Proposed Architecture for Punjabi Language Question Answering using Text Mining

[1]Vishal Gupta, [2]G.S. Lehal

Abstract

In this paper we have proposed an approach for Punjabi language web question answering using text mining. Text Mining is a technique for automatically discovering knowledge from unstructured text documents. Traditional search engines like Google, give us only the correct links to web pages. These search engines will not produce the useful answers, but will only give useful links. Our proposed method will give you, the correctly ranked answers of questions typed by you. This architecture is very helpful for people of Punjab. Now they have not to waste their too much time in exploring the web links returned by google, but our system will directly produce the ranked answers of question. Good ranked answers are more relevant than lower ranked answers. We have proposed this architecture for Punjabi language text

Keywords : *Text Mining, Web Mining, Web Question Answering, Query Reformulation, Synonyms detection, Vector space Model.*

1. Introduction

Text Mining[1] is a technique for automatically discovering knowledge from unstructured text documents. Nowadays there is a large amount of digital documents accessible from the web. These documents may satisfy almost every information need. However, without the appropriate mechanisms that help users to find the required information when they need it, all these documents are practically useless. In order to solve this dilemma several information access approaches have emerged. Two popular examples are: information retrieval [8] (IR) and question answering (QA). Information retrieval addresses the problems associated with the retrieval of documents from a collection in response to a user query. The goal of an IR system is to search a text collection and return as result a subset of documents ordered by decreasing likelihood of being relevant to the given query. The most popular IR systems are the search engines for the web. For instance, Google, Altavista and Yahoo. The current IR systems allow finding relevant documents for a given user need, but are incapable to return a concise answer for a specific information request [12]. The alternative to IR systems for resolving specific questions are the question answering (QA) systems. These systems are capable to answer questions formulated by the users in natural language. Recent developments in QA are mainly focused on answering factual questions (those having a simple named entity as the answer), and are mainly suited to English as the target language. This paper presents the basis of a statistical QA system capable to find answers to factual questions in Punjabi from the web. This system is supported on the idea that the questions and their answers are commonly expressed using (almost) the same set of words. Therefore the answers may be extracted using simple lexical pattern matching methods, rather than sophisticated linguistic analyses of either questions and documents.

[1]Lecturer, University institute of Engineering & Technology , Panjab University Chandigarh, India. E-mail : vishal_gupta100@yahoo.co.in, vishal@pu.ac.in

[2]Professor, Department of Computer Science, Punjabi University Patiala, India.

Now we will briefly explain how our architecture will give correctly ranked answers of questions typed by you. First of all, our system will extract the string of question text, typed by user. After extracting the question string, we will eliminate the stop words from that string, as we are having a list of stop words for English and Punjabi languages. Then we will extract the keywords from that remaining question string. Keywords are usually chosen as nouns adjectives verbs etc. Next phase is to get the synonyms of these keywords by applying bilingual dictionary approach or Vector Space Model. Then query reformulation is done by using these keywords and synonyms of remaining query string. Then required web pages are extracted by string matching with these query reformulations. Our last module will return the answer snippets from the documents returned by search engine and will assign ranks to these answer snippets. Then we will select the top 20 snippets as answers to our question.

The rest of the paper is organized as follows. Section 2 briefly presents the current approaches on QA. Section 3 shows the architecture of our QA system and describes the methods for question reformulation and answer extraction. Section 4 presents current implementation, and finally, section 5 gives the future plans and summary.

