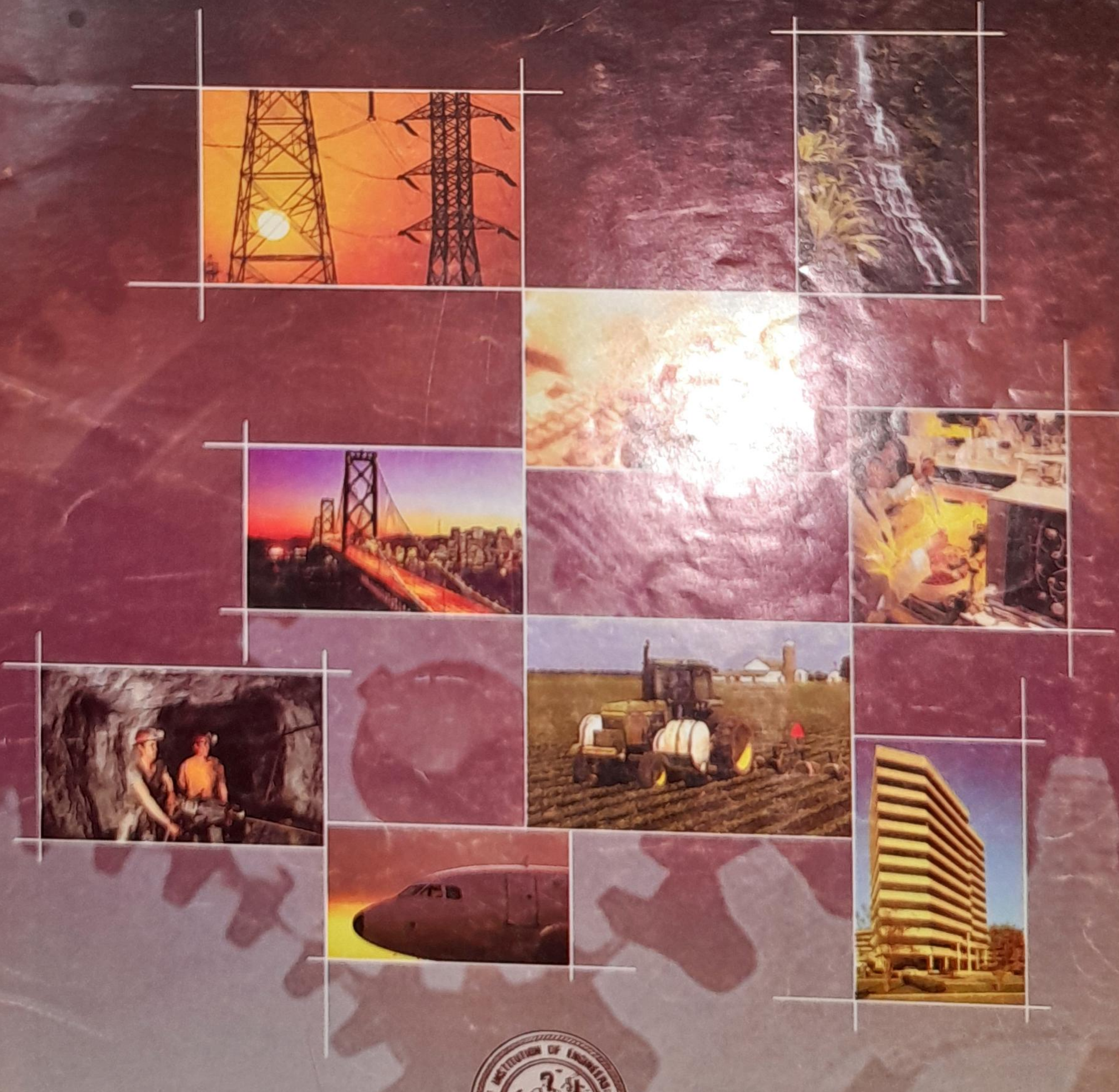
Rs. 100

Journal of The Institution of Engineers (India)

Volume 87, May 29, 2006



Information Extraction from Natural Language Texts

K R Chowdhary, *Member*

Dr V S Bansal, *Fellow*

This paper presents the process of information extraction (IE) from natural language (NL) texts for query answering. The proposed system of IE is implemented in two phases. In the first phase, the NL documents, relevant to the query under consideration are retrieved using theory of fuzzy sets. This is followed by IE for answering the query in the second phase. A special type of grammar, called structured description grammar, has been described with a view to facilitate fast and unambiguous answers for the query. The application of the schemes proposed for IE has been shown in illustrative examples.

