RESEARCH ARTICLE OPEN ACCESS

Implementation Paper on Document Recommendation in Conversations

Anshika[1], Sujit Tak[2], Sandeep Ugale[3], Abhishek Pohekar[4]
(Dept. of Computer Engineering, D.Y.P.I.E.T., Savitribai Phule Pune University, Pune)

Abstract:

This paper is focused towards extraction of important words from conversations, and then these words are utilized to recover a small number of relevant documents, which can be given to users as per their needs. In any case, even a short audio fragment contains of large number of words, which are conceivably identified with a few topics and also, using automatic speech recognition (ASR) framework reduces errors in the output. It is difficult to provide the data needs of the conversation members appropriately. A calculation to remove decisive words from the yield of an ASR framework (or a manual transcript for testing) to support the potentially differing qualities of subjects and decrease ASR commotion is proposed. At that point, a technique is used to make many implicit queries from the selected keywords. These queries will in return produce list of relevant documents. The scores demonstrate that our proposition is modified over past systems that consider just word re-occurrence or topic closeness, and speaks to an appropriate answer for a report recommender framework to be utilized as a part of discussions.

Keywords — **Document recommendation, information retrieval, keyword extraction.**

I. INDRODUCTION

Humans are surrounded by abundance of data, accessible as records, data stores, or mixed media sources of information. This data can be accessed by using suitable web indexes, however when these are accessible, clients frequently don't start a search action, in light of the reality that their present work does not allow them to do such tasks, or they are not mindful that applicable data is accessible. In this paper, a novel technique of suggesting archives just-in-time that is identified with clients' present work is suggested. At the time, when these tasks are primarily conversational, for instance when clients take part in a meeting, then their data needs can be understood by the keywords present in speech, acquired through continuous automatic speech recognition (ASR) engine. These certain implicitly generated questions are utilized to recover and suggest reports or documents from the websites or a local repository, which clients can decide to, investigate in more detail if they discover them intriguing. The focus of our paper is on showing framework for utilization in meeting room or conference where information needs to be fetched in-time, to help involved people in better understanding of the topic. This shown framework will be including speech to text translation, extraction of words from this text, clustering those words for topic identification, generating implicit queries from identified main words and retrieving documents from the available storehouses related to the words in the queries.

Various libraries (regional/language specific) can be added to the ASR engine to improve speech recognition efficiency.

II. RELATED WORK

A. *Remembrance Agent: A continuously running automated information retrieval system.*

Authors: B. Rhodes and T. Starner

Remembrance Agent (RA) is a program which can display list of documents relevant to user needs. RA executes without continuous user interaction. RA is basically used to give suggestions to user which can be used or ignored as desired. Computer used to wait for user interaction, RA makes use of such wait cycle to perform its search operation which is relevant to user's context. RA reminds about relevant data of user task. RA gives suggestions in one line description at bottom of screen. For the front-end for RA, EMACS-19 and lisp are used. One-line suggestions and number to indicate relevancy is displayed by front-end. Full text document can be obtained as per the request. A program which is used as back-end for RA produces suggestions for given text query from pre-indexed documents. SMART decides suggestions for RA, depending upon common word frequency.

This system consists of two sources of suggestions. In first source, first three lines print suggestions from last year's personal email (approximately up to 60MB). And in case of second source, last line prints suggestions from 7MB files which were entered over past few years. RA should suggest documents related to keywords in query. This works on 'scopes', which are centred on current location of cursor. There are three types of scopes used in RA. First scope is around the last thousands of words. Second is around the last 50 words. And third is around the last 10 words. User can customize scope coverage, time updates.

B. *Enforcing topic diversity in a document recommender for conversations.*

Authors: M. Habibi and A. Popescu-Belis

In our system, we also focus on building of concise, diverse and relevant lists of documents which provide information to the users according to their need.

From this paper, we study the concept of merging lists of documents which are obtained through multiple implicit queries, formed for small conversations fragments.

The aim of the method used in this paper is to get a unique and precise list of documents that can satisfy user's need in real time. The obtained list should cover the maximum number of implicit queries and topics.

The proposed method rewards:

a. **Topic similarity** – It is concerned with selection of the most relevant documents to the conversation.
b. **Topic diversity** – It is concerned with coverage of the maximum implicit queries and topics in a precise and relevant list of recommendations when more than one topic is discussed in the conversation.

ACLD system is used as framework for this document recommender system. The ACLD system monitors the conversation, and generates queries based on the keywords detected by a real-time ASR system. The queries are then fired periodically for retrieving documents. These documents are then recommended to the users according to their needs.

Here a diverse merging of lists is done. It is the process of generating a short, diverse and relevant list of recommended documents. The generated list should cover the maximum number of topics of each conversation fragment.