2. Literature Survey

The question-answering (QA) paradigm, i.e. the process of retrieving precise answers to natural language (NL) questions [6], was introduced in late 1960-ies and early 1970-ies within the framework of NL understanding. It was mainly implemented in interfaces to problem solving systems for specific domains. The advent of WWW has reintroduced the need for user friendly querying techniques that reduce information overflow, and poses new challenges to the research in automated QA. Important QA application areas are information extraction from the entire Web ("intelligent" search engines), online databases etc. NLP techniques are used in applications that make queries to databases, extract information from text, retrieve relevant documents from a collection, translate from one language to another, generate text responses, or recognize spoken words converting them into text. NLP-based QA systems may utilize machine learning to improve their syntax rules , lexicon , semantic rules , or the world model. The first QA systems[9][10][11] used information retrieval techniques to retrieve the most relevant text passages based on the keywords of questions and documents. Current approaches use a variety of linguistic resources to help in understanding the questions and the matching sections in the documents. The most common linguistic resources include: part-of-speech tagging, parsing, named entity extraction, semantic relations, dictionaries, and Word Net[13][14][15]. Despite of the promising results of these approaches, they have two main inconvenients: (i) the construction of such linguistic resources is a very complex task; and (ii) these resources are highly binding to a specific language. In recent years, the combination of the web growth and the explosive demand for better information access has motivated the interest in QA systems for the web. Current approaches of QA on the web use a variety of linguistic resources for processing the questions and the web documents. However, the size of the web complicates their usage. As a result, new probabilistic methods based on the web redundancy have emerged. This paper presents a statistical QA system capable to find answers to factual questions in English and Punjabi language from the web. Its main idea is that the questions and their answers are commonly expressed using the same words, and that the probability of finding a simple (lexical) matching between them increases with the redundancy of the target collection. Therefore, given a question, our system generates several query reformulations manipulating the order of the words from the question. Then it sends each reformulation to a search engine, and collects the returned snippets (document summaries). Finally, it extracts the most frequent n-grams (sequences of words) from the snippets. Each n-gram is defined as a possible answer to the given question. The present work extents that of Brill[16] in the sense that it studies the application of this approach for questions and documents in English and Punjabi language. The main difference is on the query reformulation method. While Brill uses a lexicon to determine the part-of-speech of the question words as well as its morphological variants, we construct the query reformulations just manipulating the word order without using any previous knowledge about the words.

3. The Method

In this paper we have proposed a Text Mining approach of Web Question Answering System for Punjabi web documents. First of all, our system will extract the string of question text. typed by user. After extracting the question string, we will eliminate the stop words from that string, as we are having a list of stop words for Punjabi language. Then we will extract the keywords from that remaining question string. Keywords are usually chosen as nouns adjectives verbs etc. Next phase is to get the synonyms of these keywords by applying bilingual dictionary approach or Vector Space Model. Then query reformulation is done by using these keywords and synonyms of remaining query string. Then required web pages are extracted by string matching with these query reformulations. Our last module will return the answer snippets from the documents returned by search engine and will assign ranks to these answer snippets. Then we will select the top 20 snippets as answers to our question.

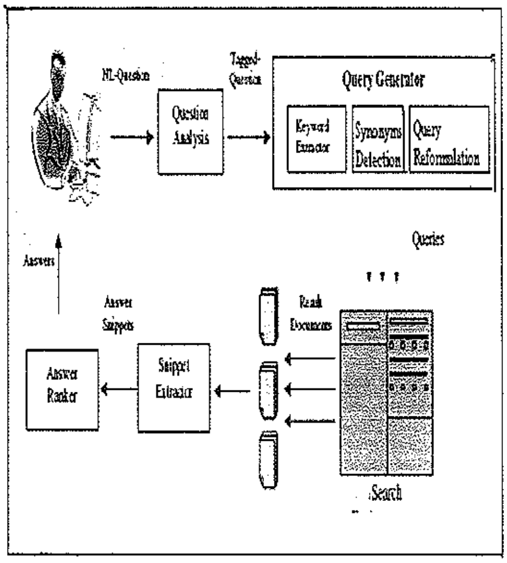

Figure 1: Architecture of the System

Figure1[4] shows the proposed architecture of our system. The various phases of our system are explained below:

A. Query Analysis

In query analysis[4][7] we will analyze the question string typed by user to select the keywords. The system takes in a natural language (NL) question in Punjabi ,from the user. This question is then passed to a Part-of-Speech (POS) tagger which parses the question and identifies POS of every word involved in the question. This tagged question is then used by the query generators which generate different types of queries, which can be passed to a search engine.

