8.2 for each participant. The average system usability and recommendation usability values over four participants are 65.6 and 77.5 respectively. Moreover, the average number of recommended Wikipedia pages that people opened in their browsers (number of open tabs at the end of the experiment) was 10 documents per participant. Of course, these are scores obtained through a single experiment, and serve to show how the evaluation settings and metrics can be used. To be interpretable, such numbers should be obtained from a large number of groups, which should compare, if possible, several experimental conditions.

Table 8.2: System usability and recommendation utility scores computed from questionnaires filled in by each participant, noted A through D. These are only sample scores, obtained from the pilot experiment, and serve to test the evaluation framework rather than provide reliable evaluation results.

Participant ID	System usability score	Recommendation usability score
A	65	90
B	57.5	80
C	72.5	70
D	67.5	70

8.4 Discussion

In this chapter, we implemented the ideas proposed in this thesis to generate relevant and diverse suggestions within a document recommender system for conversations. Moreover, we designed a new user interface to show the results along with some indications about why they are recommended through the highlighting of shared keywords.

We conducted a pilot experiment by defining a brainstorming scenario and setting up a meeting, in which four people decide on the title and main topics of their lectures about science to an audience of children. We proposed a way to measure the system's usability and the utility of recommendations, using questionnaires filled after the meeting.

In this chapter, we performed only a pilot evaluation, because a large-scale evaluation of the system needs a large pool of teams, which make the process of online evaluation difficult in terms of money and time required (Post et al., 2007). Moreover, such evaluations are often adapted to the comparison of two conditions, and do not produce absolute results about a system's quality. In fact, user-oriented design and evaluation of meeting support technology were intended as large topics of research in the IM2 NCCR and the AMI/AMIDA EU projects. The synthesis books published after the end of these projects (Renals et al., 2012; Bourlard and Popescu-Belis, 2013) show the difficulties of such evaluations and the need for more standardized evaluation methods and tasks.

9 Conclusions and Perspectives

In this thesis, we have defined a set of novel methods to improve the relevance and the diversity of documents suggested or retrieved by a just-in-time retrieval system designed for conversational environments. We have addressed the issue of maximizing the coverage of users' information needs in the presence of the potential multiplicity of topics and ASR noise within lists of document results. Moreover, we have aimed at generating concise list of documents instead of a long one, which needs more inspection from users, and also have minimized the interaction of users with the system to avoid distracting users from the main topics of their discussion, which is one of the main goals of such systems. To conclude, we summarize the main contributions of our thesis, and provide the future perspectives arising from these findings.

9.1 Summary of Contributions

We have presented a set of techniques to improve the relevance and diversity of the documents provided by a document recommender system during a meeting or a conversation. The system helps users to find relevant documents when they are reluctant to do a search during their conversation, by recommending a list of documents which are relevant to the topics of their conversation. Moreover, the system can automatically provide clarifications during the discussion in response to explicit queries, by disambiguating them using the conversational context.

First, we proposed an offline evaluation method based on crowdsourcing to compare the methods proposed in this thesis with state-of-the-art ones in Chapter 3. The evaluation method is affordable, as it does not need setting up a long series of meetings. To increase the reliability of the comparative scores obtained from crowdsourcing the task, we proposed a qualification control method. The method was applied both to individual judgments, to reduce the effect of those which disagree with the majority vote, and to entire tasks, to reduce the impact of undecided ones on the global scores. We concluded that the average of crowds' judgments which are passed through this qualification control filter is more robust to different designs of the task compared to other options. Moreover, the method does not need any prior knowledge to compute scores.

Using this evaluation method, we first compared several initial versions of the just-in-time retrieval framework which existed before this thesis in Chapter 3. We showed that submitting implicit queries made from dictionary-based keywords instead of the entire words of a fragment to a search engine provides documents which are more relevant to the content of the conversation. We also retrieved documents that answer queries explicitly asked by users. We compared the documents retrieved for these queries with those obtained by just-in-time retrieval from the conversation fragment preceding the query. The results demonstrated that the just-in-time retrieval system cannot be as accurate as the documents answering explicit queries. Therefore, we answered explicit queries using a separate module in addition to the part that provides recommendations based on the system's implicit queries.

As the queries prepared from keywords extracted from the conversation fragment lead to more relevant results than those made from the entire text, we developed a diverse keyword extraction method to represent users' information needs using a set of keywords extracted at regular time intervals from the ASR transcript of their conversation. The method maximizes the coverage of topics conveyed in the conversation fragment while minimizing the effect of ASR noise on the keyword set. We have first measured the topical diversity and relevance of keywords to the transcript of a conversation fragment in Chapter 4. We performed binary comparisons between the proposed method and several baselines based on word and topic frequency using the proposed evaluation method. To demonstrate the generality of our keyword extraction technique in terms of the level of representativeness of keywords, we performed our experiments over the manual and ASR transcripts of the AMI Meeting Corpus, in addition to the fragments that were artificially constructed as multi-topic from the Fisher Corpus. The results show the superiority of the proposed method compared to the baselines in terms of the representativeness of keyword sets, topical diversity with the highest α-NDCG value and additionally less number of words from ASR noise. As a secondary application, we utilized these keywords to represent the content of videos in a video recommendation setting.

We then compared the proposed diverse keyword extraction method with the baselines in terms of the documents that were retrieved when submitting the keyword sets as queries to a search engine in Chapter 5. We performed our comparisons (using again crowdsourcing) over the ELEA conversational corpus, due to its natural diversity in the conversations and the existence of Wikipedia articles relevant to the topics that are discussed. The results again showed that the documents retrieved for the keyword sets extracted by our method surpasses those from other methods.

Furthermore, we presented several policies for constructing implicit queries from the keyword sets extracted from conversation fragments. We prepared two types of queries for each conversation fragment, single queries vs. multiple queries. A single query is made of the entire keyword set, and carries various topics, while multiple queries are obtained by clustering the keyword set into topically separated subsets, each containing one main topic of the fragment. We compared these policies in terms of the relevance of the recommended documents to the content of a conversation again using the ELEA Corpus and the comparison method we proposed. The scores confirmed that the document results from multiple queries were preferable in comparison to those of single queries.

This achievement along with the goal of displaying a short list of recommended documents to the users compelled us to propose a diverse merging method to combine the lists of documents retrieved for multiple implicit queries in Chapter 6. The proposed method generates one concise and diverse list of documents for each fragment. The list is built by merging documents from the document lists retrieved for each implicit query, with the goal of maximizing the selection of documents from all the result lists while considering the significance of each query in terms of their relevance to the corresponding conversation fragment. The evaluation has been performed over the manual transcripts of the ELEA Corpus using our evaluation method. The results confirmed the superiority of preparing multiple queries instead of a single query for each conversation fragment, by showing that the diverse re-ranking methods do not improve the retrieval results of single queries. Moreover, the scores pointed out the superiority of the proposed diverse merging method compared to existing merging methods, including one used by another just-in-time retrieval system.

To address the need for a separate module that answers users' explicit spoken queries, without interrupting them for clarifications, we introduced a query refinement method which expands queries using the content of users' conversation in Chapter 7. The method represents the content using the keywords extracted from the conversation fragment preceding the query using the diverse keyword extraction method, selected for its representativeness and robustness to ASR noise. To evaluate our method, we built the AREX dataset which contains explicit queries added at different locations in the AMI Meeting Corpus, along with relevance judgments for a pool of documents allowing to compute MAP scores (in a modified way that supports non-binary judgments). The scores on both manual and ASR conversation fragments indicate the superiority of our query expansion method.

9.2 Future Directions

Future directions of research can extend the methods proposed in this thesis in several directions: using novel word representation methods, employing emotional information to detect when to interrupt users for recommendation, designing new user interfaces to improve the system's usability, and performing end-to-end evaluation of the system in context.