The method proceeds in two steps:

a. Document and Query Representation:

In this part, the representation of the queries and the corresponding list of documents from the Apache Lucene search engine are done. This representation is done using topic modelling techniques.

b. Ranking Documents:

In this step, ranking of documents is done by using topical similarity and then, rewarding the coverage of different lists is taken care.

C. *A Speech-based Just-in-Time Retrieval System using Semantic Search*

Authors: A. Popescu-Belis, M. Yazdani, A. Nanchen, and P. N. Garner

A just-in time retrieval system for extraction of useful information is required in our project. We have used ACLD i.e., Automatic Content Linking Device (ACLD) for our project. ACLD is a just-in-time document recommendation system for meetings.

From this paper, we study the concept of content linking for our recommendation system. This is done by storing documents with tags.

a. What is ACLD?

ACLD is a recommendation system which analyses spoken input from one or more speakers and retrieves relevant documents for user as per their need.

b. Content Linking: Scenarios of Use

One of the main usage scenarios for the ACLD is meeting rooms. The ACLD searches the relevant documents for members in the meeting without interrupting their discussion flow. In other usage scenarios, ACLD can be used for live or recorded lectures.

The ACLD provides related and relevant documents which are obtained from various repositories, required for the lectures.

c. Description of the ACLD

ACLD system performs following functions:

i. Document Preparation and Indexing:

This function is concerned with the extraction of text, and then the indexing of the documents. Apache Lucene software is used for this purpose.

ii. **Sensing the User's Information Needs**

The ACLD system makes use of the AMI real-time ASR system. One of its main features is a pre-compiled grammar. This feature helps in maintaining accuracy even in real time on a low resource machine.

iii. **Querying the Document Database**

The Query Aggregator retrieves the most relevant documents from one or more databases using ASR words. The current version of the ACLD makes use of semantic search, while word-based search is used for previous version.

iv. **Semantic Search over Wikipedia**

The main focus of our method of semantic search is to improve the relevancy of the retrieved documents as well as the robustness of system to remove noise from the ASR. Here, for document retrieval, the graph-based model of semantic relatedness is used.

v. **The User Interface (UI)**

The UI mainly makes all information produced by the system available to user in a configurable way. The users are allowed to use a larger or smaller amount of information according to their needs in this.

ACLD is the first just in time retrieval system that uses involuntary speech. Access to multimedia documents and web pages using a robust semantic search method is also supported.

D. *User Interactions with Everyday Applications as Context for Just-in-time Information Access*
Authors: J. Budzik and K. J. Hammond

Information retrieval is one of the important functions of our project. The retrieved information is then used for obtaining relevant and related documents. As per observations, rich contextual information obtained from our interactions with everyday applications, such as, word processors, web browsers etc. can be used to support just in time retrieval of relevant documents.

This paper provides us the concept of just in time information retrieval by using user interactions with everyday applications as a context. For obtaining more clear view about just in time information retrieval, we are using Watson system as an example.

Previous efforts in building context for information access can be classified into four classes:

a. **Relevance feedback in information retrieval:**

In relevance feedback supporting system, we start with a standard query and then judge the result. Usually a result is judged as relevant or not relevant. Then, this result of user's judgment can used to customize the original query by casting positive or negative search terms.

b. **Systems that use user profiles:**

The aggregation of user profiles can be done by gathering terms based on rating of documents.

c. **Word-sense disambiguation:**

Explicit word-sense disambiguation on the user's part, or approved information intrinsic in the hypertext documents structure can be used for reducing ambiguity in some systems.

d. **Knowledge engineering approaches:**

The behaviour of user in a particular application is modelled and then explicit queries are generated in a particular state of tasks in such approaches.

III. PROPOSED SYSTEM

We propose a system where, output is taken from Automatic Speech Recognition (ASR) system. This output is processed using topic modelling technique instead of directly selecting words on re-occurrence basis. Various functions are used in the system to process. In first step, noisy words like articles (a, an, the), unwanted words, common unnecessary symbols, etc. are removed from the conversation fragment. Human critics ranks various conversational corpora like Fisher, AMI and ELEA which provides audio pieces to test our proposed methods on the basis of relevance with respect to audio pieces from the conversational corpora.

Now, a set (mathematics term) is formed from words that are obtained after noisy word removal process. Clustering is carried on the words present in the set by using clustering algorithm. In our system we are using k-means as it is simpler to use and implement and which in turn, reduces the processing time of the system and do not put much load on the system.

According topical similarity and co-occurrence of words, the clusters are formed. Then, from each cluster, one query will be formulated implicitly. This will generate more precise query. Documents will be retrieved for each topically different query. One of the primary tasks is to present the list of documents in user-friendly way. Two column

GUI is used for this for showing document to user. They are:

1. Query relevant document.
2. Document's similarity percentage to the keyword (They are based on co-occurrence of words).