B. Query Generator Phase

In this phase query is reformulated. We are having the list of stop words for Punjabi language . So the first task of this phase is to eliminate the stop words from our query string. This phase includes three sub phases.

1) Keyword Extractor

After eliminating the stop words from the query string, next task is to extract the keywords. Normally, we chose Nouns, verbs and adjectives as keywords.

2) Synonyms Detection

In this phase we will select the synonyms[1][2][3][18] of keywords selected in previous phase.

The algorithm[18] for selecting synonyms of Punjabi language is as follows:

(a) Algorithm

Step1

Store the bilingual Punjabi dictionary in SQL database.

Step2

Input the Punjabi language Key words whose synonyms are to be found

Step3

Fetch the corresponding record of that word in record set.

Example-

ਚੰਗਾ ---- nice, good, fine

Step4

Fetch all those records containing any of the R.H.S. entries of previous record on R.H.S.

Example

ਵਧੀਆ ---Excellent,good,fine,fair .

i.e. we will fetch all those records in which R.H.S. field contains any of the entries among nice, good or fine.

Step5 (Result)

These selected records are synonyms of the Punjabi language Key word.

The Flow chart of Synonym detection phase[18] is as follows:

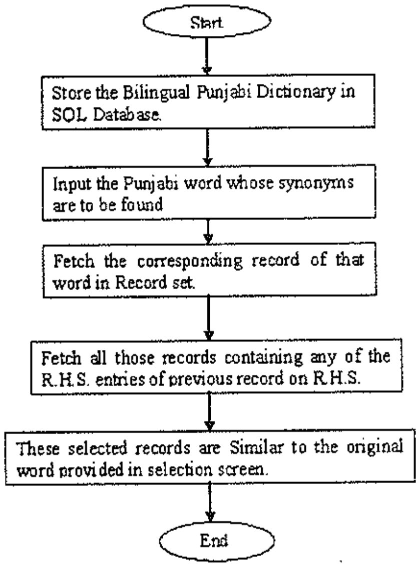
Start
Store the Bilingual Punjabi Dictionary in SQL Database.
Input the Punjabi word whose synonyms are to be found
Fetch the corresponding record of that word in Record set.
Fetch all those records containing any of the R.H.S. entries of previous record on R.H.S.
These selected records are Similar to the original word provided in selection screen.
End

Figure2. Flow Chart for Synonym Detection Phase

For English language synonyms detection, WorldNet dictionary can be used, and then the same algorithm can be used over WorldNet dictionary.

3) Query Reformulations

Given a question, this module generates a set of query reformulations [5]. These reformulations are expressions that were probably used to write down the expected answer. We will make reformulations using Key words as well as Synonyms of these key words, after removing the stop words from question string. In the algorithms described below, we represent a question Q as a set of words, i.e., Q = {w0, w1,…,wn-1}. Here w0 corresponds to the wh-word, and n indicates the number of words of the question. On the other hand, we represent a query reformulation R as a symbol string. This string consists of words, spaces, and quotation marks, and it satisfies the format of a conventional search engine query. For instance the reformulation R = wi wj corresponds to the query wi AND wj.

All the cases are illustrated for the question: *Who received the Nobel Peace Prize in 1992?*

(a) First reformulation: "bag of words"

This reformulation is the set of non stop-words of the question. It is built as follows:

```
For each $w_i \in Q \mid i \geq 1$
    If $w_i$ is not a stop word
        $R \leftarrow w_i$
Save R
```

The reformulation generated for the example query is:

(*received Nobel Peace Prize 1992*)

(b) Second reformulation: "verb movement"

One of our first observations after examine a list of factual questions was that the verb is frequently used right after the wh-word. We also know that in order to transform an interrogative sentence into a declarative one is necessary to eliminate the verb, or to move it to the final position of the sentence. The resulting sentence is expected to be more abundant in the web that the original one. In order to take advantage of this phenomenon, but without using any kind of linguistic resource, we propose to build a set of query reformulations eliminating, or moving to the end of the sentence, the first and second words from the question.

