

# Answering FAQs: An Intelligent Approach for Extracting Answers for Queries From Subject-Oriented Multidocuments

Shivendra Prasad Tiwari, Project Associate  
Dept of Comp. Sc. & Engg., IIT:Delhi, Hauz Khas New Delhi-110016, India.  
stiwari@cse.iitd.ernet.in

## Abstract

From the ocean of text data, Extracting the important and query based text is a big NLP problem. It is a difficult task, as it requires mining text content in shortest answer length accurately and efficiently. This paper describes an attempt toward solving this problem with an intelligent approach. The proposed idea is based on the human intelligence-once a document is seen; it takes less time to answer a query from that particular document. When system becomes familiar with the document by reading more than once, it takes less time and effort to answer i.e. system uses a self-developed artificial memory. A function  $\sigma(q, D)$  is defined as a sequence of steps to compute the answer  $\Lambda$  for query  $q$  from the given set of documents  $D$ . System takes a query and set of documents as input. On the basis of the weight of keyword term(s) in query the important sentences are obtained by calculating cohesiveness with query, and then categorized to get and order of relevance by using fuzzy membership function. The most cohesive sentences are given as possible answer options to the user, to get answer  $\Lambda$ . The intelligent module generates an index table of keywords with page# & Para# and marks the contextual text in document; learns the queries, corresponding cohesiveness with answers, which is used for answer prediction to the next query .

**Keywords-** FAQ, Query, Multidocument Summarization, NLP, Case Frames, Fuzzy Logic etc.

## 1. Introduction

**Query-based summaries** are produced in reference to a user query while generic summaries attempt to identify salient information in text without the context of a query. The difference between single- and multi-document summarization (SDS and MDS fig. 1) is quite obvious, however some of the types of problems that occur in MDS are qualitatively different from the ones observed in SDS: e.g., addressing redundancy across information sources and dealing with contradictory and complementary information.

Query-relevant summarization (QRS) aims to provide a more effective characterization of a document by accounting for the user's information need when generating a summary. For the purpose of training people a large number of books, manuals and tutorials are released. Gathering answers from such documents efficiently is a need of today and a big topic of research.

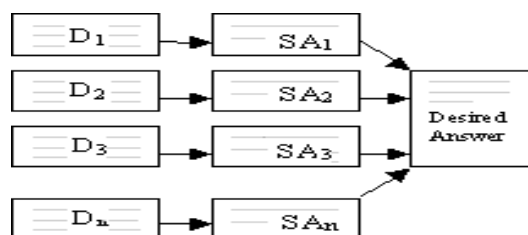


Figure 1. Multi document Summarization

In the proposed system Fig 2, A finite set of subject oriented documents and a finite collection of beginning questions called FAQ are taken as problem and the idea is to generate concise answers for each query accurately & efficiently with AI approach. A set of subject oriented documents and a query is given to the system which preprocesses the query with transformation of documents into a simple text file, as data may be a scanned file, an html document or in some different forms. System reads the document for each query, and an index of keywords is generated which will help to find the answers next time for new query. On the basis of the keywords found in query, weights of the sentences are calculated in each of the documents. Zadeh introduced Fuzzy set in 1965 to represent data and information processing, non-statistical uncertainty. Fuzzy membership functions are used to categorise the sentences into “Most cohesive”, “Relevant” and “Irrelevant” sentences. The most cohesive sentences are taken as sub-answers of the query and are sent to user as optional answer to remove the redundancy. Now the union of sentences is derived from resulted set of sub-answers  $\Lambda_1, \Lambda_2, \dots, \Lambda_k$  to produce the complete answer.

The paper is divided in various parts. In the first part the system and related fields are introduced; section 2 describes the related previous work in the field of text summarization and recent R&D. Section 3 states problem outline and 4 is taken as a major part of proposed system.

In section 5 & 6 Discussion, System Evaluation, Future work in this area and conclusion is give along with limitations and study reference at last.