To improve the performance of the methods proposed in Chapters 4, 6, and 7, instead of using the LDA topic model with the bag-of-words assumption, our proposal can be extended with methods which include syntactic or hierarchical information to represent each fragment or a word within the fragment, in addition to semantic information. For instance, our proposal could be combined with methods that represent words by considering both short-range syntactic dependencies and long-range semantic dependencies, such as topical n-grams (Wang et al., 2007b) or paragraph vector representation (Le and Mikolov, 2014).

To reduce the effect of ASR noise, the users' information needs could be extracted directly from their speech signal or the ASR lattice, instead of the ASR transcript containing the first hypothesis of the ASR lattice. The implicit queries can be expanded with similar words instead of only using the words from users' conversation, taking advantage of the numerous existing techniques for word similarity (Yazdani and Popescu-Belis, 2013). Considering the order of the words in implicit queries instead of using them as the bag of words also has the potential to improve the retrieval results.

Quite naturally, the diverse merging method proposed in Chapter 6 can be applied to other settings as well. One interesting application is the diverse ranking of the results of an ambiguous query. Current diverse retrieval methods re-rank the first N documents retrieved for an ambiguous query. However, the first N document results may not cover all the aspects of the ambiguous query. One solution is to prepare several queries for each ambiguous query, each one addressing a single aspect of the query, which are obtained by expanding the ambiguous query based the corresponding aspect. Then diverse merging method can be applied so that in the final list of documents, the coverage of all the aspects of the query is maximized.

One of the important issues for a document recommender system is to determine the appropriate timing of the recommendations. In this thesis, we recommended documents every two minutes, at the end of an ongoing speaker turn, and considered as input all the words uttered during this time. A segment size of two minutes enabled us to collect an appropriate number of words (neither too small nor too large) in order to extract keywords, model the topics, and formulate implicit queries. Although it is possible to use verbal information to detect topic changes and perform online segmentation (Mohri et al., 2010), topic changes are not necessarily appropriate moments to make recommendations, because it would be useless to recommend documents about a topic that the users no longer discuss (Jones and Brown, 2004). Rather, it should be possible to analyze non-verbal information to detect emotional changes and recommend the results only after detecting specific emotions like "unconvincing", "confusing", "informativeness" and so on. Using verbal and non-verbal features within a cognitively-grounded model of "interruptibility" is a promising future research direction.

Furthermore, it is essential to adjust the number of documents recommended for every fragment. In this thesis, we performed our experiments over the first five or six top results to cover on average all implicit queries of a fragment within the list of documents. However, we have not considered the level of users' interest, to avoid annoying them with too many documents. Again, emotional analysis could serve here to adjust the number of recommendations based on the level of user "interest" in the discussion.

We believe that users do not wish to be overloaded with demands for clarification regarding their explicit queries. Thus, another future direction could be dynamic recognition of the situations in which it is necessary to ask users for further clarification instead of automatically inferring it from the context of users conversation. For instance, this is certainly needed when the query is truly ambiguous and the system could not properly disambiguate it from the users' conversation, as we have shown, in the presence of a high level of ASR noise (in that case, the MAP score over the results of the proposed query refinement method were very close to those of the non-refined method).

Besides, it should be possible to significantly assist users with moving through the previous fragments and their corresponding recommended documents faster and easier, for instance by using the navigation graph previously defined for the MUST-VIS lecture recommender system (Bhatt et al., 2013). Fragments of conversation and recommended documents would occupy the nodes of such a graph. Each fragment would be connected to other fragments and to the corresponding recommended documents using the edges of the graph. The explanations could appear as labels assigned to both nodes and edges, containing the summary of a fragment or a retrieved document, e.g. using a keyword cloud representation. The link from each fragment to a recommended document could moreover be labeled with the keywords of the implicit query for which they were retrieved. The evaluation of such a system would follow the path sketched in our last chapter.

The usability and the utility of the system should be measured with user-oriented experiments, having groups of subjects comparing the two systems: the system providing explanations using a navigation graph vs. the one not using it. The task which will be likely used can be a brainstorming meeting, e.g. for design or planning – a context in which document recommendations are particularly useful.

A Machine Translation Using Contextual Information

In this appendix, we will show that the diverse keyword extraction method proposed in Chapter 4 can be used for a different application than just-in-time retrieval. Indeed, the keywords can be used to improve the results of sentence-level machine translation (MT) by adding them as a representation of a sentence's context. In this case, the keywords are extracted from sentences that are adjacent to the source sentence in a text.

We propose to re-rank the N-best sentences in the target language obtained by the Moses statistical MT system based on their topical similarity to the source sentence augmented with the keywords. We compare our method with the baselines using a subjective evaluation. The results show a small improvement brought by our method, under certain conditions.

A.1 Introduction

One of the most popular approaches to MT is statistical MT, which learns the translation models from parallel corpora aligned at the sentence level, and translates the source sentences by the most probable target sentences obtained by these models. However, this approach ignores the document level information to select the best translation candidate. Several studies employ the document-level information to improve translation, either through monolingual topic models using a corpus in the source language, or through multilingual topic models using parallel document pairs.

However, because of the lack of enough parallel corpora at document level, the researchers mostly focused on capturing multilingual topics from comparable corpora at document level, which are more available. Nevertheless, the topic models extracted from comparable corpora are generally used to rank the potential translations of a word in context, instead of being used for translation at the sentence level, because they do not have the syntactic or grammatical information to translate sentences. In previous studies, the N-best results of a phrase-based statistical machine translation (SMT) system have been re-ranked using multilingual topical information. The candidate target sentences were re-scored based on the topical similarity defined among the topics of a target sentence and the entire source document (Tam et al., 2007). However, the weakness of this method is that there are many sentences in a document whose topics may be different from those of the entire document.

In the method presented here, we aim to utilize the information obtained by a phrase-based SMT system using parallel corpora at the sentence level, and in addition the contextual information obtained by bilingual topic models inferred from comparable corpora at the document level. Specifically, we expect to improve the quality of translations in the case of ambiguous words with various meanings.

In our model, we will first disambiguate source sentences by augmenting them with a set of keywords as the representatives of their local context. Each representative keyword is weighted based on its topical similarity to the source sentence. Then we re-rank the target sentences obtained by the Moses SMT system for each source sentence based on their topical similarity to the augmented source sentence. Topics are modeled using the polylingual topic model (Mimno et al., 2009) which is an extension of the Latent Dirichlet allocation method (LDA) for multilingual settings.

This appendix is structured as follows. In Section A.2, related work is reviewed, especially previous studies on multilingual topic models. In Section A.3, we describe our proposal for re-ranking translation candidates by employing the local context of a sentence in a document, represented through keywords. Section A.3.4 describes the data and evaluation method, followed by Section A.4 which presents the experimental results and their analysis.

A.2 Related Work

The phrase-based statistical machine translation approach provides state-of-the-art results. Phrase probabilities measure the co-occurrence frequency of a phrase pair, and are estimated from parallel corpora aligned at the phrase level (Zens et al., 2002; Koehn et al., 2003). However, this approach does not capture document-level constraints on the meaning and the relation of the words and phrases.

Several studies circumvent this deficiency by representing words or phrases using monolingual topic models (Su et al., 2012) or multilingual topic models obtained from parallel corpora aligned at document level (Zhao and Xing, 2008; Tam et al., 2007). Xiao et al. (2012) suppose that all the sentences in a document share the same topic with their document. Although this is true for many sentences, Xiong and Zhang (2013) have found that around 40% of sentences have topics different from those of their document obtained by experimenting over NIST MT03/05 datasets, and thus replaced the document topics with the topics of neighboring sentences. However, because of a relative insufficiency of parallel corpora for several language pairs and different domains, other approaches which depend on comparable corpora have obtained much interest.