Two examples of these kind of reformulations are:

("*the Nobel Peace Prize in 1992 received*")

("*Nobel Peace Prize in 1992*")

(c) Third reformulation: "components"

In this case the question is divided in components. A component is an expression delimited by a preposition. Therefore, a question Q with m prepositions is represented by a set of components C = {c1, c2,..., cm+1}. Each component ci is a substring (subset of words) of the original query.

Some examples of this kind of query reformulations are:

("*received the Nobel Prize*" "*of Peace*" "*in 1992*")

("*in 1992 received the Nobel Peace Prize*")

(d) Fourth reformulation: "components without the first word"

In order to construct this set of reformulations we eliminate the main verb of the question (commonly expressed by the word w1), and then we apply the method of reformulations by components. Some examples of these reformulations are:

("in 1992 *the Nobel Peace Prize*")

("*the Nobel Prize*" "*of Peace*" "*in 1992*")

C. Search Engine

Searcg Engine is one of the most important parts of our system since our knowledge base is the web. The quality of answers depends on high-quality precise documents. We extract those web pages, which have all keywords in the same sentences. Google is also using the similar search strategy.

D. Snippet Extractor

This module extracts answer snippets from the documents returned by the Search Engine. We have adopted a way similar to what Google does for web pages. We extract snippets containing lines that have *all* the query words/

phrases and a line before and after that line. This is similar to Google's approach of having a default *AND* condition between keywords for retrieving documents. We enforce this condition for extracting snippets and have found that it leads to very high accuracy than allowing lines which either do not have all keywords or have keywords scattered over different lines.

E. Answer Ranker

Since the documents retrieved form the Web are automatically ranked [17] by the search engine on their relevance to the Query. We believe that the possible sought after answers could be found within the first few retrieved documents. Therefore, the system analyzes only the first 20 HTML Web pages from the thousands retrieved. From those sentences, the ones which contain keywords from the query are extracted and ranked according of the number of keywords from the query that they submit.

4. Current Implementation

We have successfully implemented first two phases, keyword selection phase and Synonyms detection Phase of our proposed architecture. These phases are having accuracy of about 70%. Remaining 30% of mistakes, are due to inconsistencies or syntax mistakes in Bilingual Punjabi dictionary. we have calculated the accuracy, by randomly picking the 1000 words from Bilingual Punjabi dictionary, and then calculated the accuracy for each of these 1000 words. Then we have taken the average of accuracies for these 1000 words and that is our final accuracy.

The accuracy graph for the first two phases is as follows:

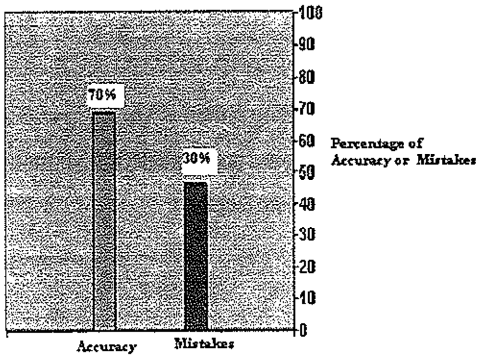

Figure3. Accuracy of Results for First Two Phases

The main table, that we have used, in this project is, pbi_to_eng. The main parameters of table pbi_to_eng are as follows:

Table1. Sub field details of table pbi_to_eng

Name	Data Type	Size
Word_name	varchar	24
Type	varchar	32
Meaning	varchar	254

This table is used to store Bilingual Punjabi Dictionary. The first field is Word_name in Punjabi language. The second filed is type of word like noun, verb, adjective etc. The third field is meaning of word in English language for example:- for Punjabi word ਚੰਗਾ ---- nice, good, fine

This table is containing around 30,000 words of Punjabi language. The approximate time taken by the Dictionary method, for discovering the similar words is around 30-100 seconds. We have implemented these two phases using windows 2000 professional operating system, Visual Basic6 as programming language, SQL Sever7 as Backend Server and Hardware Configuration is as follows:

Central Processing Unit (CPU)-Pentium III 700 MHz Random Access Memory (RAM)-192 MB Hard Disk-40GB Mother Board-Intel Desktop Board CA810E If we will use a processor of slow speed and small RAM then only speed of project will be suffered.