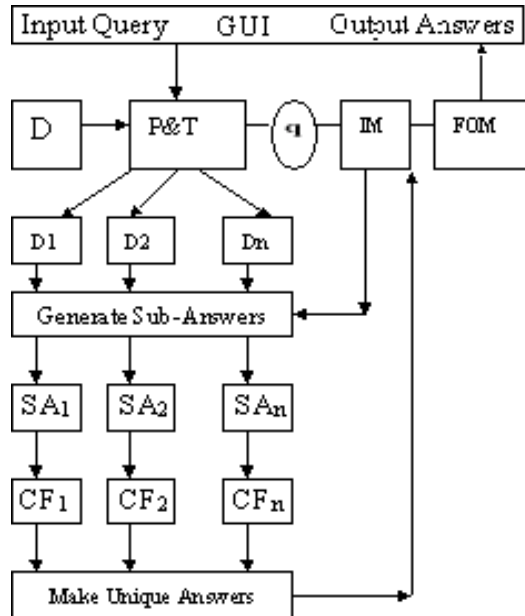


Figure 2. System Block Diagram

## 2. Related Work

In a recent work T.Sakurai et.al.[1] presented a genre-independent method for getting answers to query. The major drawback of the system is that it take answers from most relevant document only whereas a document may not contain a complete answer. It also used statistical methods without any intelligency which requires to whole processing repeatedly for each query. A.Berger et. al.[7] given a method of query relevant summarization using FAQ which was a statistical method. Learning algorithms are also used for summarization and feature extraction [4]. A big effort was made by A. Jatowt [20] to get frequent changes in WWW sites. Liong Chiue et al. [6] presented concept of Query Based Event Extraction along a Timeline in which they tried to solve problem of query based summarization by system that extracts events relevant to a query from a collection of documents, and places such events along a timeline. Each event was represented by a sentence extracted from documents, based on the assumption that “important” events are widely cited in many documents for a period of time within which these events are of interest.

Thus, so far the research carried out and succeeded in goals up to some extent but there is a lake of an intelligent system which should be able to derive concise answers accurately & efficiently, for a finite set of frequent queries.

## 3. Problem Outline

By question answering, we mean a system, which automatically extracts answers from a potentially lengthy document (or set of documents) to a user-specified question  $q$ . Devising a high quality question-answering system would be of great service to anyone lacking the inclination to read an entire user’s manual just to find the answer to a single question. The success of the various automated question-answering services on the Internet underscores the commercial importance of this task.

There are three major [11] problems introduced by having to handle multiple input documents for multi-document summarization and there are some other problems given as below:

1. Formulate query in answer mode.
2. Recognizing and coping with redundancy in the documents.
3. Developing an artificial memory, identifying important differences among documents i.e. finding out an ordered list of documents in the form of degree of relevance.
4. Ensuring Summary coherence &
5. The most important thing is to handle uncertainty in selecting most cohesive answer for the question.
6. Making system intelligent-making an artificial memory.
7. Finding out Accurate answers.

Here in this system, the accuracy & efficiency factors are taken as major Challenges and are tried to solve artificially. Problem is taken as a finite set of subject oriented documents and a finite set of queries out of which it is needed to find out the concise answers corresponding to each particular query. Since it is difficult to solve these problems comprehensively, we focused on all of these problems in separate parts. To recognize the redundancy in summary various redundancy measures are used. Here a method  $\sigma$  is a set of step is to define which can generate answers  $\Lambda$  for query  $q$ .

$$\sigma(q,D)= \Lambda$$

## 4. Proposed System

The proposed system is an effort to remove the complexities in getting answer for a query efficiently & accurately from text in a particular context by using AI & SC methods. The system uses Fuzzy membership instead of making hard coded statistical boundaries for selecting the important sentences and with the intelligent module Indexing-Marking, learning & prediction are made to make the system more

efficient as long as the system is becoming trained and older. It learns the queries and stores the corresponding answers into memory so that the similar new query can be answered with no effort, by just calculating degree of cohesiveness and taking a simple comparison with the older queries. The answering algorithm is divided in four modules: i.e. Preprocessing & Transformation, Intelligent module, Core summary Generation and finally finding Super-Answer  $\Lambda$  by defining following algorithm oriented function.

$$\sigma(q,D)=\Lambda$$

**Algorithm: - Answering FAQs (Defining  $\sigma$ ).**

**Input:-** Set of subject oriented Documents

**Output :-** An answer  $\Lambda$  with very fewer efforts and higher accuracy ; an Intelligent tool to predict answers for queries or FAQs.

**Steps:**

I. **Preprocessing & Transformation of query & Documents:** Query is made answer oriented and documents are made in text file format.

