A Frequent Term and Semantic Similarity based Single Document Text Summarization Algorithm

Naresh Kumar Nagwani
Assistant Professor, Department of CS&E,
National Institute of Technology Raipur

Dr. Shrish Verma
Associate Professor, Department of E&TC,
National Institute of Technology Raipur

ABSTRACT
Text summarization is an important activity in the analysis of a high volume text documents. Text summarization has number of applications; recently number of applications uses text summarization for the betterment of the text analysis and knowledge representation. In this paper a frequent term based text summarization algorithm is designed and implemented in java. The designed algorithm works in three steps. In the first step the document which is required to be summarized is processed by eliminating the stop word and by applying the stemmers. In the second step term-frequent data is calculated from the document and frequent terms are selected, for these selected words the semantic equivalent terms are also generated. Finally in the third step all the sentences in the document, which are containing the frequent and semantic equivalent terms, are filtered for summarization. The designed algorithm is implemented using open source technologies like java, DISCO, Porters stemmer etc. and verified over the standard text mining corpus.

Keyword

Text Summarization, Frequent Words, Semantic Similar, Summarization Algorithm.

1. INTRODUCTION
Text database contains high volume of text data. Document retrieval retrieves number of documents still beyond the capacity of human analysis, e.g. at the time of writing the query "Summarization" in Google returned more than 9,090,000 results (as on 30th Jan 2011 from Google). Thus document retrieval is not sufficient and we need a second level of abstraction to reduce this huge amount of data: the ability of summarization. This work tries to address this issue and proposes an automatic text summarization (TS) technique. Roughly summarization is the process of reducing a large volume of information to a summary or abstract preserving only the most essential items.

1.1 Text Mining
The problem of Knowledge Discovery from Text (KDT) is to extract explicit and implicit concepts and semantic relations between concepts using Natural Language Processing (NLP) techniques. Its aim is to get insights into large quantities of text data. KDT, while deeply rooted in NLP, draws on methods from statistics, machine learning, reasoning, information extraction, knowledge management, and others for its discovery process. KDT plays an increasingly significant role in emerging applications, such as Text Understanding. Text mining is similar to data mining, except that data mining tools are designed to handle structured data from databases, but text mining can work with unstructured or semi-structured data sets such as emails, full-text documents and HTML files etc. As a result, text mining is a much better solution for companies. To date, however, most research and development efforts have centered on data mining efforts using structured data. Several text mining applications are search engine, text categorization, summarization, and topic detection.

1.2 Text Summarization
Text summarization or rather automatic text summarization corresponds to the process in which a computer creates a shorter version of the original text (or a collection of texts) still preserving most of the information present in the original text. This process can be seen as compression and it necessarily suffers from information loss. Thus a TS system must identify important parts and preserve them. What is important can depend upon the user needs or the purpose of the summary.

1.2.1 Application of Text Summarization
Some of the applications are listed here; the reader is referred to for a detailed discussion.

- Text Summarization can be used to save time.
- Text Summarization can speed up other information retrieval and text mining processes.
- Text Summarization can also be useful for text display on hand-held devices, such as PDA. For instance a summarized version of an email can be sent to a hand-held device instead of a full email.

1.2.2 Classification of Text Summarization Techniques
Text Summarization is condensing the source text into a shorter version preserving its information content and overall meaning. The text summarization techniques can be classified by using the way by which the summarization is going to be performed over the text data. Following are the two broad level classifications of text summarization techniques.

1.2.2.1 Extractive and Abstractive Text Summarization
Text Summarization methods can be classified into extractive and abstractive summarization. An extractive summarization method consists of selecting important sentences, paragraphs etc. from the original document and concatenating them into shorter form. The importance of sentences is decided based on statistical and linguistic features of sentences. An abstractive

summarization method consists of understanding the original text and re-telling it in fewer words. It uses linguistic methods to examine and interpret the text and then to find the new concepts and expressions to best describe it by generating a new shorter text that conveys the most important information from the original text document. [17]

1.2.2.2 Single Document and Multi Document Text Summarization

Text summarization techniques can also be classified on the basis of volume of text documents available in the text database. If summarization is performed for a single text document then it is called as the single document text summarization. If the summary is to be created for multiple text documents then it is called as the multi document text summarization technique.

1.3 Semantic Similarity

Semantic similarity is a concept whereby a set of documents or terms within term lists are assigned a metric based on the likeness of their meaning / semantic content. Various semantic similarity techniques are available which can be used for measuring the semantic similarity between text documents.

Some of the Semantic similarity methods are studied and summarized in the work done by Nagwani and Singh [11], here is the summary of a few methods. Semantic similarity methods are classified into four main categories.

- Edge Counting Methods - Measure the similarity between two terms (concepts) as a function of the length of the path linking the terms and on the position of the terms in the taxonomy.
- Information Content Methods - Measure the difference in information content of the two terms as a function of their probability of occurrence in a corpus.
- Feature based Methods - Measure the similarity between two terms as a function of their properties (e.g., their definitions) or based on their relationships to other similar terms in the taxonomy
- Hybrid methods - Combine the above three mentioned methods for calculating the semantic similarity.