Keywords : Information retrieval; Information extraction; Fuzzy thesaures; Question answering; Structured description language

INTRODUCTION

The process of IE for the purpose of answering specific queries presents formidable challenges. This is essentially due to, (i) a large amount of information is available on widely distributed web sites, (ii) a gamut of domain specific as well as general information keeps appearing in news papers, magazines and journals, and (iii) the emergence of rapidly expanding knowledge in the form of e-books.

It is proposed to deal with these challenges by developing an IE system as shown in Figure 1. It may be seen that the process of IE is carried out in two phases. In the first phase called FETCH, documents relevant to the query under consideration are retrieved using information retrieval (IR) technique based on the theory of fuzzy sets[1]. The use of fuzzy set theory helps in dealing with the uncertainty due to incomplete matching between the query and the keywords present in the repository of documents. The FETCH phase is followed by BROWSE phase[2] in which the query and the retrieved relevant documents are parsed in terms of structured description grammar (SDG), which has been specially developed for the purpose of query answering. This method of IE is easy to implement and it provides fast and easily intelligible answer of the query.

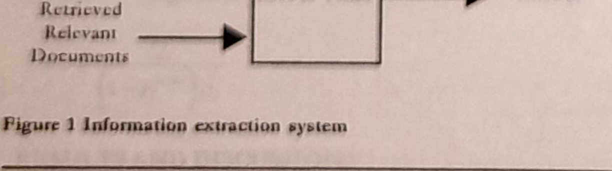


Figure 1 Information extraction system

K R Chowdhary is with the Department of Computer Science and Engineering; while Dr V S Bansal is with the Department of Electrical Engineering, MBM Engineering College, JNV University, Jodhpur 342 011.

This paper was received on August 05, 2003. Written discussion on the paper will be entertained July 31, 2006.

FETCH PHASE — RELEVANT DOCUMENTS RETRIEVAL

Generation of Augmented Query

This method of document retrieval is based on the fuzzy set theory. This is more realistic than those based on classical binary set theory. In fuzzy information retrieval (FIR), the relevance of index terms to individual documents is expressed by relevance relation

$$\mathrm{R} : XY \in [0,1]$$

such that the membership value $R(x_i, y_i)$ specifies, for each $x_i \in X$, $y_i \in Y$, the grade of relevance of index term x_i with document term y_i. Here X and Y are sets of terms, where X represents the query and Y represents the document. The value of these grades range from 0 (no relevance) to 1 (full relevance). The grade of relevance depends on many factors like (i) position of term y_i in the text document, (ii) frequency of occurrence of y_i in the document, (iii) x_i and y_i are terms formed from the same basic stem word and, (iv) y_i is synonym of x_i, then the proximity of meaning of x_i and y_i decide value of $R(x_i, y_i)$.

The criteria (i) and (ii) as mentioned above are user defined and they can be programmed in the implementation according to the user needs. The stem word criteria (iii) requires a table, which helps to locate the stem word for a given word, and then the stem word is substituted in the original text before the retrieval technique is applied. Another important relation for *IR*, based on criteria (iv) is fuzzy thesaurus. A fuzzy thesaurus shows the relationships between pairs of words based on their centrality or degree of relevance. The structure of a fuzzy thesaurus is,

<*WC*1> <*WC*2> <*RD*>

where *WC* stands for word category and *RD* is relationship degree between the words *WC*1 and *WC*2. For a are typical case it can be

war, battle, 1

war, crime, 0.7

When relationship degree is zero, no entry is recorded in the fuzzy thesaurus. The thesaurus can be treated as a reflexive fuzzy relation, say T, defined over X^2. For each pair of index terms $(x_i, x_j) \in X^2$,

$T(x_i, x_j)$ expresses the degree of association of x_i with x_j, such that the degree to which the meaning of the index term x_i is compatible with the meaning of the index term x_j. The role of this relation is to deal with the problem of synonyms among the index terms. The relation helps to identify the relevant documents which otherwise would not be identified in the absence of perfect match between the keywords in the user query and those in the text document.