II. **Generate Sub-Answers  $\Lambda_1, \Lambda_2, \Lambda_3, \dots, \Lambda_k$  :** Cohesiveness is calculated by calculating TF, Similarity & Cooccurrence and the Most cohesive sentences are derived to formulate sub-answers by Using Fuzzy Membership functions.

III. **Generate Super-Answer  $\Lambda$  :** All the sentences are collected & categorized into three classes using Fuzzy membership functions say "Most-cohesive", "relevant" and "irrelevant" and then most cohesive sentences are sent to the user as answer options which is collected to formulate Super Answer.

IV. **Make system Intelligent:** System is made intelligent by creating artificial memory, Marking Documents, Learning answers, and making Predictions for new queries.

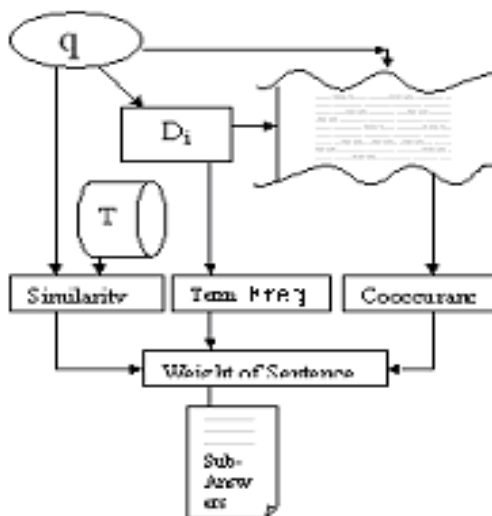


Figure 3. Calculating Weights for Sentences

## 4.1 Preprocessing & Transformation

This section is used to perform two things. First, **formulate the query** by removing irrelevant words, special characters and question oriented terms and then insert some terms instead of wh like question to make the query in answer mode. Second, **Transform documents** in uniformed manner i.e. text file format. The raw documents can be taken from various source like some documents can be scanned from books, some may be in the form of tables, web page documents, research papers in post script form, spatial maps, PDF files, or otherwise can be a front page of news paper. In this step data is transformed by using various methods used in [10]. Here every document is supposed to be in the text form. It is possible that there may be some irrelevant documents. An information retrieval (IR) system can be used to provide a list of retrieved documents that contains query[1] although here subject-oriented multidocuments are supposed to be taken.

**Sub-Algorithm: - Preprocessing.**

**Input:-** Query in String Form And subject-oriented documents may be different format.

**Output :-** An answer oriented query and set of documents as text files.

**Steps:**

i. **Query Formulation:** Tokenize query, remove the question-oriented terms, and replace them with answer-oriented terms, as answer oriented sentence matches well with answers.

ii. **Document Transformation:** Data stored in multiple forms like in text, web page, or may be scanned documents are converted in text file formats [10,20].

## 4.2 Generating Sub-Answers

Deriving sub-answers  $\Lambda_1, \Lambda_2, \Lambda_3, \dots, \Lambda_k$  is based on the research work in [1,18] which uses fuzzy membership for selecting the summaries from all the documents instead of taking only the heavier text from a document among all the retrieved documents.

Table 1. Keywords and Their Weights

id	Keywords	Weight
1	Natural	.854
2	Language	.973
3	Processing	.832
4	.....	.....
.....	.....	.....

**Sub-Algorithm: - Sub-Answers.**

**Input:-** Query And set of Documents.

**Output :-** Sub-Answers.

**Steps:**

- i. Findout noun as keyword terms in query.
- ii. Calculate Term Frequency  $TF_{D_i}(t)$ .
- iii. Compute Similarity  $SQ_{D_i}(q,s_j)$  between query and sentences  $s_j$  in  $D_i$ .
- iv. Findout Co-occurance Level  $CO_{D_i}(t)$ .
- v. Compute the weight with following formula for all the sentences in each documents, i.e. for  $i=1$  to  $n$  and  $j=1$  to  $m$ . Where  $n$  is total number of documents and  $m$  is total number of sentences in  $i^{th}$  document.

$$[W_{s_j}(t)]_{D_i} = TF_{D_i}(t) + SQ_{D_i}(q, s_j) + CO_{D_i}(t) \dots (1)$$

- vi. Generate Index of Keywords & sentence-weight table.

term by similarity  $SQ_{D_i}(q,s_j)$  in  $D_i$  is calculated according to table given in [1]

**4.2.4 Co-occurrence level between query and documents [1,17].**

A term which frequency occurs with the sentence in the document  $D_i$  can be seen as implicitly related to the query. It is assumed that co-occurrence frequency of term(s)  $\{t\}$  provide a measure of how well that term describes the query related topics. The weight  $CO_{D_i}(t)$  by occurrence frequency is defined by:

$$CO_{D_i}(t) = \log_2 ( COF(t) / TF_{D_i}(t) ) \dots (3)$$

$COF(t)$  denotes number of times the term  $t$  co-occurs within 10-word window in the preprocessed/retrieved document [1].