To this end, multilingual topic models are inferred using comparable corpora aligned at document level (Ni et al., 2009; Mimno et al., 2009), but they were mostly utilized to identify the potential translations of a word in a document, instead of translating the entire sentence including the word, because the phrase-level information was not embedded in them. For instance, Vulić et al. (2011) ranked the potential word candidates in the target language based on the topical similarity between words in the source and target language. In further studies, Vulić and Moens augmented each word using the semantic information from the entire words of source and target vocabulary(Vulić and Moens, 2013a), or the contextual information defined by the co-occurrence words in a predefined context window (Vulić and Moens, 2013b), and then measured the similarity of words based on these semantic or contextual information.

However, the state-of-the-art MT systems (phrase-based SMT such as Moses) translate sentences by using sequences of phrases, instead of translating word by word. Therefore, Gong et al. (2011) re-ranked the N-best target candidate sentences obtained by a phrase-based SMT by scoring them based on the topical similarity of them with those of the entire source document. The topics were defined by a multilingual topic model as the extension of latent semantic analysis topic model (LSA) for multilingual applications (Tam et al., 2007). Nevertheless, there are sentences in a document, the topics of which are different from those of the entire document (Xiong and Zhang, 2013). To overcome this problem with the use of comparable corpora aligned at document level for the translation of sentences, in this chapter, we re-rank the N-best target candidate sentences based on their topical similarity with the augmented source sentences that are obtained by adding topically-relevant keywords from the source document. Moreover, each keyword is also weighted based on its topical similarity to the source sentence.

A.3 A Model for Sentence-Level Content-based Translation Using Comparable Corpora

The proposed MT system re-ranks the N-best translation candidates in the target language for each source sentence. The N-best candidates are obtained from a phrase-based baseline system (Moses) as described in Subsection A.3.1. We first represent words in both source and target languages using topical information obtained by a polylingual topic model (Mimno et al., 2009) as explained in Subsection A.3.2. The procedure of re-scoring target candidates then augments each source sentence with the content keywords that are relevant to the source sentence, which are extracted from the nearby sentences in the document. The amount of relevance of keywords to the source sentence is represented by a weight which is measured by computing the topical similarity between each keyword and the source sentence. Finally, we re-score and re-rank candidate target sentences based on their topical similarity with the augmented source sentence. The whole procedure is detailed in Subsection A.3.3.

A.3.1 Baseline Phrase-based Statistical Machine Translation

The phrase-based SMT models are frequently used and have state-of-the-art performance (Koehn et al., 2003). The phrase probability is computed from the co-occurrence frequency of a phrase pair in the phrase-aligned training data, and lexical probability is used to validate the quality of the phrase pair by checking how well its words are translated to each other.

According to the definition proposed by Koehn et al. (2003), the phrase-based translation model uses Bayes' rule to reformulate the translation probability for translating a source sentence into a target language as:

$$\arg\max_{f} p(f|e) = \arg\max_{f} p(e|f)p(f) \tag{A.1}$$

where $p(f)$ is the probability given by the language model and $p(e|f)$ is the probability given by the translation model. For each sentence in the source language, the method can provide a lattice of hypotheses with their probabilities, from which a ranked list which contains the N-best translation candidates in the target language can be derived. Here, we use Moses, an open-source phrase-based SMT system (Koehn et al., 2007), to generate the N-best target sentences for each source sentence.

A.3.2 Representing Words Using Multilingual Topic Models

We train a BiLDA topic model defined by Mimno et al. (2009), which is an extension of the standard LDA model (Blei et al., 2003) for bilingual purposes. The method assumes that the document pairs share the same distribution over topics. The probabilities $p(z_T|w_T)$ and $p(z_S|w_S)$ represent the distributions over the topic z of each word w in the target T and source S languages respectively.

For BiLDA topic model training, we use the implementation available in PolyLDA++[1] provided by Richardson et al. (2013). We set the hyper–parameters as $\alpha = \frac{50}{k}$ and $\beta = 0.01$ following Vulić et al. (2011), where k denotes the number of topics. We train the BiLDA topic model using Gibbs sampling with 1000 iterations.

A.3.3 Re-ranking the N-best Target Sentences Using Topical Information

We first extract the set of content words C from an interval Q which includes M sentences before and after the source sentence to be translated. We apply our diverse keyword extraction method (defined in Chapter 4 of this thesis) to determine C. The method maximizes the coverage of the topics of the selected interval with the keyword list C.

In addition, we weigh each extracted keyword $c_i \in C$ with a weight w_i, with $0 \leq w_i < 1$, based on the likelihood of observing the keyword c_i given the source sentence e, as formulated in

[1] https://bitbucket.org/trickytoforget/polylda

the following equation:

$$w_i = p(c_i|e) = \sum_{z_S \in Z_S} p(c_i|z_S) \cdot p(z_S|e) \tag{A.2}$$

where $p(z_S|e)$ is the average distribution of topic z in relation to the source sentence e, and $p(c_i|z_S)$ is the topic-word distribution calculated using the topic model. We define the augmented source sentence as follows:

$$e_{aug} = \{(e_1, 1), \cdots, (e_{|e|}, 1), (c_1, w_1), \cdots, (c_{|C|}, w_{|C|})\} \tag{A.3}$$

In other words, e_{aug} contains the words from the source sentence to be translated, e, with the weight 1, and the content words with a weight calculated in Equation A.2. We compute the score s_n for each target sentence f_n as follows:

$$s_n = p(f_n|e) = \sum_{z_T \in Z_T, z_S \in Z_S} p(f_n|z_T) p(z_T|z_S) p(z_S|e_{aug}) \tag{A.4}$$

In this equation, $p(z_T|z_S)$ is considered to be $1_{(z_T = z_S)}$, and $p(f_n|z_T)$ is the average distribution of topic z in relation to the target sentence f_n. We also compute $p(z_S|e_{aug})$ as follows:

$$p(z_S|e_{aug}) = \frac{1}{|e| + \sum_{i=1}^{|C|} w_i} \{\sum_{a \in e} p(z_S|a) + \sum_{i=1}^{|C|} w_i \cdot p(z_S|c_i)\} \tag{A.5}$$

Finally, the best candidate $\hat{f}$ for the source sentence e is computed by maximizing the following equation:

$$\hat{f} = \arg\max_{f_n} s_n \tag{A.6}$$

A.3.4 Data and Evaluation Method

We first describe the data and the evaluation method we used to assess our proposal, and then provide the results of our experiments and their analysis.

We trained the bilingual topic models, and also learned the translation and language models from the Europarl corpus. The European Parliament Proceedings Parallel Corpus (2011 release) is a corpus used for machine translation, extracted from the proceedings of the European Parliament (Koehn, 2005). It has versions in 21 European languages but we used only the English-French language pair.

For extracting topic models we used document aligned texts and for learning translation models we utilized sentence aligned texts. Although we trained topic models using a parallel corpus in our experiments, any comparable corpus can be used like Wikipedia articles. Following Mimno et al. (2009) who extract 400 topics from Europarl corpus, we set the number of topics to 400.

We obtained the N-best translation for each source sentence using the Moses SMT system (version 2.1.1 released in March 2014). We extracted five keywords using our diverse keyword extraction method from the five sentences before and five sentences after the source sentence and augmented this sentence with these keywords. We extracted the first 200 best translations of Moses and then re-ranked them based on the method proposed above.

As our method only contributes to improving the semantic of the translation sentences, not their syntax, and also to avoid skipping words which do not fit the topic models, we selected when re-ranking the 200 candidate sentences only the ones that have the same length as the 1-best translation candidate obtained by the Moses.