Results of Synonyms detection phase for Punjabi word ਮੁਕਾਬਲਾ are as follows :

Table2 for Synonyms Detection Phase

ਮੁਕਾਬਲਾ	ਹੋੜ	ਦੌੜ	ਪ੍ਰਤਿਯੋਗਤਾ	ਟਾਕਰਾ
ਟੱਕਰ	ਮੋੜਨਾ	ਭਿੜਨਾ	ਭੇੜ	ਝਗੜਨਾ
ਜੋੜ	ਮੇਚ	ਟਕਰਾਉਣਾ	ਲੜ	ਝੜਪ

In future first of all, we will finish the implementation of all phases of our proposed architecture. Currently we are implementing Query reformulation phase. Results can be improved by removing inconsistencies or syntax mistakes in Bilingual Punjabi dictionary.

5. Future Plans And Summary

In future, we will finish the implementation of all phases of our proposed architecture. some improvements are also possible like, applying some more reformulations, applying word stemming techniques, incorporating binary search method in Punjabi language synonyms detection and using more powerful answer ranking techniques etc.

We have proposed this architecture for Punjabi language question answering. This system may be extremely helpful for people of Punjab. Google is a search engine and it will not give you the exact answer for the question typed by you, but it will only list the useful links to web pages and we have to manually explore these web pages. But our proposed architecture will give correct answers for yours questions. Instead of showing useful links as in Google, our system will only give the ranked answers according to their relevance. Good ranked answers are more relevant than lower ranked answers. Our proposed system analyzes only the first 20 HTML Web pages from the thousands retrieved as these are more relevant.

References

[1] Berry Michael .W, *"Automatic discovery of similar words in survey of text mining: Clustering, Classification and Retrieval, Springer Verlag"*, New York, 24-43, 2004, J. Clerk Maxwell, *"A Treatise on Electricity and Magnetism"*, 3rd ed., vol. 2. Oxford: Clarendon, pp 68–73, 1892.

[2] Gurmukh Singh, Mukhtiar Singh Gill, S.S. Joshi, *"Punjabi to English Bilingual Dictionary"*, Punjabi University, Patiala, 1999.

[3] Vishal.and G.S. Lehal, *"Creation of thesaurus from bilingual Punjabi dictionary using text Mining, International conference of Challenges of E-commerce and Networks"*, APIIT SD panipat, India,2005.

[4] Jignashu Parikh, M. Narasimha Murty, *"Adapting Question Answering Techniques to the Web"*, Proceedings of the Language Engineering Conference IEEE, 2002.

[5] Alejandro Del-Castillo-Escobedo , Manuel Montes-y-Gómez, Luis Villaseñor-Pineda, *"QA on the Web: A Preliminary Study for Spanish Language"*, Proceedings of the Fifth Mexican International Conference in Computer Science IEEE, 2004.

[6] Andrea Andrenucci, Eriks Sneiders, *"Automated Question Answering: Review of the Main Approaches"*, Proceedings of the Third International Conference on Information Technology and Applications (ICITA) IEEE, 2005.

[7] Oliver Mason, "*QTAG-A portable probabilistic tagger*", Corpus Research, the University of Birmingham, U.K, 1997.

[8] R. Baeza, B. Ribeiro, *"Modern information retrieval"*, ACM Press, New York, Addison-Wesley, 1999.