Various methods can be used for the construction of fuzzy thesaurus. For example, experts in the domain of text can be asked to identify, in a given set of index terms, the pairs of words whose meaning they consider are associated, and provide the degree of association for each pair. In FIR, an inquiry can be expressed in the form of a fuzzy set (say Q) based on the index term X. Then, by composing Q with the fuzzy thesaurus T, a new fuzzy set on X is obtained, which represents the augmented query A, given by

$$A = Q^{\circ} T \tag{1}$$

where $^{\circ}$ is called max-min composition operator. Each term of A can be expressed as,

$$A(x_j) = \text{max-min}\ [\, Q(x_i),\ T(x_i, x_j)\,]$$

where $x_i \in X$, for all $x_j \in X$. The relevance of retrieved documents, expressed by F, is then obtained by composing the augmented query, expressed by the fuzzy set A, with the relevance relation R, as given by equation (2).

$$F = A^{\circ} R \tag{2}$$

Fuzzy Ordering of Retrieved Relevant Documents

The application of the technique discussed above is now illustrated here. A set of three different documents D_1, D_2, and D_3 as given in the Appendix are considered. Each document is represented by a set of keywords, which closely identify the corresponding document. The keywords have been chosen based on the multiplicity of their occurrence in the document. The following four questions (queries) are considered whose answers lie, possibly, in the set $\{D_1, D_2, D_3\}$.

Q_1: Which king had liberal policy towards the religion?

Q_2: Who was the queen of Jahangir?

Q_3: What architectures were built by Shah Jahan?

Q_4: Which Mughal kings had interest for arts?

These questions can be expressed in terms of their keywords, also called index terms. In the case of Q_1 the keywords are : q_1 = king, q_2 = liberal policy, q_3 = religion. Therefore,

$$\begin{aligned} Q_1 &= \{q_1, q_2, q_3\} \\ &= [\,0.9,\ 0.6,\ 0.7\,], \text{(a vector representation of } Q_1) \end{aligned}$$

Here 0.9, 0.6, 0.7 are called the centralities of q_1, q_2, q_3, respectively. The centralities indicate the presence of certain qualities, whose computations are modelled as computation of fuzzy membership degree. The relevant part of the fuzzy thesaurus T_1 restricted to support Q_1, is given by the matrix

$$T_1 = \begin{array}{c} \\ q_1 \\ q_2 \\ q_3 \end{array} \begin{array}{c} \begin{array}{cccccc} q_1 & q_2 & q_3 & q_4 & q_5 & q_6 \end{array} \\ \begin{bmatrix} 1 & 0 & 0.5 & 0.9 & 0.1 & 0.7 \\ 0 & 1 & 0.4 & 0.2 & 0.9 & 0.3 \\ 0.5 & 0.4 & 1 & 0.6 & 0.4 & 0.8 \end{bmatrix} \end{array} .$$

where q_4, q_5, q_6 are the terms : ruler, open, discipline respectively. The composition of $Q_1{}^{\circ}\, T_1$ results in fuzzy set A_1, which represents the augmented query. Its vector form is given as

$$\begin{aligned} A_1 &= \text{max-min}\ [Q_1{}^{\circ} T_1] \\ &= \begin{bmatrix} 0.9 & 0.6 & 0.7 \end{bmatrix}^{\circ} \begin{bmatrix} 1 & 0 & 0.5 & 0.9 & 0.1 & 0.7 \\ 0 & 1 & 0.4 & 0.2 & 0.9 & 0.3 \\ 0.5 & 0.4 & 1 & 0.6 & 0.4 & 0.8 \end{bmatrix} \\ &= \begin{bmatrix} 0.9 & 0.6 & 0.7 & 0.9 & 0.6 & 0.7 \end{bmatrix} \end{aligned}$$

Assume that part of the fuzzy relevance relation which is restricted to support A_1 is given by the R_1. Each index term $q_1, q_2, q_3, q_4, q_5, q_6$ is related to these documents D_1, D_2, D_3 by some degree.