To evaluate the quality of the translations obtained using different approaches, BLEU score usually can be used. BLEU automatically calculates the precision score for each sentence translated by a machine translation system by comparing it against reference translation provided by human and then averages over all the sentences in the test set (Papineni et al., 2002). However, since the BLEU scores of all methods were very close, we performed subjective evaluation using an expert proficient in both French and English (in future work, several experts would increase the reliability of the conclusions). We used five documents from the test set (news-test-2013) from WMT '13 for performing subjective evaluation. The documents are about the following topics: voting rights and ID documents in the USA; taking or not the test for prostate cancer; the discovery of Higgs' boson; palliative care institutions in Canada; and an interview about the Paris Saint-Germain football team.

For evaluation, the expert looked at the sentences obtained by the three MT systems presented below, considering only the triples where at least one sentence differed from the others. The expert examined only the content (not the inflection) of the words which are different across the sentences, by comparing them with the words from the reference sentence and with regard to the words from the source sentence. The evaluation for each version is coded as follows. If the word(s) are identical to the reference word then a code value of 2 is assigned to them. If the word(s) are correct but not like the reference, then a code value of 1 is assigned to them. Otherwise the word(s) will be given 0 value. One possibility is to group together the values of 2 and 1, and count how many non-zero values were assigned to the words found different by the expert. However, keeping the three possible scores (2, 1 or 0), we assign a score to each translation method by counting the number of sentences with higher scores compared to those of the counterpart translation method (2 vs. 1 or 1 vs. 0). Thus, we can assess the improvement brought by one method.

A.4 Experimental Results

We compared our method – noted KC for Keyword-based Context – with two baseline methods. The first baseline is simply the translation obtained by the Moses system, noted M. The second method is similar to the proposed method here, but augments the source sentences using the entire words in 5 sentences before and after the source sentences instead of using content words. Then the method re-ranks the N-best results of the Moses, and selects the best one. This method is noted EC for Entire Context. We also compared our method with EC to emphasize the contribution brought by the diverse keyword extraction method in comparison with a simpler use of context for re-ranking MT hypotheses.

Table A.1: Comparison scores obtained using subjective evaluation over five different documents from test set. The compared methods are noted KC (the proposed method), M (the Moses baseline), and EC (method using all words of adjacent sentences as context). The numbers show the proportion of times one system is better than the other, hence the scores for "EC > M" and "EC < M" sum up to 100%. The results indicate the following ranking: EC < M < KC.

Compared Methods	Comparison values (%)					
	USA politics	Cancer test	Higgs' boson	Palliative care	PSG football team	Average over all documents
EC > M	43	67	22	50	52	46
EC < M	57	33	78	50	48	54
KC > M	50	67	40	50	50	52
KC < M	50	33	60	50	50	48
KC > EC	57	50	100	62	50	64
KC < EC	43	50	0	38	50	36

The results of our comparisons are provided for each of the five test documents in Table A.1. The numbers show the proportion of times one system is better than the other, hence the scores for "EC > M" and "EC < M" sum up to 100% (only different translations are counted). The average comparison values are 52% for KC vs. 48% for M, 46% for EC vs. 54% for M, and 36% for EC vs. 64% for KC. These results indicate the following ranking: EC < M < KC. The ranking shows that words from minor topics added by EC from the adjacent sentences can degrade the results of machine translation systems, while relevant keywords selected from the context by our method can improve the translation output.

There are a few cases in which the scores assigned to the sentences are zero by all three compared methods. In these cases, we examined the 200 best candidate translations and found out that among them there was no better translation to be selected. While it is possible that a better translation could be found below the 200 best ones, it is also likely that in many cases the translation model did not learn an appropriate phrase pair to use.

We provide two examples of the results in Table A.2. In the first example, KC outperforms M. In this example, the English word "charge" from the source sentence has two possible meanings in French, equivalent to "electric charge" and "accusation". The correct translation in this example is by "electric charge" which is indeed correctly selected by our method.

In the second example, we compare KC and EC translation results. In this example, the word "require" in the English source should be translated into "exigent" in French (third person plural of transitive verb "exiger"). However, the EC method translated it into "ont besoin" which means "need", which has the reverse meaning. The KC method translated it into "exiger",

Table A.2: Two examples of machine translation results: example (1) shows KC outperforms M, and example (2) shows the superiority of KC over EC.

Example (1)	
Source sentence	When, in fact, a particle having an electric **charge** accelerates or changes direction, ...
Reference sentence	Quand , en effet , une particule ayant une **charge** électrique accélère ou change de direction , ...
M translation	Lorsque , en fait , une **accusation** électriques particules avoir une légère modification direction , ...
KC translation	Lorsque , en fait , une **charge** électriques particules avoir une légère modification direction , ...

Example (2)	
Source sentence	The new election laws **require** voters to show a photo ID card and proof of US citizenship.
Reference sentence	Les nouvelles lois électorales **exigent** que les électeurs présentent une carte, d'identité avec photo et une preuve de citoyenneté américaine .
EC translation	Les nouvelles lois électorales **ont besoin** d'électeurs de montrer une photo de carte d'identité et la preuve de la citoyenneté américaine.
KC translation	Les nouvelles lois électorales **exiger** que les électeurs de montrer une photo carte d'identité et la preuve de la citoyenneté américaine .

which has the correct meaning, though not the correct mode/number/person.

A.5 Conclusion

We integrated the local context of the source sentence captured by our diverse keyword extraction method with the translation information obtained from the Moses SMT system. We used sentence-aligned parallel corpora for training the model used by the Moses and document-aligned parallel corpora for learning multilingual topic models. Unlike previous methods which require document-aligned parallel corpora to utilize contextual information for machine translation, our method can perform this using comparable corpora as well.

We showed that keywords selected by the diverse keyword extraction method from adjacent sentences and added to the source sentence can improve the results of machine translation. We also provided an example of the results obtained by our method and the baselines.

Bibliography

Rakesh Agrawal, Sreenivas Gollapudi, Alan Halverson, and Samuel Ieong. Diversifying search results. In *Proceedings of the Second ACM International Conference on Web Search and Data Mining*, pages 5–14. ACM, 2009.

Az Azrinudin Alidin and Fabio Crestani. Context modelling for Just-in-Time mobile information retrieval (JIT-MobIR). *Pertanika Journal of Science & Technology*, 21(1):227–238, 2013.

Omar Alonso and Ricardo Baeza-Yates. Design and implementation of relevance assessments using crowdsourcing. In *Proceedings of the European Conference on Information Retrieval (ECIR)*, volume 6611, pages 153–164, 2011.

Omar Alonso and Matthew Lease. Crowdsourcing 101: Putting the "wisdom of the crowd" to work for you. In *Tutorial at WSDM 2011 (4th ACM International Conference on Web Search and Data Mining)*, Hong Kong, China, 2011.

Omar Alonso, Daniel E. Rose, and Benjamin Stewart. Crowdsourcing for relevance evaluation. *SIGIR Forum*, 42:9–15, 2008.

Abu Shamim Mohammad Arif, Jia Tina Du, and Ivan Lee. Towards a model of collaborative information retrieval in tourism. In *Proceedings of the 4th Information Interaction in Context Symposium*, pages 258–261, 2012.

Abu Shamim Mohammad Arif, Jia Tina Du, and Ivan Lee. Examining collaborative query reformulation: a case of travel information searching. In *Proceedings of the 37th international ACM SIGIR conference on Research and development in information retrieval*, pages 875–878, 2014.

Javed A. Aslam and Mark Montague. Models for metasearch. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 276–284. ACM, 2001.

