Available online at http://www.journalcra.com

International Journal of Current Research
Vol. 8, Issue, 06, pp.32430-32434, June, 2016

INTERNATIONAL JOURNAL OF CURRENT RESEARCH

ISSN: 0975-833X

REVIEW ARTICLE

MULTI-DOCUMENT SUMMARIZATION USING SIMILARITY MEASURES

Sanket M. Sathe, *Pranay N. Lonkar, Nayan A. Shendre, Sonali M. Shingade and Prof. Nihar Ranjan

Department of Computer Engineering, SITS, Pune, India

ARTICLE INFO

Article History:
Received 23rd March, 2016
Received in revised form
10th April, 2016
Accepted 05th May, 2016
Published online 15th June, 2016

Key words:

Text Summarization,
Tokenization,
Stop Words,
Stemming,
Term Frequency,
Inverse Document Frequency.

ABSTRACT

As the data is growing everyday and comparatively less amount of useful information is made available on the internet, it becomes necessary to introduce a mechanism that can easily search out relevant information from that bulk of data. This is what has contributed toward the rise of the concept of multi-document summarization where the whole document is condensed to a smaller version retaining its original meaning. We are going to work on automated creation of summaries of one or more text documents using similarity measure algorithm – cosine similarity.

Citation: Sanket M. Sathe, Pranay N. Lonkar, Nayan A. Shendre, Sonali M. Shingade and Prof. Nihar Ranjan, 2016. "Multi-document summarization using similarity measures", *International Journal of Current Research*, 8, (06), 32430-32434.

INTRODUCTION

Multi-document summarization is aimed at extraction of information from multiple texts written about the same topic. The resulting summary report allows its users to quickly familiarize themselves with information contained in a large group of documents. Multi-document summarization generates information reports that are both concise and comprehensive. While the goal of a concise summary is to simplify information search and cut the time by pointing to the most relevant source documents, complete multi-document summary should itself contain the required information, hence limiting the need for accessing original files to cases when modification is required. Automatic summaries present information extracted from multiple sources algorithmically, without any subjective human intervention, thus making it totally impartial. The major challenge of multi-document summarization is due to the multiple resources from which information is extracted. Multiple documents consist of the risk of higher redundant information than would typically be found in a single document.

Corresponding author: Pranay N. Lonkar,
Department of Computer Engineering, SITS, Pune, India.

Also, the ordering of extracted information from a set of documents into a coherent text in order to create a coherent summary is a non-trivial task (Bollegala, 2007). Summarization can be either extractive or abstractive. Extractive summarization involves assigning measure of most importance to some units (e.g. sentences, paragraphs) of the documents and extracting those with highest marks to comprise in the summary. Abstractive summarization usually needs information fusion, sentence compression and reformulation. Abstractive summarization is a difficult problem because it requires deeper analysis of source documents and concept-to-text generation.

At present, most of the researches and commercial systems in automatic text summarization are extractive summarization. Regarding generality of summaries, two types can be distinguished: generic and query-driven summaries. The generic summaries tries to represent all relevant topics of a source text, while the query-driven summaries focus on the user's desired query keywords or topics. Most of the existing successful summarization systems are used in domain of news articles where each document is assumed to have a 'mono-concept'. It is supposed in these systems that a document has information about a single event, accident, or news.

In such systems, one of the key tasks is to group many documents either on time bases or on topics extracted from user-input query. For example MEAD (2) selects centroid sentence of each cluster, and searches for similar or strongly related sentences to centroids. CLASSY (Schlesinger *et al*., 2008) ranks sentences with their inclusion of user query terms and their associated signature words. The organization of the paper is as follows. Section II presents literature survey and section III discusses text summarization process. In section IV, we discuss the similarity measures followed by experimental result in Section V and conclusion and future work in Section VI. Section VII is about acknowledgement and lastly, section VIII contains all references.

