A Novel Approach towards Multi-Document Text Summarization

Prof. Sarika N Zaware (HOD), Asmit Gautam, Sumedha Nashte, Puneet Khanuja

Dept of Computer Engineering, AISSMS's Institute Of Information Technology, University of Pune, India

Abstract – With an increase in the size of documents, the task of searching the important and relevant information has become very tedious. Hence, a summarizer would prove vital towards reducing human efforts. Text summarization is an important activity in the analysis of high volume text documents and is currently a major research topic in Natural Language Processing. The summarizer proposed generates a summary based on tokenization, stop word removal, Term Frequency-Inverse Document Frequency (TF-IDF), calculation of similarity measures, separation of important sentences and generation of the final summary using custom input.

Keywords: Text Summarization, Term Frequency-Inverse Document Frequency(TF-IDF),Natural Language Processing

I. INTRODUCTION

Automatic text summarization is the process of reducing a text document with a computer program in order to create a summary that retains the most important points of the original document. As the problem of information overload has grown and the quantity of data has increased , so has the interest in automatic summarization.

Two fundamental techniques are identified to automatically summarize texts, i.e. abstractive and extractive summarization [8]. In contrast to abstraction, which requires using complex techniques from Natural Language Processing (NLP) including grammars and lexicons for parsing and generation, extraction can be easily viewed as the process of selecting important excerpts from the original document and concatenating them into a shorter form.

With an exponential increase in the size of the internet documents, searching for valuable information has become a tedious task. Introduction of automation in such a field reduces the human effort and time taken to find the relevance of a document to the person.

II. SYSTEM MODEL

The Multi-Document text summarization application is a client-server architecture based model. The client side sends a request to the server. The server processes the request and sends the result or summary created, to the client. The result/summary is viewed on the client's machine. The client-server based architecture is implemented to facilitate the use of online application without having the need to install it on the client machine.

The client first logs in with a username and password. This username and password is authenticated. A valid user gets access to the online application. The user then selects a single or multiple documents according to his preference. The client submits these documents to the server along with the custom request. A custom request will be a compression ratio. A compression ratio is a measure of how big the user wants the abstract to be.

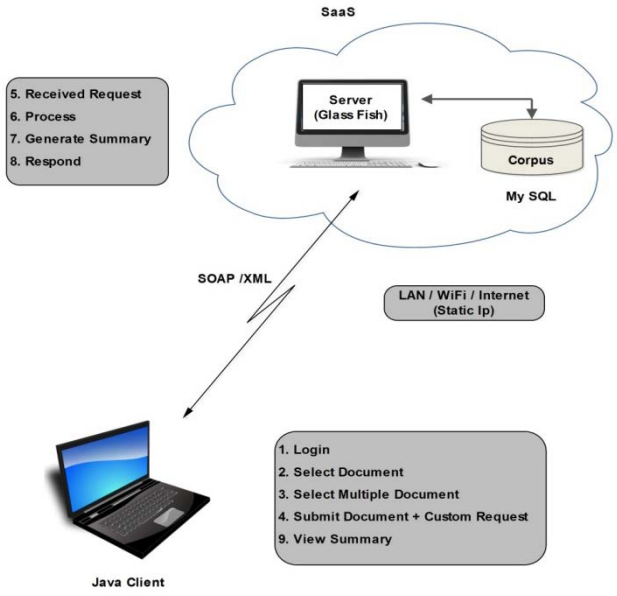

Figure 1: The proposed system model for summarizer

This compression ratio, though, must not be greater than the length of the documents, i.e. the value should be between 0-1 (0-100%).

The server receives the request from the client. The documents are then processed for summarization along with the custom request. The abstract is then created. This abstract is sent to the client. Final summary is viewed on the client machine.

III. PREVIOUS WORK

Text summarization is an important application which can be used on a day to day basis. A lot of work has been carried out in the area to reduce human effort.

Text summarization is the process of generating a summary by extracting the representative sentences from it. Anusha et al [1]l provided a technique for summarization of domain-specific text from a single web document. The summarizer uses TF/IDF, Compression Ratio to summarize the text but the generic design of an automatic summarizer is still very challenging. Work has been done based on extracting keywords and important sentences from a document. These can be converted to a starmap. Such a system can be used to eliminate duplication or for planning a lesson. The system can also facilitate plagiarism detection or useful in evaluating examination answers.