$$R_1 = \begin{array}{c} \\ q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \\ q_6 \end{array} \begin{array}{c} \begin{array}{ccc} D_1 & D_2 & D_3 \end{array} \\ \begin{bmatrix} 0.1 & 0.1 & 0.1 \\ 0.5 & 0.2 & 0.1 \\ 1 & 0.1 & 0 \\ 0.3 & 0.4 & 0.2 \\ 0.6 & 0.3 & 0 \\ 0.8 & 0.3 & 0 \end{bmatrix} \end{array}$$

Next, by composing $A_1{}^{\circ} R_1$, by equation (2), results in a fuzzy set F_1, which characterizes the fuzzy relevance of retrieved documents.

$$F_1 = \text{max-min}\ (A_1{}^{\circ} R_1) = [\,0.7 \ \ 0.4 \ \ 0.2\,]$$

The fuzzy retrieval matrix F_1 shows that out of D_1, D_2, D_3, the document D_1 has highest relevance for the query Q_1. Therefore, the answer to the question, 'Which king had liberal policy towards religion?', lies mostly in the document D_1.

Similarly, the queries Q_2, Q_3, and Q_4 are processed and the results are as given below:

$Q_2 = \{q_1, q_2\}$ = {queen, Jahangir} = [0.8 0.9], q_3 = empress, q_4 = king, q_5 = kingdom, F_2 = [0.3 0.9 0.2]. F_2 shows that answer of Q_2 lies mostly in the document D_2.

Q_3 = {architecture, Shah Jahan} = [0.7 0.9], q_3 = fort, q_4 = museum, q_5 = mosque, q_6 = temple, F_4 = [0.4 0.4 0.9], which shows that the most relevant document for Q_3 is D_3, and therefore it has highest possibility of providing the answer of Q_3.

Q_4 = {king, art} = [0.8 0.7], q_3 = painting, q_4 = beauty, F_5 = [0.1

0.7 0.7], which shows that the relevant documents for Q_4 are both D_2 and D_3. Therefore the answer of Q_4 lies in D_2 and D_3. The possible solution is to find the answer from D_2 as well from D_3, and then merge them together.

Having located the most probable documents, which are expected to carry the answer for the query under consideration, the next phase is to find the answer through IE from the fetched document(s) using the technique as discussed here.

BROWSE PHASE INFORMATION EXTRACTION

In one of his famous poems the author Rudyard Kipling[3] wrote

> 'I keep six honest serving-men
>
> They taught me all I knew;
>
> Their names are *What* and *Why* and
>
> *When*
>
> And *How* and *Where* and *Who*.
>
> I send them over land and sea...'

Clearly, these interrogations are seeking answers for questions in the form of who, what, when, where, why, and how. If taken in the present context, their computer based processing can solve major problems of IE from NL texts.

For the extraction of information from NL texts, first the text needs to be represented in some structured form to help in precisely locating the desired information in the text. The classical method, which is one of the most commonly used mathematical modelling system for constituent structure in English and other natural languages (NL) is context free grammar (CFG)[4,5]. A CFG consists of a set of lexicons of words and symbols, and production rules, each of which expresses the way the symbols of the language can be grouped or ordered together. It is expressed as,

$$G = (N, T, S, P)$$

where G is CFG, N is set of symbols called non terminals, T is set of terms in the language sentences, S ($S \in N$) is start symbol, and P is set of production rules. Commonly used production rules for construction of English language sentences and non terminals are shown in Figure 2.

Figure 3 shows the construction of the sentence 'Akbar followed a liberal policy for religion', through parsing, making use of these rules. When the parsing tree of question and answer sentences are matched, the corresponding matching part of question (who, what, etc), in the answer carrying sentence can be determined, which would provide the answer.

NP → PropN | N | Det Nom | Nom | NP PP | Pron
Nom → N | N Nom | Adj N
S → NP VP
S → Aux NP VP
S → VP
VP → V
VP → V NP
VP → V NP PP
VP → V PP
PP → Prep NP