LITERATURE SURVEY

Various research is done in the past regarding text document summarization. In the following we inspect some review papers related to text summarization. The paper (Kumar Nagwani *et al*., 2011) describes that document retrieval is not sufficient and we need a second level of abstraction to decrease this huge amount of data: the ability of summarization. Yan *et al*. while describing about SRRank, suggested that multi-document summaries can be used to quickly browse document collections, and it has been shown that multi-document summaries can be useful in information retrieval systems. Jade *et al*. proposed the situations where multi- document summarization would be useful: (1) the user is faced with a collection of unrelated documents and desires to measure the information landscape contained in the collection, or (2) there is a collection of related documents, removed from a larger more diverse collection as the result of a query, or a topically-cohesive cluster. Rafeeq Al-Hashemi (2010) described about the categories of summarization task: extractive summarization and abstractive summarization. Extractive methods work by choosing a sub-set of existing words, phrases, or sentences in the original text to form the summary supposing that these sentences express the meaning of the whole text. Abstraction based methods create a compact version of text expressing the summarized meaning of the original text.

In order to generate one summarization document from multiple articles (Yohei), one approach is to compute each sentence's importance weight within each document. The simplest strategy is to remove important sentences equally from every related document according to the rates of summarization and organize them chronologically. By weighing sentence importance with tf/idf value of a contained lexical set or words in the heading, we can extract sentences specific to each document. Another method is considering each sentence across the document set. In order to implement this strategy the importance value of each sentence is adjusted from 0 to 1 by dividing the sum of tf/idf values contained in each sentence and comparing sentences' importance values across all documents.

TEXT SUMMARIZATION PROCESS

The proposed method can be described in seven steps as shown in Figure I.

- Selection of text documents: In the first step text documents which are required to be summarized are given by the user.
- Append and Tokenization: Text documents are appended and then the file content is tokenized into individual word.
- Removal of stop words: Many of the most frequently used words in English are worthless in IR and text mining – these words are called stop words. For example: a, an, the, is, are, and, to, of, etc... Actually, there is no standard dictionary for stop words. Stop words accounts 20-30% of total word counts. To improve the efficiency of the summarizer, it is necessary to remove them.
- Stemming: We find out the root/stem of a word. Various suffixes are removed; number of words is reduced by having exactly matching stems. Stem is not (necessarily) morphological root. For example: Abate, abated, abatement, abatements, abates might all stem to "*abat*".
- Generation of list of frequent words: After eliminating stop words the term-frequent data and inverse document frequency is calculated from text documents and frequent terms are selected which are used to generate text document summary.
- Sentence Generation: Similarity measure is evaluated using cosine algorithm and important sentences are generated. Unique sentences are clustered and re-sort. Finally, the summary is generated.
- Update details in database: When the summary is generated then its details is stored in the database and is available to the user for information analysis.
- Setup Web Service: A web service to provide summary of given text documents, will be set up. The Web Service client will send request message consisting of document then the server sends the summary as the response message.

Pre-processing (Rajesh Prasad and Uday Kulkarni, 2010) is the first component of the system with three different phases: sentence segmentation, removing stop words and, stemming. After applying pre-processing techniques, individual sentences and their unique ID are obtained from the text document:

- Segmentation process is achieved by finding out the delimiter (“.”full stop) so that, the sentences in the document are separated.
- Stop words (Pant *et al*., 2004) are detached from the document during the feature extraction step since they are considered as unimportant and contain noise. Stop words are predefined and are stored in an array and the array is utilized for comparison with the words in the provided document.
- Word stemming (Lovins, 1968) converts every word into its root form. Word stemming is practically removing the prefix and suffix of the specified word which in turn becomes applicable for comparison with other words.

SIMILARITY MEASURES

A similarity measure or similarity function is a real-valued function that quantifies the similarity between two objects. They take on large values for similar objects and zero for very dissimilar objects.

Table 1. Compression Ratio Evaluation

No. of Documents	No. of lines in the document(s) (X)	No. of lines in the summary (Y)	Compression Ratio (Y/X)
1	8	7	0.87
2	15	14	0.93
3	25	22	0.88
4	34	31	0.91