Rony Attar and Aviezri S. Fraenkel. Local feedback in full-text retrieval systems. *Journal of the ACM (JACM)*, 24(3):397–417, 1977.

Jing Bai, Dawei Song, Peter Bruza, Jian-Yun Nie, and Guihong Cao. Query expansion using term relationships in language models for information retrieval. In *Proceedings of the 14th ACM Int. Conf. on Information and Knowledge Management*, pages 688–695, 2005.

Chidansh A. Bhatt, Andrei Popescu-Belis, Maryam Habibi, Sandy Ingram, Stefano Masneri, Fergus McInnes, Nikolaos Pappas, and Oliver Schreer. Multi-factor segmentation for topic visualization and recommendation: the MUST-VIS system. In *Proceedings of the 21st ACM international conference on Multimedia*, pages 365–368. ACM, 2013.

Jagdev Bhogal, Andy Macfarlane, and Peter Smith. A review of ontology based query expansion. *Information Processing and Management*, 43(4):866–886, 2007.

Steven Bird. NLTK: the natural language toolkit. In *Proceedings of the COLING/ACL on Interactive presentation sessions*, pages 69–72, 2006.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

Dan Bohus and Eric Horvitz. Models for multiparty engagement in open-world dialog. In *Proceedings of the SIGDIAL 2009 Conference: The 10th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 225–234, 2009.

Hervé Bourlard and Andrei Popescu-Belis, editors. *Interactive Multimodal Information Management.* EPFL Press, Lausanne, 2013.

Jonathan Boyd-Graber, Jordan Chang, Sean Gerrish, Chong Wang, and David Blei. Reading Tea Leaves: How humans interpret topic models. In *Proceedings of the 23rd Annual Conference on Neural Information Processing Systems (NIPS)*, pages 288–296, 2009.

John Brooke. SUS-a quick and dirty usability scale. *Usability evaluation in industry*, 189(194): 4–7, 1996.

Jay Budzik and Kristian 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)*, pages 44–51, 2000.

Jamie Callan. Distributed information retrieval. In *Advances in information retrieval*, pages 127–150. Springer, 2000.

Jaime Carbonell and Jade Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 335–336. ACM, 1998.

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

Jean Carletta. Assessing agreement on classification tasks: The kappa statistic. *Computational Linguistics*, 22:249–254, 1996.

Claudio Carpineto and Giovanni Romano. A survey of automatic query expansion in information retrieval. *ACM Computing Surveys (CSUR)*, 44(1):1–56, 2012.

Claudio Carpineto, Renato De Mori, Giovanni Romano, and Brigitte Bigi. An information-theoretic approach to automatic query expansion. *ACM Transactions on Information Systems (TOIS)*, 19(1):1–27, 2001.

Ben Carterette and Praveen Chandar. Probabilistic models of ranking novel documents for faceted topic retrieval. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 1287–1296, 2009.

Asli Celikyilmaz and Dilek Hakkani-Tur. Concept-based classification for multi-document summarization. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5540–5543, 2011.

Praveen Chandar and Ben Carterette. Analysis of various evaluation measures for diversity. In *Proceedings of the DDR workshop*, pages 21–28. Citeseer, 2011.

Paul-Alexandru Chirita, Claudiu S. Firan, and Wolfgang Nejdl. Personalized query expansion for the web. In *Proceedings of 30th annual int. ACM SIGIR conf. on Research and development in IR*, pages 7–14, 2007.

Gokul Chittaranjan, Oya Aran, and Daniel Gatica-Perez. Exploiting observers' judgments for nonverbal group interaction analysis. In *Proceedings of the IEEE Conference on Automatic Face and Gesture Recognition (FG)*, 2011.

Christopher Cieri, David Miller, and Kevin Walker. The Fisher Corpus: a resource for the next generations of speech-to-text. In *Proceedings of 4th International Conference on Language Resources and Evaluation (LREC)*, pages 69–71, 2004.

Charles L. Clarke, Nick Craswell, and Ian Soboroff. Overview of the TREC 2009 web track. Technical report, DTIC Document, 2009.

Charles L. A. Clarke, Maheedhar Kolla, Gordon V. Cormack, Olga Vechtomova, Azin Ashkan, Stefan Büttcher, and Ian MacKinnon. Novelty and diversity in information retrieval evaluation. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 659–666, 2008.

Andras Csomai and Rada Mihalcea. Linking educational materials to encyclopedic knowledge. In *Proceedings of the 2007 conference on Artificial Intelligence in Education: Building Technology Rich Learning Contexts That Work*, pages 557–559, 2007.

Mary Czerwinski, Susan Dumais, George Robertson, Susan Dziadosz, Scott Tiernan, and Maarten Van Dantzich. Visualizing implicit queries for information management and retrieval. In *Proceedings of the SIGCHI conference on Human factors in computing systems (CHI)*, pages 560–567, 1999.

Fernando Diaz and Donald Metzler. Improving the estimation of relevance models using large external corpora. In *Proceedings of 29th annual int. ACM SIGIR conf. on Research and development in IR*, pages 154–161, 2006.

Susan Dumais, Edward Cutrell, Raman Sarin, and Eric Horvitz. Implicit queries (IQ) for contextualized search. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 594–594, 2004.

Eibe Frank, Gordon W. Paynter, Ian H. Witten, Carl Gutwin, and Craig G. Nevill-Manning. Domain-specific keyphrase extraction. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI 1999)*, pages 668–673, 1999.

Philip N. Garner, John Dines, Thomas Hain, Asmaa El Hannani, Martin Karafiát, Danil Korchagin, Mike Lincoln, Vincent Wan, and Le Zhang. Real-time ASR from meetings. In *Proceedings of Interspeech*, pages 2119–2122, 2009.

John S Garofolo, Cedric GP Auzanne, and Ellen M Voorhees. The trec spoken document retrieval track: A success story. *NIST SPECIAL PUBLICATION SP*, 500(246):107–130, 2000.

Zhengxian Gong, Guodong Zhou, and Liangyou Li. Improve SMT with source-side topic-document distributions. In *MT Summit*, pages 496–501, 2011.

Weiwei Guo and Mona Diab. Modeling sentences in the latent space. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 864–872. Association for Computational Linguistics, 2012.

Maryam Habibi and Andrei Popescu-Belis. Query refinement using conversational context: a method and an evaluation resource. In *Proceedings of the NLDB 2015 (20th International Conference on Applications of Natural Language to Information Systems*, 2015a.

Maryam Habibi and Andrei Popescu-Belis. Using crowdsourcing to compare document recommendation strategies for conversations. In *Workshop on Recommendation Utility Evaluation: Beyond RMSE (RUE 2011)*, pages 15–20, 2012.

Maryam Habibi and Andrei Popescu-Belis. Diverse keyword extraction from conversations. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 651–657, 2013.

Maryam Habibi and Andrei Popescu-Belis. Enforcing topic diversity in a document recommender for conversations. In *Proceedings of the 25th International Conference on Computational Linguistics (Coling)*, pages 588–599, 2014.

Maryam Habibi and Andrei Popescu-Belis. Keyword extraction and clustering for document recommendation in conversations. ACM/IEEE Transaction on Audio, Speech and Language Processing, 2015b.

Thomas Hain, Lukas Burget, John Dines, Philip N. Garner, Asmaa El Hannani, Marijn Huijbregts, Martin Karafiat, Mike Lincoln, and Vincent Wan. The AMIDA 2009 meeting transcription system. In *Proceedings of INTERSPEECH*, pages 358–361, 2010.

Peter E. Hart and Jamey Graham. Query-Free information retrieval. *International Journal of Intelligent Systems Technologies and Applications*, 12(5):32–37, 1997.

David F. Harwath and Timothy J. Hazen. Topic identification based extrinsic evaluation of summarization techniques applied to conversational speech. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5073–5076, 2012.