Figure 2 Context free grammar for English language

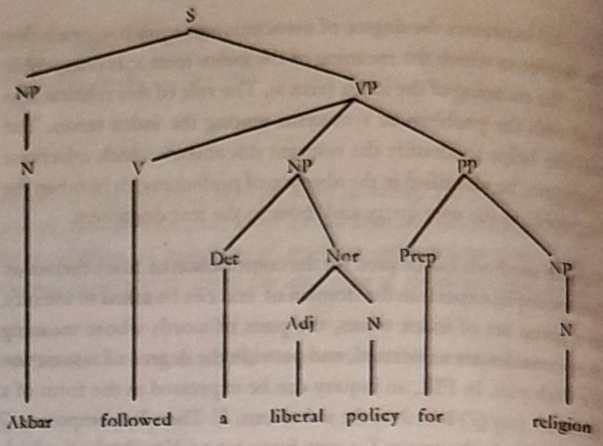


Figure 3 Parse-tree for the sentence 'Akbar followed a liberal policy for religion'

Sentence Representation using Structured Description Grammar

A method is proposed for information representation, and extraction based on structured description grammar (SDG). In this, the text sentences are mapped into a transition graph or state diagram, as shown in Figure 4. It shows seven types of sentences labeled 1, 2, ..., 7, based on the number of wh-pronouns and their positions in the sentences. It may be seen that the structure explicitly shows the interrogations who, what, where, when and why. The positions of these interrogations are invariant. Following examples demonstrate that English language sentences structures map into the transition diagram of SDG.

1. Akbar | followed a liberal policy for religion.

   Who | What

2. Jahangir | married Nur Jahan.

   Who | What

3. Red fort | is | at Delhi.

   Who | What | Where

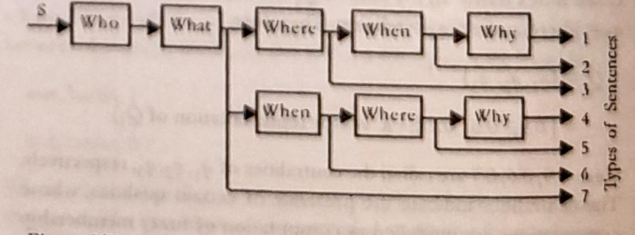


Figure 4 Transition diagram of sentence structure

4. Shah Jahan | built Taj Mahal | during his rule | at Agra | in the memory of queen Mumtaj.

Who | What | When | Where | Why

The sentences 1 to 4 as given can also be generated with the help of transition diagram as shown in Figure 4. It can be easily verified that the sentences 1 to 4 are of types 7, 7, 3, 4, respectively.

It may be seen that the structures of the sentences in SDG are simpler in comparison to CFG. Further, in SDG, i) a query is allowed to carry more than one sub queries, ii) sub answers for a query which are distributed in a single document can be aggregated to fill up the slot like structures in the transition diagram to generate the answer, and iii) sub answers for a query which are distributed in multiple documents can also be aggregated to fill up the slot like structures in the transition diagram to generate the single answer.

Question Answering System

Once the relevant document is retrieved for the given query, the question answering system (QAS) extracts the answer from the relevant document[6]. The various modules of QAS are shown in Figure 5. Its inputs are the question and the most relevant retrieved documents. The question processing is in three stages parsing, keywords expansion, and named entity (NE) tagging. The NE identifies the type of entity as shown in Table 1. The parser identifies the asking point (who, what, when, etc) in the question. The terms in the question sentence are expanded through their synonyms so that there is a maximum matching between the terms in the question and those in the potential answer sentence. Each input sentence from the relevant document is also first parsed by the parser , and then the NEs are tagged. Finally, the matching is performed between the keywords in the question and in text sentences from the document. At the end, the answer of the query is extracted.

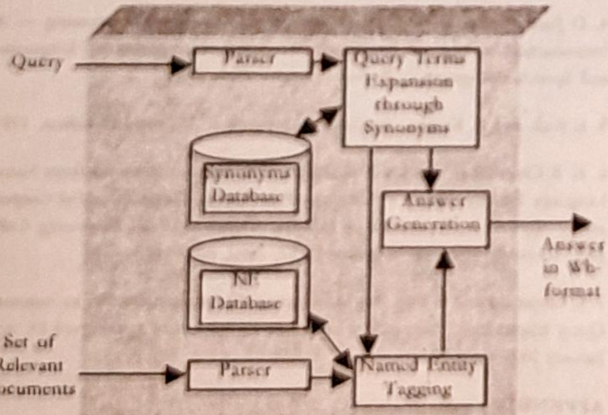


Figure 5 Question answering system (QAS)

Table 1 Wh-pronouns and their answer types

Wh-pronouns	Answer Type (token of NE)
Who	PERSON
What	ACTION, STATUS
When	TIME
Where	PLACE
Why	REASON

Information Extraction for Answering Queries

Once the document through fuzzy set based IR is decided, it is required to determine the sentence in the relevant document, which contains the answer. The following discussion demonstrates how to locate the most probable sentence, which is likely to carry the information for answering the query.