At various stages of processing, different technique are used to remove errors. Ranking of document helps user in accessing them before displaying.

IV. MAIN ALGORITHM

1) Latent Dirichlet Allocation

LDA, used for topic modelling, assumes the following generative process for each document w in a large and structured set of texts D:

1. Choose N ~ Poisson(ξ)
2. Choose θ ~ Dir (α).
3. For each of the N words wn:
   a. Choose a topic zn ~ Multinomial (θ).
   b. Choose a word wn from p (wn |zn, β), a multinomial probability conditioned on the topic zn.

A k-dimensional Dirichlet random variable θ can take values in the (k −1)-simplex (a k-vector θ lies in the (k−1)-simplex if θi ≥ 0, Σk i=1 θi = 1), and has the following probability density on this simplex:

$$p(\theta|\alpha) = \{ \Gamma (\sum_{k\ i=1} \alpha i) / (\prod_{k\ i=1} \Gamma(\alpha i)) \} \ \theta\alpha 1-1\ 1 \cdots \theta\alpha k-1\ k$$

2) Diverse Keyword Extraction

Algorithmic form is as follows:

Input: A given text t, a set of topics Z, the number of keywords k;
Output: a set of keywords S

$S \leftarrow \emptyset$;
while $|S| \leq k$ do
$S \leftarrow S \cup \{\text{argmax}_{w \in t \setminus S} (h(w))$ where
$h(w) = P_{z \in Z}\ p(z|t)[\ \mathbf{r}_{\{w\},z} + \mathbf{r}_{S,z}]^{\lambda}\}$;
end
return S;

Where, $\mathbf{r}_{S,z}$ is the topical similarity with respect to topic z of the keyword set S selected from the fragment t.

V. SYSTEM ARCHITECTURE

Figure 4 shows ASR system records user's conversation. The output of ASR is send further to noise removal function. Then diverse keywords are extracted using keyword extraction method. After that these diverse keywords are clustered. Further these keywords are used for obtaining result of implicit queries. Then relevant documents are recommended to user.