David F. Harwath, Timothy J. Hazen, and James R. Glass. Zero resource spoken audio corpus analysis. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8555–8559, 2013.

Timothy J. Hazen. Topic identification. In Gokhan Tur and Renato De Mori, editors, *Spoken language understanding: Systems for extracting semantic information from speech*, chapter 12, pages 319–356. John Wiley & Sons, 2011a.

Timothy J. Hazen. Latent Topic Modeling for audio corpus summarization. In *Proceedings of the 12th Annual Conference of the International Speech Communication Association*, pages 913–916, 2011b.

Monika Henzinger, Bay-Wei Chang, Brian Milch, and Sergey Brin. Query-free news search. *World Wide Web: Internet and Web Information Systems*, 8(2):101–126, 2005.

Matthew D. Hoffman, David M. Blei, and Francis Bach. Online learning for Latent Dirichlet Allocation. In *Proceedings of 24th Annual Conference on Neural Information Processing Systems*, pages 856–864, 2010.

Thomas Hofmann. Probabilistic latent semantic analysis. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pages 289–296. Morgan Kaufmann Publishers Inc., 1999.

Anette Hulth. Improved automatic keyword extraction given more linguistic knowledge. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2003)*, pages 216–223, 2003.

Birong Jiang, Endong Xun, and Jianzhong Qi. A domain independent approach for extracting terms from research papers. In *Databases Theory and Applications*, pages 155–166. Springer, 2015.

Qianli Jin, Jun Zhao, and Bo Xu. Query expansion based on term similarity tree model. In *Proceedings of the International Conference on Natural Language Processing and Knowledge Engineering*, pages 400–406, 2003.

Gareth J.F. Jones and Peter J. Brown. Context-aware retrieval for ubiquitous computing environments. In *Mobile and ubiquitous information access*, pages 227–243. Springer, 2004.

David R. Karger, Sewoong Oh, and Devavrat Shah. Budget-optimal crowdsourcing using lowrank matrix approximations. In *Proceedings of the Allerton Conference on Communication, Control and Computing*, 2011.

Faiza Khan Khattak and Ansaf Salleb-Aouissi. Quality control of crowd labeling through expert evaluation. In *Proceedings of the Second Workshop on Computational Social Science and the Wisdom of Crowds at NIPS*, 2011.

Philipp Koehn. Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5, pages 79–86. Citeseer, 2005.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 48–54. Association for Computational Linguistics, 2003.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions*, pages 177–180. Association for Computational Linguistics, 2007.

Victor Lavrenko and W Bruce Croft. Relevance based language models. In *Proceedings of 24th annual int. ACM SIGIR conf. on Research and development in IR*, pages 120–127, 2001.

Quoc V. Le and Tomas Mikolov. Distributed representations of sentences and documents. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, pages 1188–1196, 2014.

Jingxuan Li, Lei Li, and Tao Li. Multi-document summarization via submodularity. *Applied Intelligence*, 37(3):420–430, 2012.

Hui Lin and Jeff Bilmes. A class of submodular functions for document summarization. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 510–520, Portland, OR, 2011.

Feifan Liu, Deana Pennell, Fei Liu, and Yang Liu. Unsupervised approaches for automatic keyword extraction using meeting transcripts. In *Proceedings of the 2009 Annual Conference of the North American Chapter of the ACL (HLT-NAACL)*, pages 620–628, 2009a.

Zhiyuan Liu, Peng Li, Yabin Zheng, and Maosong Sun. Clustering to find exemplar terms for keyphrase extraction. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP 2009)*, pages 257–266, 2009b.

Zhiyuan Liu, Wenyi Huang, Yabin Zheng, and Maosong Sun. Automatic keyphrase extraction via topic decomposition. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing (EMNLP 2010)*, pages 366–376, 2010.

Xavier Llorà, Kumara Sastry, David E. Goldberg, Abhimanyu Gupta, and Lalitha Lakshmi. Combating user fatigue in iGAs: partial ordering, support vector machines, and synthetic fitness. In *Proceedings of the 2005 Conference on Genetic and Evolutionary Computation (GECCO '05)*, pages 1363–1370, Washington, D.C., 2005.

Hans Peter Luhn. A statistical approach to mechanized encoding and searching of literary information. *IBM Journal of Research and Development*, 1(4):309–317, 1957.

Yutaka Matsuo and Mitsuru Ishizuka. Keyword extraction from a single document using word co-occurrence statistical information. *International Journal on Artificial Intelligence Tools*, 13(1):157–169, 2004.

Andrew K. McCallum. MALLET: A machine learning for language toolkit. http://mallet.cs.umass.edu, 2002.

Iain McCowan. Microphone arrays and beamforming. In *Multimodal Signal Processing: Human Interactions in Meetings*, pages 28–39. Cambridge University Press, Cambridge, UK, 2012.

Rada Mihalcea and Paul Tarau. TextRank: Bringing order into texts. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2004)*, pages 404–411, 2004.

Tomas Mikolov, Kai Chen, Greg S. Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *Preprint arXiv:1301.3781*, 2013a.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th Annual Conference on Neural Information Processing Systems (NIPS)*, pages 3111–3119, 2013b.

David Mimno, Hanna M Wallach, Jason Naradowsky, David A Smith, and Andrew McCallum. Polylingual topic models. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2-Volume 2*, pages 880–889. Association for Computational Linguistics, 2009.

Mehryar Mohri, Pedro Moreno, and Eugene Weinstein. Discriminative topic segmentation of text and speech. In *International Conference on Artificial Intelligence and Statistics*, pages 533–540, 2010.

George L. Nemhauser, Laurence A. Wolsey, and Marshall L. Fisher. An analysis of approximations for maximizing submodular set functions. *Mathematical Programming Journal*, 14(1): 265–294, 1978.

Ani Nenkova and Kathleen McKeown. A survey of text summarization techniques. In Charu C. Aggarwal and ChengXiang Zhai, editors, *Mining Text Data*, chapter 3, pages 43–76. Springer, 2012.

Xiaochuan Ni, Jian-Tao Sun, Jian Hu, and Zheng Chen. Mining multilingual topics from wikipedia. In *Proceedings of the 18th international conference on World wide web*, pages 1155–1156. ACM, 2009.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics, 2002.

Nikolaos Pappas and Andrei Popescu-Belis. Combining content with user preferences for non-fiction multimedia recommendation: a study on ted lectures. *Multimedia Tools and Applications*, 74(4):1175–1197, 2015.

Laurence A. F. Park and Kotagiri Ramamohanarao. Query expansion using a collection dependent probabilistic latent semantic thesaurus. In *Advances in Knowledge Discovery and Data Mining*, pages 224–235. Springer, 2007.

Andrei Popescu-Belis, Erik Boertjes, Jonathan Kilgour, Peter Poller, Sandro Castronovo, Theresa Wilson, Alejandro Jaimes, and Jean Carletta. The AMIDA Automatic Content Linking Device: Just-in-Time document retrieval in meetings. In *Proceedings of MLMI 2008 (Machine Learning for Multimodal Interaction)*, LNCS 5237, pages 272–283, 2008.

Andrei Popescu-Belis, Majid Yazdani, Alexandre Nanchen, and Philip N. Garner. A speech-based Just-in-Time retrieval system using semantic search. In *Proceedings of the 2011 Annual Conference of the North American Chapter of the ACL (HLT-NAACL)*, pages 80–85, 2011.

Wilfried Post, Erwin Elling, Anita Cremers, and Wessel Kraaij. Experimental comparison of multimodal meeting browsers. In *Human Interface and the Management of Information. Interacting in Information Environments*, pages 118–127. Springer, 2007.