Summary of the document can also be based on sentence scoring techniques. Use of a combination of techniques like word based scoring, sentence based scoring, and graph based scoring yields a better result than individually using any of the above would produce. A combination of the algorithms may produce better results but can be much slower and at times not as efficient as any of the individual.[3]

Core concepts can be extracted from a text corpus in two phases. In the first phase, keywords are extracted by tokenizing, stop word removal and generating N-grams. The second phase includes co-word occurrence extraction and calculation of centrality measure. Lack of good stemming engine reduced the quality of tokens.[4]

Another approach to summarization uses clustering. Since documents often cover a number of topic themes with each theme represented by a cluster of highly related sentences, clustering has been explored in order to provide more informative summaries. The rank of terms on this topic theme should be very distinct from the rank of terms in other topic themes. A newly emerged framework uses sentence clustering results to improve or refine the sentence ranking results .Distribution of ranks of sentences in each cluster should be quite different from one another, which may serve as features of clusters and new clustering measures of sentences can be calculated accordingly.[5]

Another approach can be by giving importance of a sentence as input and text is evaluated with the help of Simplified Lesk Algorithm. According to the given percentage for summary, number of sentences is selected from the ordered number. System gives best results upto 50% summarization and satisfactory results upto 25% summarization.[7]

IV. PROPOSED METHODOLOGY

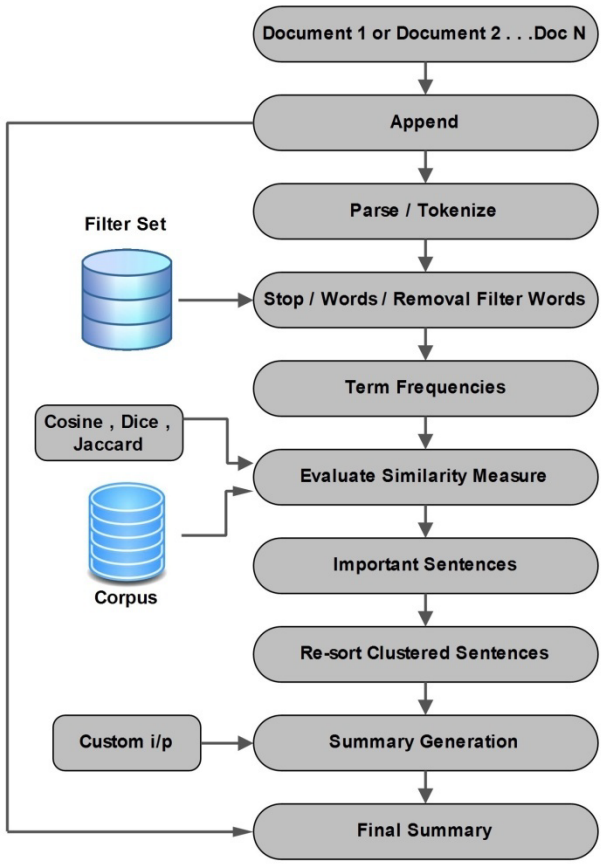

Figure 2: The Flow Diagram for the summarizer

The proposed system can be described in the Ten distinct phases of the working of the summarizer.

- LOGIN PHASE
  In this phase the user logs in with a username and a password. The user logs on validation of username and password.
- DOCUMENT SELECTION PHASE

The user selects single or multiple documents according to requirement.

The selected document/s are sent to the System

- APPEND PHASE

  As we are using the multi-document text summarizer we have to append the different documents after uploading them on the server machine. Combining the documents into one reduces the time as the summary is generated only once.

- PARSE/TOKENIZE PHASE

  The selected and appended documents are then sent to the parser.

  Tokenized keywords are stored in the database.

  Sentences are numbered and then the punctuations are removed.

  The output we get is a tokenized document.

- STOP WORD REMOVAL

  The token database is then compared with the stop word database for stop word removal.

  Stop words like “and , of , the” etc. are removed

- TF-IDF CALCULATION

  The revised tokens are then calculated for TF-IDF.

  TF: (Term Frequency)

  Wi= tfi where tfi is the number of times the term occurred in the document

  TF*IDF: (Inverse Document Frequency)

  Wi =tfi*idfi=tfi*log(N/dfi)) where dfi is the number of documents contains term i, and N the total number of documents in the collection.[9]

  TF-IDF for the tokens is calculated and stored in the TF-IDF database.