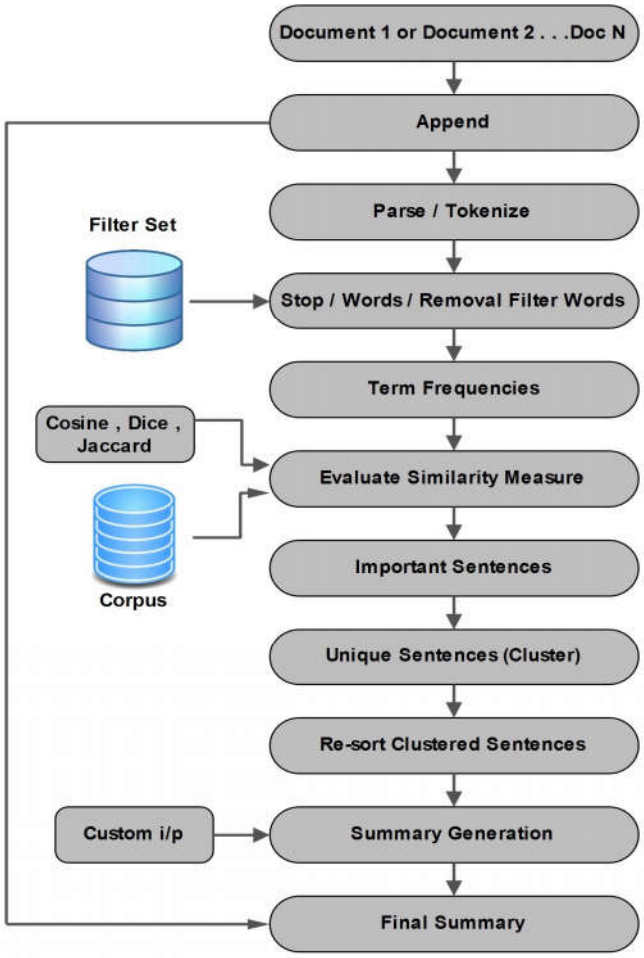

Figure 1. The Proposed System

Phases of the summarizer:

Input:

1. Multiple text documents for which summary is to be generated.
2. Value of *N* for generation of *N* lines of summary.

Operations:

1. Data Pre-processing Phase
   - Retrieve text documents
   - Eliminate stop words
   - Apply stemming
2. For the entire text content
   - Get the TF and IDF
   - Apply Cosine Similarity algorithm
   - Re-sort the important sentences
   - Add sentences to summary-sentence-list

Output:

1. Summary for text documents.

Cosine similarity is a commonly used similarity measure for real-valued vectors, used in (among other fields) information retrieval to score the similarity of documents in the vector space model. Cosine similarity is a measure of similarity between two vectors of an inner product space that measures the cosine of the angle between them. The cosine of 0° is 1, and it is less than 1 for any other angle. It is thus a judgment of orientation and not magnitude: two vectors with the same orientation have a cosine similarity of 1, two vectors at 90° have a similarity of 0, and two vectors diametrically opposed have a similarity of -1, independent of their magnitude. Cosine similarity is particularly used in positive space, where the outcome is neatly bounded in (0, 1).

Formula for cosine similarity coefficient (X,Y): $\frac{|X \cap Y|}{|X|^{1/2}.|Y|^{1/2}}$

X represents any of the documents given as input and Y represents the corresponding query.

EXPERIMENTAL RESULTS

Experimental Setup

The algorithm is the functionality provided by a web service which can be invoked in submission of documents. The user-uploaded documents are sent to the server, where the service is invoked. On the server all the processing steps are applied. Then, cosine similarity algorithm is applied for generating summary. Finally, the server sends this summary to the client.

Evaluation Parameters

A summary must be shorter than the original input text and it must contain the important information of the original, and not other, totally new, information. There are two measures to capture the extent to which a summary conforms to these requirements with regard to text documents:

- Compression Ratio
- Retention Ratio

We choose to measure the length. So a good summary is one in which compression ratio is small (tending to zero). Compression ratio is obtained using the following formula:

$$CR = \frac{(\text{length of the summary})}{(\text{length of all input text documents})}$$

The following are the observations made from the text document summarizer obtained using cosine similarity algorithm:

This data is being generated using 4 test cases where we considered the value of Y depending upon the at most summary the summarizer can produce. The compression ratio keeps changing as our text summarizer accepts number of lines from the user for the summary.