Filip Radlinski and Susan Dumais. Improving personalized web search using result diversification. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 691–692. ACM, 2006.

Steve Renals, Hervé Bourlard, Jean Carletta, and Andrei Popescu-Belis, editors. *Multimodal Signal Processing: Human Interactions in Meetings.* Cambridge University Press, Cambridge, UK, 2012.

Bradley J. Rhodes. The wearable Remembrance Agent: A system for augmented memory. *Personal Technologies*, 1(4):218–224, 1997.

Bradley J. Rhodes and Pattie Maes. Just-in-Time information retrieval agents. *IBM Systems Journal*, 39(3.4):685–704, 2000.

Bradley J. Rhodes and Thad Starner. Remembrance Agent: A continuously running automated information retrieval system. In *Proceedings of the 1st International Conference on the Practical Application of Intelligent Agents and Multi Agent Technology*, pages 487–495, London, 1996.

John Richardson, Toshiaki Nakazawa, and Sadao Kurohashi. Robust transliteration mining from comparable corpora with bilingual topic models. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing, Nagoya, Japan*, pages 261–269, 2013.

Korbinian Riedhammer, Benoit Favre, and Dilek Hakkani-Tur. A keyphrase based approach to interactive meeting summarization. In *Spoken Language Technology Workshop, 2008. SLT 2008. IEEE*, pages 153–156, 2008.

Stephen E. Robertson. The probability ranking principle in IR. In Karen Sparck Jones and Peter Willett, editors, *Readings in information retrieval*, pages 281–286. Morgan Kaufmann Publishers Inc., 1997. ISBN 1-55860-454-5.

Stephen E. Robertson, Steve Walker, Micheline Beaulieu, and Peter Willett. Okapi at TREC-7: automatic ad hoc, filtering, VLC and interactive track. *NIST Special Publication SP*, pages 253–264, 1999.

Joseph J. Rocchio. Relevance feedback in information retrieval. In Gerard Salton, editor, *The SMART Retrieval System: Experiments in Automatic Document Processing*, chapter 14, pages 313–323. Prentice-Hall, Englewood Cliffs, NJ, 1971.

Stuart Rose, Dave Engel, Nick Cramer, and Wendy Cowley. Automatic keyword extraction from individual documents. *Text Mining*, pages 1–20, 2010.

Gerard Salton and Chris Buckley. Improving retrieval performance by relevance feedback. *Readings in Information Retrieval*, 24:5, 1997.

Gerard Salton and Christopher Buckley. Term-Weighting approaches in automatic text retrieval. *Information Processing and Management Journal*, 24(5):513–523, 1988.

Gerard Salton, Chung-Shu Yang, and Clement T. Yu. A theory of term importance in automatic text analysis. *Journal of the American Society for Information Science*, 26(1):33–44, 1975.

Dairazalia Sanchez-Cortes, Oya Aran, Marianne Schmid Mast, and Daniel Gatica-Perez. A non-verbal behavior approach to identify emergent leaders in small groups. *IEEE Transactions on Multimedia*, 14(3):816–832, 2012.

Rodrygo L. T. Santos, Craig Macdonald, and Iadh Ounis. Exploiting query reformulations for Web search result diversification. In *Proceedings of the 19th Int. Conf. on the World Wide Web*, pages 881–890, 2010.

Padhraic Smyth, Usama Fayyad, Michael Burl, Pietro Perona, and Pierre Baldi. Inferring ground truth from subjective labeling of venus images. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1085–1092, 1994.

Mark Steyvers and Tom Griffiths. Probabilistic topic models. *Handbook of latent semantic analysis*, 427(7):424–440, 2007.

Jinsong Su, Hua Wu, Haifeng Wang, Yidong Chen, Xiaodong Shi, Huailin Dong, and Qun Liu. Translation model adaptation for statistical machine translation with monolingual topic information. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 459–468. Association for Computational Linguistics, 2012.

Yik-Cheung Tam, Ian Lane, and Tanja Schultz. Bilingual LSA-based adaptation for statistical machine translation. *Machine translation*, 21(4):187–207, 2007.

Paul Thomas and David Hawking. Evaluation by comparing result sets in context. In *Proceedings of the 15th ACM International Conference on Information and Knowledge Management (CIKM)*, pages 94–101, 2006.

David Traum, Priti Aggarwal, Ron Artstein, Susan Foutz, Jillian Gerten, Athanasios Katsamanis, Anton Leuski, Dan Noren, and William Swartout. Ada and Grace: Direct interaction with museum visitors. In *Proceedings of the 12th international conference on Intelligent Virtual Agents*, pages 245–251, 2012.

Peter Turney. Learning to extract keyphrases from text. Technical Report ERB-1057, National Research Council Canada (NRC), 1999.

Saúl Vargas, Pablo Castells, and David Vallet. Explicit relevance models in intent-oriented information retrieval diversification. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pages 75–84. ACM, 2012.

Ellen M. Voorhees and Donna K. Harman. *TREC: Experiment and Evaluation in Information Retrieval*. MIT Press, Cambridge, MA, 2005.

Ivan Vulić and Marie-Francine Moens. Cross-lingual semantic similarity of words as the similarity of their semantic word responses. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2013)*, pages 106–116, 2013a.

Ivan Vulić and Marie-Francine Moens. A study on bootstrapping bilingual vector spaces from non-parallel data (and nothing else). In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP 2013)*, pages 1613–1624, 2013b.

Ivan Vulić, Wim De Smet, and Marie-Francine Moens. Identifying word translations from comparable corpora using latent topic models. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 479–484. Association for Computational Linguistics, 2011.

Dong Wang, Dilek Hakkani-Tur, and Gokhan Tur. Understanding computer-directed utterances in multi-user dialog systems. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 8377–8381, 2013.

Jinghua Wang, Jianyi Liu, and Cong Wang. Keyword extraction based on PageRank. In *Proceedings of Advances in Knowledge Discovery and Data Mining (PAKDD)*, pages 857–864. Springer, 2007a.

Jun Wang and Jianhan Zhu. Portfolio theory of information retrieval. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 115–122. ACM, 2009.

Xuerui Wang, Andrew McCallum, and Xing Wei. Topical n-grams: Phrase and topic discovery, with an application to information retrieval. In *Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on*, pages 697–702. IEEE, 2007b.

William Webber, Alistair Moffat, and Justin Zobel. A similarity measure for indefinite rankings. *ACM Transactions on Information Systems (TOIS)*, 28(4):20:1–20:38, 2010.

Jacob Whitehill, Ting-fan Wu, Jacob Bergsma, Javier R. Movellan, and Paul L. Ruvolo. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2035–2043. Curran Associates, Inc., 2009.

Shengli Wu and Sally McClean. Result merging methods in distributed information retrieval with overlapping databases. *Information Retrieval*, 10(3):297–319, 2007.

Xinyan Xiao, Deyi Xiong, Min Zhang, Qun Liu, and Shouxun Lin. A topic similarity model for hierarchical phrase-based translation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 750–758. Association for Computational Linguistics, 2012.

Deyi Xiong and Min Zhang. A topic-based coherence model for statistical machine translation. In *Twenty-Seventh AAAI Conference on Artificial Intelligence*, 2013.

Jinxi Xu and W. Bruce Croft. Query expansion using local and global document analysis. In *Proceedings of 19th annual int. ACM SIGIR conf. on Research and development in IR*, pages 4–11, 1996.

Jinxi Xu and W. Bruce Croft. Improving the effectiveness of information retrieval with local context analysis. *ACM Trans. on Information Systems (TOIS)*, 18(1):79–112, 2000.

Majid Yazdani. *Similarity Learning Over Large Collaborative Networks*. PhD thesis, EPFL Doctoral School in Information and Communication (EDIC), 2013.