- SIMILARITY MEASURE EVALUATION

  We calculate the similarity between the documents using the following:

1) Jaccard :

   The Jaccard coefficient measures similarity between finite sample sets, and is defined as the size of the intersection divided by the size of the union of the sample sets:

   $$J(A,B)=|A\cap B|/|AUB|$$

   (If A and B are both empty, we define $J(A,B) = 1$.[10]

2) Cosine:

   The cosine of two vectors can be derived by using the Euclidean dot product formula:

   $$a.b=||a||\ ||b||\ \cos\Theta$$

   The resulting similarity ranges from −1 meaning exactly opposite, to 1 meaning exactly the same, with 0 usually indicating independence, and in-between values indicating intermediate similarity or dissimilarity.[11]

3) Dice:

   For sets X and Y of keywords used in information retrieval, the coefficient may be defined as twice the shared information (intersection) over the sum of cardinalities

   When taken as a string similarity measure, the coefficient may be calculated for two strings, x and y using bigrams as follows:

   $S=2n_t/(n_x+n_y)$

   where n_t is the number of character bigrams found in both strings, n_x is the number of bigrams in string x and n_y is the number of bigrams in string y.

   A combination of any two or all three can be used for enhancing the efficiency of the system.[12]

- IMPORTANT SENTENCE SELECTION

  Unique sentences are selected from the document/s according to the TF-IDF calculated

  The summary is thus generated from the unique sentences selected and displayed to the user.

- RE-SORT THE CLUSTERED SENTENCES

  After selection of the important sentences, the sentences must be sorted in an order to provide a meaningful abstract

- SUMMARY GENERATION

  The final summary is generated using the above discrete sentences and the custom input taken from the user. This summary can be then sent to the user.

V. CONCLUSION

Researchers generally depend on a well built summary for the research paper in the literature. The summary is very important from the point of view of the analysis of the paper and the hard work of the researchers behind them.

Since there is a wide amount of textual information available on Web which cannot be analyzed by humans, a service oriented approach can be useful for retrieving important information from text. In this project we used a text summarization application for summarizing the given text by extracting relevant sentences for creating summary. The proposed method is basically an extraction based approach which can be applied to a single document or multiple documents.

VI. FUTURE SCOPES

In future it can be used as an independent application for text summarization of heavy data on different network topologies. This application can be deployed on single server machine and

other client machines can use it like SaaS from the main frame. Cloud can be used to make the application work online.

As the contents of all research and development is getting transformed into electronic media, our system can be used to maintain the database of the contents(Sorting as per their fields).It can also be proposed as the main structure for creating an E-Library. It can be an added feature in our normal text editor and viewers for summarizing the text can just use the feature for small text abstraction as and when they are reading it reading it. It can be added in O.S. for providing high quality of efficiency for professionals dealing with many numbers of files.

REFERENCES

[1] Anusha Bagalkotkar , Ashesh Khandelwal,Shivam Pandey,Sowmya Kamath S, ”A Novel Technique for Efficient Text Document Summarization as a Service.”,IEEE 2013.

[2] M Kalaisevan, A Kathiaravan, “A Pioneering Tool For Text Summarization using StarMap”,IEEE 2013.

[3] Rafael Ferreira ,Frederio Fraetas, Luciano Favaro, ”A Context Based Text Summarization System”, 2014, 11th IAPR International Workshop on Document Analysis System.

[4] Ammar lalalimanesh, ” Knowledge Discovery in Scientific Databases Using Text Mining and Social Network Analysis”, 2012 IEEE Conference on Control, Systems and Industrial Informatics (ICCSII) Bandung, Indonesia, September 23-26,2012.

[5] Xiaoyan Cai and Wenjie Li, ” Ranking Through Clustering: An Integrated Approach to Multi-Document Summarization”, IEEE TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING, VOL. 21, NO. 7, JULY 2013.

[6] Sarah Rastkar, Gail C. Murphy, Member, IEEE, and Gabriel Murray, “Automatic Summarization of Bug Reports”, IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 40, NO. 4, APRIL 2014.

[7] Alok Ranjan Pal, Diganta Saha, “An Approach To Automatic Text Summarization Using Wordnet” IEEE 2014 .

[8] Nenkova, Ani, and Kathleen Mckcown. Automatic summarization. Now Publishers Inc, 2011.

[9] Juan Ramos, Using TF-IDF to Determine Word Relevance in Document Queries.

[10] Tan, Pang-Ning; Steinbach, Michael; Kumar, Vipin (2005), Introduction to Data Mining, ISBN 0-321-32136-7.

[11] P.-N. Tan, M. Steinbach & V. Kumar, "Introduction to Data Mining", , Addison-Wesley (2005), ISBN 0-321-32136-7, chapter 8; page 500.

[12] Kondrak, Grzegorz; Marcu, Daniel; and Knight, Kevin (2003). "Cognates Can Improve Statistical Translation Models" (http://aclweb.org/anthology/N/N03/N03-2016.pdf). Proceedings of HLT-NAACL 2003: Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics. pp. 46–48.