[9] J. Allan, M. Connel, W. Croft, F. Feng, D. Fisher and X. Li, *"INQUERY and TREC-9"*, TREC-10, 2000.

[10] G. Cormack, A. Clarke, C. Palmer and D. Kisman, *"Fast Automatic Pasaje Ranking (MultiText Experiments for TREC-8)"*, In TREC-8, 1999.

[11] M. Fuller, M. Kaszkiel, S. Kimberly, J. Sobel, R. Wilson and M. Wu, *"The RMIT/CSIRO Ad Hoc, Q&A, Web, Interactive, and Speech Experiments at TREC-8"*, In TREC-8, 1999.

[12] L. Hirshman and R. Gaizauskas, *"Natural Language Question Answering: The View from Here"*, Natural Language Engineering, Vol. 7, 2001.

[13] J. Chen, A. Diekema, M. Taffet, N. McCracken, N. Ozgencil, O. Yilmazel and E. Liddyl, *"Question answering:CNLP at the TREC-10 question answering track"*, In TREC 2001, 2001.

[14] E. Hovy, L. Gerber, U. Hermajakob, M. Junk and C. Lin, *"Question answering in Webclopedia"*, In TREC-9, 2000.

[15] E. Hovy, U. Hermajakob and C. Lin, *"The use of external knowledge in factoid QA"*, In TREC 2001, 2001.

[16] E. Brill, J. Lin, M. Banko, S. Dumais and A.Ng, *"Data-intensive question answering"*, In TREC 2001, 2001.

[17] Calkin AS. MONTERO and Kenji ARAKI, *"Information-Demanding Question Answering System, Intematiorial Symposium on Coinmumcations and Information Tcchnologes ISCIT"*, Japan, *26*- 29, October 2004.

[18] Vishal.and G.S. Lehal, Text Mining Approach for Punjabi Language Synonyms DetectionUsing Bilingual Punjabi Dictionary and Corpus, International Journal of Systemics, Cybemetics and Informatics (IJSCI),India,60-65, January 2007.

Author's Biography

Vishal Gupta is Lecturer in Computer Science & Engineering Department at University Institute of Engineering & Technology, Panjab university Chandigarh. He has done MTech. in computer science & engineering from Punjabi University Patiala in 2005. He was among university toppers. He secured 82% Marks in MTech. Vishal did his BTech. in CSE from Govt. Engineering College Ferozepur in 2003. He is also pursuing his PhD in Computer Sc & Engg. from Panjab University Chandigarh. Vishal is devoting his research work in field of Natural Language processing. He has developed a number of research projects in field of NLP including synonyms detection, automatic question answering and text summarization etc. One of his research paper on Punjabi language text processing was awarded as best research paper by Dr. V. Raja Raman at an International Conference at Panipat. He is also a merit holder in 10th and 12th classes of Punjab School education board.

Professor Gurpreet Singh Lehal received undergraduate degree in Mathematics in 1988 from Panjab University, Chandigarh, India, and Post Graduate degree in Computer Science in 1995 from Thapar Institute of Engineering & Technology, Patiala, India and Ph. D. degree in Computer Science from Punjabi University, Patiala, in 2002. He joined Thapar Corporate R&D Centre, Patiala, India, in 1988 and later in 1995 he joined Department of Computer Science at Punjabi University, Patiala. He is actively involved both in teaching and research. His current areas of research are- Natural Language Processing and Optical Character recognition. He has published more than 25 research papers in various international and national journals and refereed conferences. He has been actively involved in technical development of Punjabi and has to his credit the first Gurmukhi OCR, Punjabi word processor with spell checker and various transliteration software. He was the chief coordinator of the project "Resource Centre for Indian Language Technology Solutions- Punjabi", funded by the Ministry of Information Technology as well as the coordinator of the Special Assistance Programme (SAP-DRS) of the University Grants Commission (UGC), India. He was also awarded a research project by the International Development Research Centre (IDRC) Canada for Shahmukhi to Gurmukhi Transliteration Solution for Networking.