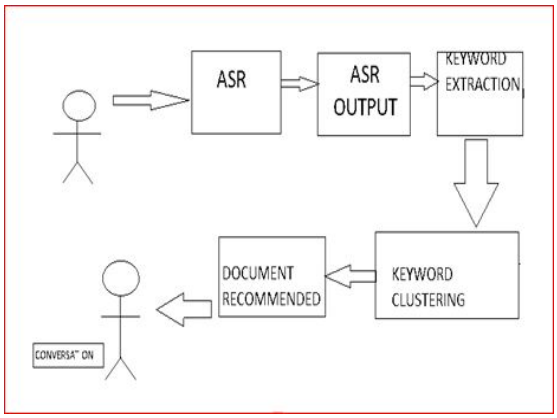

Fig: System Architecture

VI. SYSTEM SCREENSHOTS

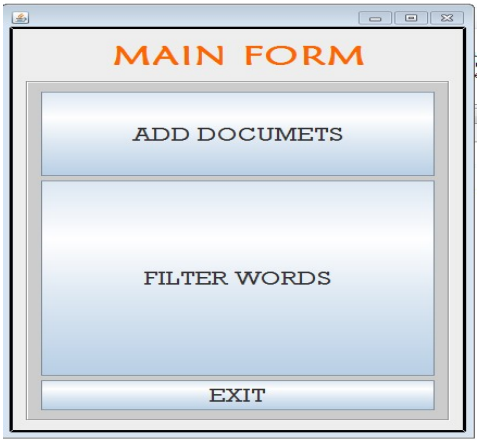

Fig. 1: Administrator UI

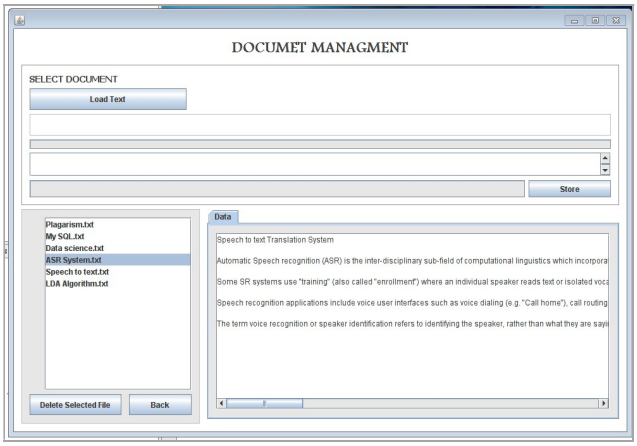

Fig. 2: Uploading documents to local repository

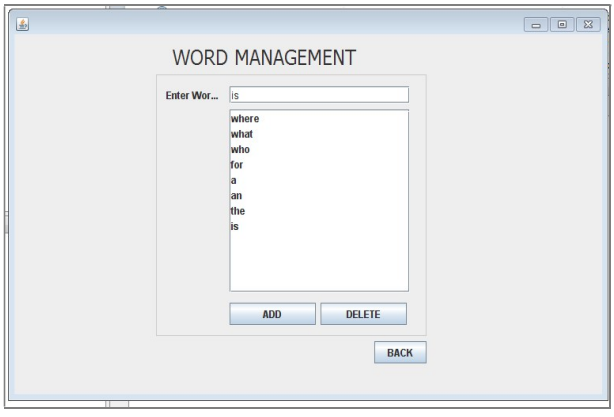

Fig. 3: Setting filter words

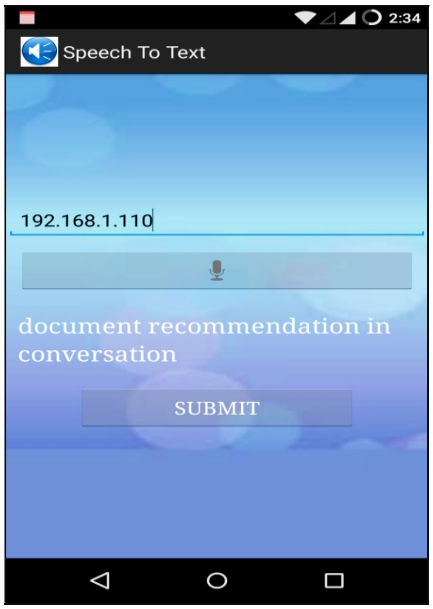

Fig. 4: Android Application (ASR Engine)

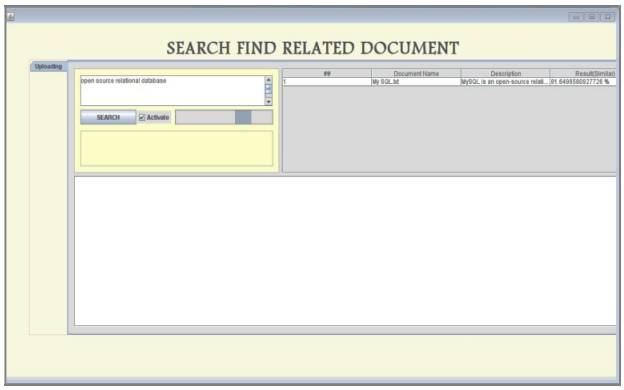

Fig. 5.1: Relevant Document Retrieved in Client UI

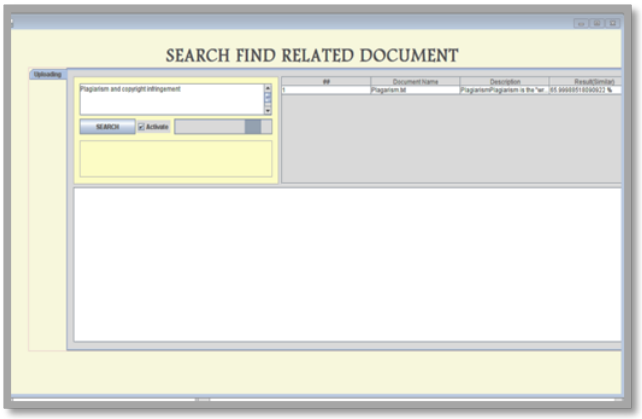

Fig. 5.2: Relevant Document Retrieved in Client UI

VII. FUTURE SCOPE

With more processing power, the system could be made real-time recommender. Also, the system can be linked with online encyclopedias like Wikipedia, etc. for more detailed recommendations. The common APIs used in ASR system can be added with region-specific APIs to improve its efficiency.

VIII. CONCLUSION

In our system architecture, there are four steps of processing after obtaining the ASR output. The speed and precision are the main criterions to judge a just-in-time recommender. Topic diversity is maintained by extracting the keywords using a diverse keyword search algorithm. There are many practical applications like meeting rooms, conferences, workshops, etc. because of implicit query formation process.

IX. ACKNOWLEDGEMENT

We would like to thank our guide Prof. B. S. Satpute for helping us in understanding complex technical issues regarding project and for helping in all the phases of system development. We are also grateful to Dr. R. K. Jain and Dr. Pramod Patil for their wholehearted support, guidance and for providing us with required amenities and infrastructure. We would like to thank to all the authors, whose paper we have referred, for their research in this field. It helped us in understanding the algorithms and various techniques in detail.

REFERENCES

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

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

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

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

5. *Maryam Habibi and Andrei Popescu-Belis, "Keyword Extraction and Clustering for Document Recommendation in Conversations", in IEEE/ACM Transactions on Audio, Speech and Language Processing, Vol. 23, NO. 4, April 2010*