Majid Yazdani and Andrei Popescu-Belis. Computing text semantic relatedness using the contents and links of a hypertext encyclopedia. *Artificial Intelligence*, 194:176–202, 2013.

Shiren Ye, Tat-Seng Chua, Min-Yen Kan, and Long Qiu. Document concept lattice for text understanding and summarization. *Information Processing and Management*, 43(6):1643–1662, 2007.

Dawei Yin, Zhenzhen Xue, Xiaoguang Qi, and Brian D. Davison. Diversifying search results with popular subtopics. Technical report, DTIC Document, 2009.

Jennifer Zaino. MindMeld makes context count in search, 2014. URL http://semanticweb.com/mindmeld-makes-context-count-search_b42725.

Richard Zens, Franz Josef Och, and Hermann Ney. Phrase-based statistical machine translation. In *KI 2002: Advances in Artificial Intelligence*, pages 18–32. Springer, 2002.

Cheng Xiang Zhai, William W. Cohen, and John Lafferty. Beyond independent relevance: methods and evaluation metrics for subtopic retrieval. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 10–17. ACM, 2003.

Chengzhi Zhang, Huilin Wang, Yao Liu, Dan Wu, Yi Liao, and Bo Wang. Automatic keyword extraction from documents using Conditional Random Fields. *Journal of Computational Information Systems*, 4(3):1169–1180, 2008.

Bing Zhao and Eric P Xing. HM-BiTAM: Bilingual topic exploration, word alignment, and translation. In *Advances in Neural Information Processing Systems*, pages 1689–1696, 2008.

Resume

Name: Maryam Habibi

Address: Idiap Research Institute, Centre du Parc, Rue Marconi 19,
CH-1920 Martigny, Switzerland
PO Box: 592,
Mobile Number: (+41) 78 602 82 54
Office Number: (+41) 27 721 77 97

Email Address: maryam.habibi@epfl.ch, maryam.habibi@idiap.ch

Home Page: http://www.idiap.ch/~mhabibi/

Education

- **PhD student:** Department of Electrical Engineering, École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland (Sep2011 – Sep2015)
- **Master of Science:** Department of Computer Engineering (Artificial Intelligence), Sharif University of Technology, Tehran, Iran (Sep2008 - Sep2010)
- **Bachelor of Science:** Department of Computer Engineering (Hardware), Sharif University of Technology, Tehran, Iran (Sep2003 - Jan2008)

Professional experience

- **Postdoctoral Research Assistant:** Knowledge Management in Bioinformatics, Humboldt-Universität zu Berlin, Berlin, Germany (Sep 2015-present)
- **PhD Research Assistant:** Natural Language Processing group, Idiap Research Institute, Martigny, Switzerland (Sep 2011-Sep 2015)
- **Research Assistant:** Speech Processing Laboratory, Sharif University of Technology, Tehran, Iran (Jan 2008-Sep2010)
- **Researcher (part time):** Asr Gooyesh, Tehran, Iran (Jan 2008 –Sep 2009)
- **Hardware Engineer (part time):** IranAir, Tehran, Iran (July 2007-Jan2008)

Computer skills

- **Computer Programming:** C, Java, Verilog, VHDL (HDL)
- **Operating Systems:** Linux, Microsoft Windows
- **Others:** Microsoft Office, MATLAB, Latex

Language skills

English (full-working proficiency), German (elementary A2), Persian (mother tongue)

Teaching experience

- **Speech Processing**, Sharif University of Technology, one semester during 2010
- **Theory of Machines & Languages**, Sharif University of Technology, four semesters during 2007-2010
- **Signals and Systems processing**, Sharif University of Technology, one semester during 2009
- **Electrical Circuits**, Sharif University of Technology, one semester during 2007
- **Digital Electronics**, Sharif University of Technology, two semesters during 2005-2007

Honors

- Ranked first out of eleven systems on MediaEval 2014 on the hyper-linking task, and third out of seven on the keyword video search task (with Idiap NLP group)
- Selected as the winner of the ACM Multimedia Grand Challenge on video annotation and search task (with Idiap NLP group)
- Ranked 267 among about half million students participating in the Nation-Wide Entrance Exam for BSc studies
- Chosen as a talented student in the bachelor program at Sharif University of Technology in 2007, and consequently was offered to study MSc without taking the national entrance examination
- Ranked as top two students of BSc hardware engineering program, and top ten students in the Computer Engineering Department at Sharif University of Technology out of 110 students
- Ranked as top two students of MSc artificial intelligence program in the Computer Engineering Department at Sharif University of Technology

Publications

1- **M. Habibi** and A. Popescu-Belis. Keyword Extraction and Clustering for Document Recommendation in Conversations. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, pp. 746-759, Volume 23, Issue 4, 2015.

2- **M. Habibi** and A. Popescu-Belis. Query Refinement Using Local Context from Conversations: a Method and a Resource for its Evaluation. *In Proceedings of the 20th International Conference on Application of Natural Language to Information Systems (NLDB 2015)*, Passau, Germany, 2015.

3- **M. Habibi** and A. Popescu-Belis. Enforcing Topic Diversity in a Document Recommender for Conversations. *In Proceedings of the 25th International Conference on Computational Linguistics (Coling 2014)*, pp. 588-599, Dublin, Ireland, 2014.

4- C. Bhatt, N. Pappas, **M. Habibi** and A. Popescu-Belis. Multimodal Reranking of Content-based Recommendations for Hyperlinking Video Snippets. *In Proceedings of the 4th ACM International Conference on Multimedia Retrieval*

(ACM ICMR 2014), special session on User-centric Video Search and Hyperlinking, pp. 225-232, Glasgow, Scotland, 2014.

5- **M. Habibi** and A. Popescu-Belis. Diverse Keyword Extraction from Conversations. *In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL 2013)*, pp. 651-657, Sofia, Bulgaria, 2013.

6- C. Bhatt, N. Pappas, **M. Habibi** and A. Popescu-Belis (2013). Idiap at MediaEval 2013: Search and Hyperlinking Task. *In Proceedings of the MediaEval 2013 Workshop*, Barcelona, Spain, 2013.

7- C. Bhatt, A. Popescu-Belis, **M. Habibi** and S. Ingram, F. McInnes, S. Masneri, N. Pappas and O. Schreer. Multi-factor Segmentation for Topic Visualization and Recommendation: the MUST-VIS System. *In Proceedings of the 21st ACM International Conference on Multimedia (MM 2013), Grand Challenge Solutions*, pp. 365-368, Barcelona, Spain, 2013.

8- **M. Habibi** and A. Popescu-Belis. Using Crowdsourcing to Compare Document Recommendation Strategies for Conversations. *In Proceedings of ACM RecSys Workshop on Recommendation Utility Evaluation: Beyond RMSE (RUE 2012)*, pp. 15-20, Dublin, Ireland, 2012.

9- **M. Habibi**, S. Rahbar and H. Sameti. Divided POMDP Method for Complex Menu Problems in Spoken Dialogue Systems. *In Proceedings of the Spoken Language Technology Workshop (SLT 2010)*, pp. 484-489, Berkeley, California, U.S.A., 2010.

10- **M. Habibi**, H. Sameti and H. Setareh. On-Line Learning of a Persian Spoken Dialogue System Using Real Training Data. *In Proceedings of the 10th International Conference on Information Sciences, Signal Processing and their Applications, (ISSPA 2010)* , pp. 133-136, Kuala Lumpur, Malaysia, 2010.

11- **M. Habibi**, H. Sameti and H. Setareh. On-Line Learning of a Persian Spoken Dialogue System Using Real Training Data. *Journal of Advances In Computer Research*, pp. 31-39, Volume 1, Issue 2, 2010.