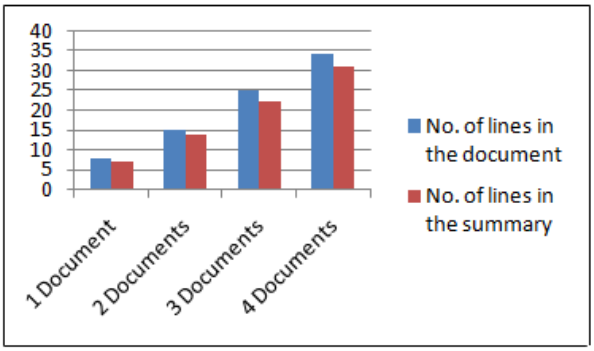

Figure 2. Graphical Representation of Compression Ratio

So the compression ratio will keep on varying but, it won't be greater than 1. The summarizer can accept documents containing multimedia content but, it can't process tabular data, images, audio or video files. The resultant summary will be generated only from the textual data from the documents. So, the summary in such cases can't be efficient to the user.

Conclusion and Future Work

Since there is vast amount of textual information available on Web which can't be analyzed by humans, a service oriented approach can be useful for retrieving important information from text documents. In this paper, we developed a text summarizer that is capable of producing a relevantly abstract summary from multiple text documents of same domain as per the number of lines requested by the user. The proposed method is basically an extraction based approach.

Our system builds upon previous work in single-document summarization - taking into account some of the major issues arising in multi-document summarization: (i) the need to carefully eliminate redundant information from multiple documents, and achieve high compression ratios, (ii) information about document and passage similarities, and weighting different passages accordingly, and (iii) the importance of temporal information. Future work includes (i) integration of multi-document summarization with document clustering to provide summaries for clusters produced by topic detection and tracking, (ii) generation of coherent temporally based event summaries and, (iii) construction of interactive interfaces so that users can effectively use multi-document summarization to browse and explore large document sets.

Acknowledgement

We would like to thank our project guide Prof. Nihar Ranjan and all the staff members of our department for giving us valuable time and contributing their experience towards this project.

REFERENCES

Bollegala, D. T. 2007. Improving coherence in multi document summarization through proper ordering of sentences, Master of Science, Information & Communication Engineering, *Graduate School of Information Science and Technology*, Tokyo, Japan.

Canhasi E., Kononenko I., Weighted archetypal analysis of the multi-element graph for query-focused multi-documentsummarization // Expert Systems with Applications, 2014, vol.41, no.2, pp.535–543.

Jade Goldstein, Vibhu Mittal, Jaime Carbonell, and Mark Kantrowitz. "Multi-Document Summarization By Sentence Extraction".

Kumar Nagwani, Naresh, and Shirish Verma. 2011. "A Frequent Term and Semantic Similarity based Single Document Text Summarization Algorithm". *International Journal of Computer Applications.*

Lloret E., Palomar M. COMPENDIUM: a text summarization tool for generating summaries of multiple purposes, domains, and genres // *Natural Language Engineering*, 2013, vol.19, no.2, pp.147–186.

Lovins, J.B., 1968: Development of a stemming algorithm. *Mech. Trans. Comput. Linguist*., 11: 22-31. DOI: 10.1234/12345678.

Luo W., Zhuang F., He Q., Shi Z. Exploiting relevance, coverage, and novelty for query-focused multi-document summarization // *Knowledge-Based Systems*, 2013, vol.46, pp.33–42.

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

Pant, G., P. Srinivasan and F. Menczer, 2004. Crawling the Web. In: Web Dynamics: Adapting to Change in Content, Size, Topology and Use, Levene, M. and A. Poulovassilis (Eds.). *Springer*, USA., pp: 153-178.

Radev, Dragomir R., Otterbacher, J., Qi, H. and Tam, D., 2003. Mead reducs: Michigan at duc 2003. In Proceedings of DUC 2003. Edmonton, AB, Canada.

Rafeeq Al-Hashemi. "Text Summarization Extraction System (TSES) Using Extracted Keywords". *International Arab Journal of e-Technology*, Vol. 1, No. 4, June 2010.

Rajesh Prasad, Uday Kulkarni, "Implementation and Evaluation of Evolutionary Connectionist Approaches to Automated Text Summarization", *Journal of Computer Science*, P: 1366 – 1376, 2010.

Su Yan, Xiaojun Wan. "SRRank: Leveraging Semantic Roles for Extractive Multi-Document Summarization". Audio, Speech, and Language Processing, IEEE/ACM Transactions on (Volume:22, Issue: 12).

Yohei SEKI. "Sentence Extraction by tf/idf and Position Weighting from Newspaper Articles". Proceedings of the Third NTCIR Workshop.

****