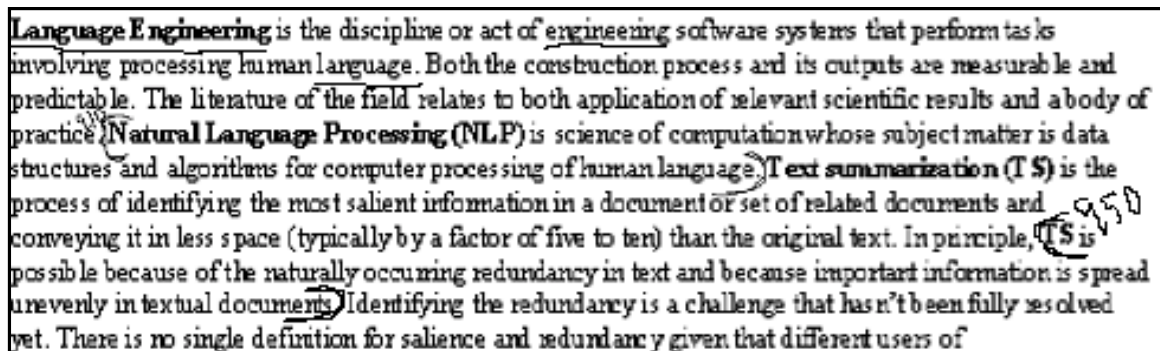


Figure 4. Example Text data for dry run

**4.2.1 Findout noun as keyword terms in query:** The keywords, noun and other noun referring terms are taken as appropriate terms  $\{t\}$ . The importance of term(s) can be calculated as the frequency of that term in the query as in [1].

**4.2.2 Term Frequency (TF) [18,1]:** TF is the sum of number of times the selected term(s) in query occur in a sentence  $s_j$  of the document  $D_i$ .  
 $TF_{D_i}(s_j) = \sum f(t), \{t\} \in s_j, \{t\} \in q \dots (2)$

Here,  $\{t\}$  refers to the set of terms in a sentence that are in query as well,  $s_j$  &  $q$  are  $j$ th sentence in  $i^{th}$  document and query respectively,  $f(t)$  refers to the frequency that a term appears in a sentence  $s_j$  in the document  $D_i$ . The importance of a sentence is determined by the summation of frequency of the keyword term(s). This calculation usually utilizes content words, nouns or keywords as the set of terms.

**4.2.3 Similarity between query and sentences [19,1].** The degree of similarity  $SQ_{D_i}(q,s_j)$  between the terms in query and the sentences is calculated based on distance in the Japanese Thesaurus (T, figure 3) “Bunrui-Goi-Hyo” or any other thesaurus like WorldNet can be used. When that term is grouped into the same category with the query in the Thesaurus, its weight of the

Table 2 Sentences and their weights

S_id	Weight	Category
1	.9863	MC
2	.6076	R
3	.1034	IR
4	.....	.....
.....	.....	.....

**4.3.5 Generate Index of Keywords & sentence-weight table:** Table generation is taken to achieve two objectives.

- **Indexing:** An index table is created in which the page number and paragraph ID with keyword is stored which is further used to find out new answers, table 3.
- **Sentence-Weight table:** Collect All the sentences which have a weight greater than the minimum threshold  $\omega$  and their corresponding weights in the table form as table 1-4.

**4.3 Generate Super-Answer  $\Lambda$**

All the sentences are collected & categorized into three classes say Most cohesive, relevant & irrelevant and then most cohesive sentences are either converted to an unique form by using method “Case Frame Structure” [21] or are given to the user as options to get answer; and then Make union of sentences to reduce redundancy.

The resulted collection of sentences presents a Super-Answer.