Let the question considered is, 'Which king had liberal policy towards the religion?'. Here the NE or asking point is 'which'. To provide matching between the terms in the question and answer sentence, all the terms in the question sentence are supplemented by their possible synonyms[7]. This is indicated by the block 'question terms expansion through synonyms', in Figure 5. Therefore, after expansion, the question becomes as follows.

Asking point : { NAME }

Keywords : { king | rule | empire, liberal | open, policy | rule | norms, religion }

While creating the keywords list, the stop words — a, the, an, on, in, towards, etc are dropped so that the matching is carried out for nouns, and verbs only. Here, the asking point 'which' is of the type NAME as it stands for the name of somebody or some thing. The set of words from the asking-point as well as from the keywords set are matched against the potential answer sentence or text passage in the selected relevant document. For selection of best answer sentence, the density of the matching words in the answer sentence is the major consideration. The matching from the relevant text document is carried out by sentence matching, or by keeping a fixed window size of text in the text document.

Let us now answer the questions Q_1 to Q_4 for which the relevant documents have been determined.

Q_1: Which king had liberal policy towards the religion?

Through FIR, it has been determined that the relevant document corresponding to this query is D_1. Next, using the keyword matching techniques as discussed above, the potential answer carrying sentence is determined, which is, 'Akbar followed a liberal policy for religion'. The 'which king' has same implication as 'who'. The questions and answers are given below, as per sentence type 7 of transition graph.

Who | What

Q_1: Which king | had liberal policy towards religion?

Ans$_1$: Akbar | followed a liberal policy for religion.

By comparing the transition graphs for the question and the answer sentence, it is easy to see that the value of who block in the answer is 'Akbar', which is answer for Q_1.

Q_2: Who was the queen of Jahangir?

The relevant document corresponding to this query is D_2, and the potential answer carrying sentence is 'Jahangir married Nur Jahan'. Through the synonyms table, the word 'queen' in the question sentence is changed to 'married'. Similarly, the syntax of the sentence is changed to conform the syntax of the question. Finally, the block wise matching of question and answer sentence is given by,

Who | What

Q_2: Who | was married to Jahangir?

Ans_1: Nur Jahan | was married to Jahangir.

This shows that the answer of the question Q_2 is Nur Jahan.

Q_3: What architectures were built by Shah Jahan?

In the question above, Shah Jahan appears in place of object. Therefore, the sentence is converted into active voice, as per the requirements of syntax diagram of SDG given in Figure 4, from the present passive voice, before the matching is carried out. The answer carrying sentences, in the form of sub answers, are distributed in the document. The question and corresponding potential answer carrying sentences, determined by text matching are given here.

Who | What

Q_3: Shah Jahan | built what architectures ?

Ans_1: He | built Taj Mahal at Agra.

Ans_2: He | also built the fort at Delhi, named as, Red Fort, and a mosque named as Jama Masjid.

When the question Q_3 is matched with Ans_1, and Ans_2, the answers are:

- Taj Mahal at Agra
- Fort at Delhi, named as Red Fort
- A mosque named as Jama Masjid

Q_4: Which king had interest for art?

Here, the documents D_1 and D_2 are equally relevant, having a degree of relevance of 0.7 in each. The potential answer carrying sentences from these documents are now given along with the question.

Q_4: Which king | had interest for art?

D_2: Ans_1: Jahangir was a highly learned man but he did not have the intellect of his father.

He | was lover of art and justice.

D_3: Ans_2: Under Shah Jahan | the art of painting also developed significantly.

D_3: Ans_3: Shah Jahan | gave liberal patronage to artists.