**Algorithm: - Super-Answer.**

**Input:-** Sentence-Weight Table.

**Output :-** Desired Answer for query q and

**Steps:**

**i. Classification of Sentences using Fuzzy membership function.**

Classification of sentences is made by using algorithmic method for generating membership functions and fuzzy production rules; the method includes an entropy minimization for screening analog values. Membership functions are derived by partitioning the variables into the desired number of fuzzy terms, and production rules are obtained from minimum entropy clustering decisions. Here three classes of sentences are supposed i.e. Most Cohesive, Relevant & Irrelevant Sentence [22].

**ii. Collect the most-cohesive sentences.** The “Most Cohesive sentences” are collected together and the remaining sentences are saved by labeling their classes.

**iii. Provide Options of sub-answers to user.** This method is a result of one of the latest research [21], in which the sentences can be converted into an unique intermediate form called “Case Form” and hence all the collected sentences can be found in their unique form. As implementing “Case Frame” is a big task therefore the sub-answers can also be given to the users to choose for the purpose of the removing redundancy.

**iv. Collect Unique Sentences.** Collect All the sentences which are unique among the “Most Cohesive” sentences. The resulted sentences(s) is the desired answer  $\Lambda$  for the query q.

**v. Make the sentence in the form of proper output by using templates as method in[5].**

#### 4.4 Make System Intelligent

The intelligent module does three major things, Marking the text, Learning & Experiencing the questionaries and finally making predictions of answers without going into the document in detail. It learns questions and corresponding answer so that if some query comes which was answered in past can be responded with no effort and by just comparing the degree of cohesiveness for the current query with the stored queries.

**Sub-Algorithm: - Intelligence.**

**Input:-** Query and A set of documents.

**Output :-** Marked Document, All the produced queries and Answers are learnt.

**Steps:**

**i. Indexing & Marking**

This is a concept in NLP, which is just inherited from human intelligence; summarization and

learning strategy for reading books and marking answers by writing just answer numbers with boundary marking through pencils and whenever the questions comes, there is no need to read the whole book and just go to the position where the answer number is marked. Here there is the same case with a slight difference. Here there is no question sequence and any question can come in any order. This problem can be solved by inserting unique number or symbol to identify the answers for a particular question. Here an index is created which stores the paragraph id, page id and the containing keywords i.e. table-3.

**ii. Learning [7]**

In the learning portion of the system, there is an efficient learning algorithm of Neural Networks, which learns the queries and there answers with the cohesiveness among them.

**iii. Prediction**

When the system is trained the it becomes older it becomes more smart. The Idea is that once a query has been answered by processing text then why it should go for again processing. Means why not it should see the previous questions? System makes a lookup in the previous question - answer - cohesiveness database and use the following if..else rule:

**IF  $Wq(q_{old}) \geq Wq_{old}(\Lambda_{q_{old}})$  THEN  $\sigma(q,D) = \Lambda$   
ELSE  $\sigma(q,D) = \Lambda'$  .....(4)**

Table 3. Index of keywords

Pg#	Para#	List of Keywords
1	1	NLP, TS, Science
1	3	Language, Engineering
2	3	Modeling, Technology
3	4	...
...	...	...

Where  $Wq_{new}(q_{old})$  is the cohesiveness between some old query  $q_{old}$  and the input query;  $Wq_{old}(\Lambda_{q_{old}})$  is the cohesiveness between  $q_{old}$  and its answer  $\Lambda_{q_{old}}$ ;  $\sigma(q,D)$  is Answering function and  $\Lambda$  &  $\Lambda'$  are two distinct answers.

If the condition is true then there is no need of processing text and just produce the same answer otherwise system processes only that part of the document which has the keywords in query according to keyword index and/or which was labled as “Relevant” and “Irrelevant” in Algorithm (4.2.ii). Which will also beneficial to make process fewer. As the system is being used day-to-day it becomes more smart. Finally according to the proposed prediction module there will a moment come when all the part of the document will be learnt to the system and it will be able to response with no effort.



**Table 4. Query-Answer-cohesiveness Table**

SNo	Query	Answers	Cohesiveness
1	What is NLP?	NLP is a science of language which provides to manipulate human language.	0.9124
2	What are Major Applications of NLP?	Major application of NLP are understanding documents, IR, text mining, summarization etc.	0.8535
3	What is Text Summarization	Text summarization is method of extracting the basic theme and information of the document.	0.9745
.....	.....	.....	.....

**5. Example**

In order to make scene of the system clearer here one simple text in Figure-4 is taken as an element of a set of documents and the system is used to dry run. Suppose the query is given “What is natural language processing?”. The query and document both are sent to preprocessing & Transformation module. The query is transformed in to answer mode as “Natural language processing is” and the document is transformed into the text file if they are in different format. Here the document is taken as the text file so it remains same. The query is sent to the Intelligent module and the cohesiveness is calculated between query and older queries; Now there are two cases, Either the cohesiveness of queries will be greater or equal to that of between older query and the corresponding answer or the cohesiveness will be less from that of. Whether further processing is needed to the system can be decided by using following equation:

$$\text{IF } Wq(q_{old}) \geq Wq_{old}(\Lambda q_{old}) \text{ THEN } \sigma(q,D) = \Lambda \\ \text{ELSE } \sigma(q,D) = \Lambda' \quad \dots(5)$$

If cohesiveness is  $\geq$  then the system has to make no processing and just response answer to the user otherwise system has to make further processing. Suppose there is no such question in the database then the weight of the keywords is found in the query that will show the weight of the query. Here the weights are calculate according to the formula in (1) and make it in tablar form in table 1, also weights for all the sentences is also calculated in table 2. Now all the sentences are stored in the table with their corresponding weights. Now fuzzy member ship function is used to categorised the sentences as “Most cohesiveness” MC , “Relevant” R and “Irrelevant” IR sentences and collect only the Most cohesive sentences. All the Most Cohesive sentences are converted to a unique form “Case Frame” structure as method in [21] or it can also be sent to the user to select the useful answer fig 5a.

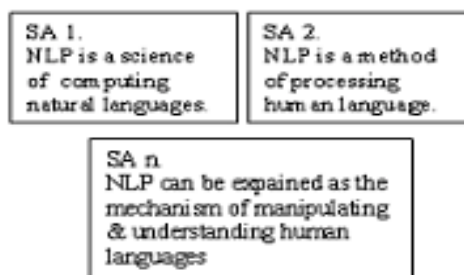


Figure 5 a Sub-Answers

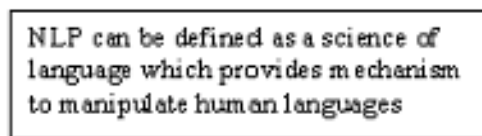


Figure 5. Super-Answer

Now, The next step is to make system intelligent. Intelligency is implemented in the system in three ways. First, The position where the Most Cohesive sentences are found in the documents, are marked with some special symbols for each of the query and a keyword index is generated; It also generate an Index table of keywords. These tables will be used futher if the next query is likely same as earlier or it has the keywords stored in index table 3. Second, Learn the query, cohesiveness and the positon of the corresponding answers which will help in predicting the answers of the queries. Finally, System can make predictions once it is learnt the question-answers. Which provides accurate answers with fewer efficiency fig 5.

**6. Discussion, Limitations & Further Possibilities of Research**

The task of answering FAQs is difficult to define and even more difficult to automate. Here a system described for extracting context based sentences from a set of documents in order to response for a given query from the user. This

allows users to have quick overviews of text documents relating to their queries. Unlike previous work on summarization, we summarize from a big summarize the text documents with following advantages:

(I) **Efficiency:** If the system is trained then there is no need to process the documents or fewer processing is needed.

(II) **Conciseness:** The Case frame is used as a technique to delete redundant sentences, which is an accurate way of removing redundancy. We believe that this system can serve, as an aid to the students, educationalists and the people doing research, as researchers has to read lots of papers and books.

(III) **Accuracy:** As every sentence in all the documents are participated in competition to be a most cohesive sentence with answer. It helps to produce all the contextual text, which is related to the answer of the question. The main idea is to score each sentence according to the weights of the keywords present in it. The most cohesive sentences are used for generating super answer.

Although the system succeeds in various scales of measurements like its efficiency, accuracy and conciseness up to some extent, yet it has some limitations, which can be made as a topic for the future research:

I. The system requires preprocessing of the queries in behind which the theme is to make it more cohesive or similar to the answer, which may not be success or may generate less accurate answer oriented sentence. In this step, it needs an accurate algorithm.