The wh-pronoun which here is similar to who, and therefore the transition diagram part of Who is used here. In $D_2 : Ans_1$, the word 'Jahangir' is subject and the word 'art' appears in the sentence, hence 'Jahangir' is one answer. In $D_3 : Ans_2$, 'Shah Jahan' is the subject and the word 'art' also appears in it, hence 'Shah Jahan' is another answer. Similarly, $D_3 : Ans_3$, results to an answer 'Shah Jahan'. Merging the above three sub answers, the complete answer is,

Jahangir,
Shah Jahan.

In case, the highest relevance document does not contain the complete information required for answering the question, the document of next lower relevancy is taken for extracting the answer. The matching of keywords in the question with those in the potential answer sentence are ranked, and the highest ranked text phrase is selected for the purpose of answering. The ranking is determined by the following criteria, (i) primary ranking is based on the percentage of question keywords found in the potential answer sentence, (ii) secondary ranking is based in the order the keywords in the answer appear in comparison to the order of those in the question and (iii) whether the keywords match exactly or their variants match in the question and answer sentence.

CONCLUSIONS

The paper presents a new process of question-answering from a repository of NL texts. The process is carried out in two phases. In the first phase the relevant document is chosen based on theory of fuzzy sets, and the second phase generates the answer of the query from the retrieved relevant documents. It may be seen from the illustrative examples that the technique outlined in the paper can be used for the generation of complete answer of a query from one or more relevant documents.

REFERENCES

1. K R Chowdhary and V S Bansal. 'Fuzzy Logic-based Information Retrieval'. *Proc National Seminar on Algorithms and Artificial Systems, Department of Computer Science, Madras University, Allied Publishers*, New Delhi, January 30-February 1, 2003, pp 297-306.

2. M Lalmas. 'Dempster-Shafer's Theory of Evidence Applied to Structured Documents: Modelling Uncertainty'. *Proc the Twentieth Annual ACM Conference on Research and Development in Information Retrieval*, Philadelphia, PA, USA, 1997, pp-110-118.

3. *http://www.kipling.org.uk/poem_fru.htm*.

4. D Jurafsky and J H Martin. 'Speech and Language Processing — An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition'. *Pearson Education Press*, 2002.

5. E Rich and K Knight. 'Artificial Intelligence'. *McGraw-Hill Edition*, 1991.

6. K R Chowdhary and Dr V S Bansal. 'Information Retrieval from Natural Language Text by Query Processing'. *Proc National Conference on Soft Computing and its Applications, Department of Electrical Engineering MBM Engineering College, JNV University*, Jodhpur, February 15-17, 2002, pp 7- 12.

7. C Carpineto and B Bigi. 'An Information-Theoretic Approach to Automatic Query Expansion'. *Proc ACM Transactions on Information Systems*, vol 19, no 1, January 2001, pp 1-27.

APPENDIX

$Keywords_1$ = {Akbar, religion}

Document D_1: Akbar was a great administrator. His administration strove for the welfare of his people irrespective of their religious beliefs. He had council of ministers and high officers to assist him. The ministers were heads of different departments. The revenue source for the Mughals were from land and trade. Akbar followed a liberal policy for religion. He believed that every religion was basically good. He used to hold talks with the leaders of various religions. Akbar found that basically all religions taught similar things. Thus, he incorporated the principles of all the religions and found a new faith, which he called as Din-e-Illahi. This was more a code of morale conduct than religion. It was his firm belief that all religions are same, therefore, he did not seek forced conversion to Islam.

Keywords$_2$ = (Jahangir, art and justice, trade)

Document D$_2$: Jahangir was a highly learned man but he did not have the intelligence of his father. He was lover of art and justice. A golden chain with bells attached, were tied to the wall of his palace. Any one who pulled the chain was heard and was accorded justice. Jahangir married Nur Jahan. She was not only peerless in beauty, but also a highly intelligent and cultured lady. She was fond of music, poetry, and painting. During Jahangir's time many European traders started coming to India and trade started with them.

Keywords$_3$ = (Shah Jahan, Mughal architecture, art)

Document D$_3$: Shah Jahan was a great general of his time. Shah Jahan brought glory to the Mughal architecture. He was a man of finest tastes. He built Taj Mahal at Agra, the new city of Delhi, which he made his capital. He also built the fort at Delhi, named as, Red Fort and a mosque, named as Jama Masjid. He built finest gardens. Under Shah Jahan, the art of painting also developed significantly. Shah Jahan gave liberal patronage to artists.