II. For calculating weights of the sentences, the important terms in queries are used as a major tool for calculating weights, which may not lead the exact meaning of the query.

III. Learning is used for a limited number of queries and for a huge collection of queries the learning algorithm may be complex or not working properly.

Finally, the proposed system is supposed to be a successful tool for answering FAQ if all the steps and modules use efficient and accurate algorithms.

## 7. Conclusion

Answering queries for text documents is an active field of research in both the IR and NLP communities. Among the large collection of text data, extracting the important and context based text from multidocument was a big NLP problem which is tried to solve by this attempt. A function  $\sigma(q, D)$  is defined to compute the answer  $\Lambda$  for

query  $q$  from the given set of documents  $D$ . The proposed system is supposed to doing following things: Give equal weights to all the subject-oriented documents as one part of answer may be in a document and other part may in other document rather than only getting one most relevant document, And then calculate the weights. Here this approach is able to generate more accurate answers. The Second thing is to develop an artificial memory which is useful to get answers quickly. The intelligent module also play an important role to learns the queries and the addresses of the answers which will be useful for making prediction of the answers without making the document processed. If it could not predict the exact answer even then the system will process fewer as it has a memory that which thing is stored where and it will leave that part of the text which are either less cohesive with the query or the keywords are not present there. It means the system is more efficient than that of the earlier approaches.

## References:

1. Query - based Multi document summarization for information retrieval, by T. Sakurai et al. NTCIR-4, Tokyo, June 2004.
2. Robust Generic and Query-based Summarization by Horacio Saggion et al University of Sheffield England –UK.
3. Headline Generation using a Training Corpus by Rong Jin et al. Presented at ICITPCL, Mexico City, Mexico, February 18 to 24, 2001
4. Automatic Text Summarization using a Machine learning approach by Joel Larocca et al. University of Parana (PUCPR).
5. Template-Filtered Headline Summarization by Liang Zhou et al. USC ISI 4676 Admiralty Way in 2004.
6. Query Based Event Extraction along a Timeline by Hai Leong Chieu et al. DSO National Laboratories, Singapore. 2004.
7. Query-Relevant Summarization using FAQs by Adam Berger et al., Carnegie Mellon University Pittsburgh
8. Hedge Trimmer: A Parse-and-Trim Approach to Headline Generation on 2003, by Bonnie Dorr, et al. University of Maryland.
9. GATE (a User Guide): Developing Language Processing Components with GATE by Hamish Cunningham1, et al University of Sheffield 2001-2002.
10. Automated Text summarization Tutorial — COLING/ACL'98 by Eduard Hovy et al. ISI University of Southern California.
11. Introduction to the Special Issue on Summarization Dragomir by R. Radev Eduard Hovy University of Michigan.

12. Statistical Machine Learning Approaches to Cross Language Text Summarization by Minh Le Nguyen SIS *Japan Advanced Institute of Science and Technology* September 2004, Ph.D. Thesis.
13. K.U.Leuven summarization system at DUC 2004 by Roxana Angheluta, et al., Belgium Interdisciplinary Center for Law & IT.
14. The Automatic Creation of Literature Abstracts by H. P. Luhn IBM JOURNAL APRIL 1958.
15. Automatic Text Summarization Evaluation A Survey of Methods and Tools by Martin Hassel NADA-KTH Royal Institute of Technology, Sweden.
16. Statistical Approaches Toward Automatic Title Generation by Rong Jin LTISCS, Carnegie Mellon University.
17. Automatic Evaluation of Summaries Using N-gram Co-Occurrence Statistics by Chin-Yew Lin et al. ISI University of Southern California.
18. Hybrid Text Summarization Method based on the TF Method and the LEAD Method by Kai ISHIKAWA et al., Miyazaki Miyamae-ku Kawasaki-shi Kanagawa, and Japan.
19. Integration of Hand-Crafted and Statistical Resources in Measuring Word Similarity by Atsushi FUJII et al. DCS Tokyo Institute of Technology.
20. QUERY-BASED DISCOVERING OF POPULAR CHANGES IN WWW. By Adam Jatowt et al. University of Tokyo, Tokyo, Japan.
21. "A Constraint-based Case Frame Lexicon" by Kemal Oflazer et al. DCE & Isc, Bilkent University, Ankara 06533, Turkey.
22. An Algorithmic Approach for Fuzzy Inference by C. J. Kim, Member IEEE.