In this work semantic similarity of frequent terms are also used to preserve the meaning of original text document in the summarized document. This paper is organized in five sections. Section two is about the relevant work done in the field. Section three is consisting of proposed methodology and section four discusses about the implementation and experiment carried over the proposed technique. Section five is the conclusion of the work done.

2. RELATED AND PREVIOUS WORK DONE

This section is consisting of brief study of text summarization and related work done so far. A text summarization evaluation technique named AutoSummENG (AUTOmatic SUMMarization Evaluation using N-gram Graphs) is proposed by Giannakopoulos et al [3]; various methods for evaluation are also discussed for the proposed technique. A language- and domain-independent statistical-based method for single-document extractive summarization is proposed by Ledeneva et al [20], to produce a text summary by extracting some sentences from the given text. The main problem for generating an extractive automatic text summary is to detect the most relevant information in the source document. An extractive text summarization algorithm is proposed by Amulfo et al [13], which use n-grams and maximal frequent word sequences as features in a vector space model.

A machine learning ranking algorithm is proposed by Amini et al [9] for single document summarization. The use of machine learning techniques for summarization allows one to adapt summaries to the user needs and to the corpus characteristics. A set of features is first used to produce a vector of scores for each sentence in a given document and a classifier is trained in order to make a global combination of these scores. The ranking algorithm also combines the scores of different features but its criterion tends to reduce the relative disordering of sentences within a document. A Two-step Sentence Extraction summarization system is designed and introduced by Jung et al [19]. The proposed system combines statistical methods and reduces noise data through two steps efficiently. Various text summarization extractive techniques has been presented and studied by Gupta and Lehal [17]. An algorithm for language independent generic extractive summarization for single document is proposed by Patel et al [1]. The algorithm is based on structural and statistical parameters. The proposed algorithm was performed over a single-document summarization for English, Hindi, Gujarati and Urdu documents.

A text summarization technique is proposed by Hashemi [12]. The proposed model consists of four stages. The preprocess stages convert the unstructured text into structured. In first stage, the system removes the stop words, pars the text and assigning the POS (tag) for each word in the text and store the result in a table. The second stage is to extract the important key-phrases in the text by implementing a new algorithm through ranking the candidate words. The system uses the extracted keywords/key-phrases to select the important sentence. Each sentence ranked depending on many features such as the existence of the keywords/key-phrase in it, the relation between the sentence and the title by using a similarity measurement and other many features. The Third stage of the proposed system is to extract the sentences with the highest rank. The Forth stage is the filtering stage, where the relevant sentences are filtered.

3. PROPOSED METHODOLOGY AND ALGORITHM

The overall proposed algorithm is represented in fig. 1, where all the steps are depicted in sequential manners. The system is divided into three major parts, an input text document, a summarizer algorithm and a summarized text document as output. The summarizer algorithm is further divided into the three parts – the text pre-processing module, frequent terms generation module along with the semantically similar terms and sentence filtering module for summarization.

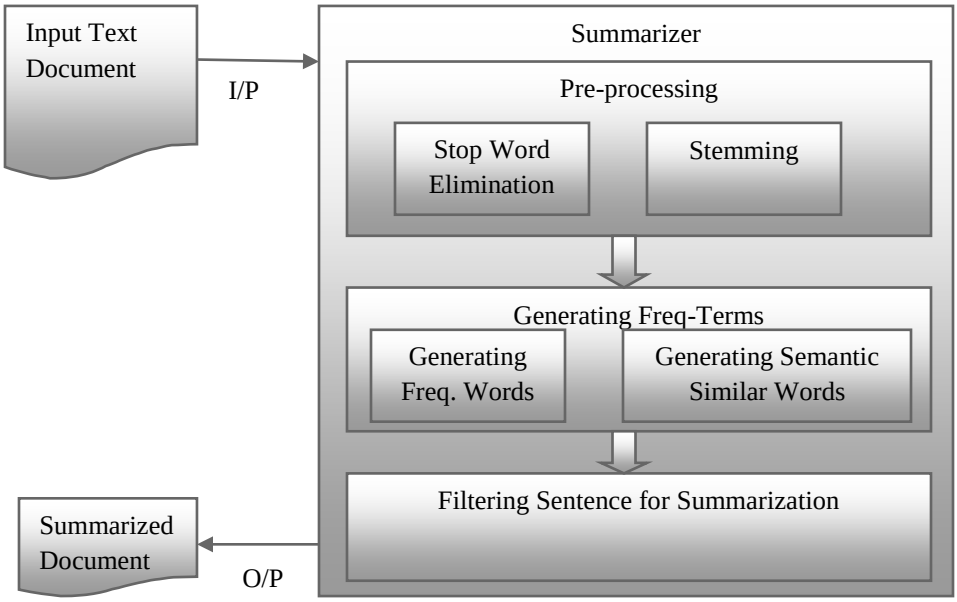

Fig 1: Overall Methodology of Frequent-Terms and Semantic Similarity Based Summarization.

The overall methodology of semantic similarity bases single document summarization can be expressed in terms of an algorithm. The algorithm takes two input parameters – the input text document and number of frequent terms. As the output it generates a summarized text document along with the two measures compression ratio and retention ratio, which are explained in the next section. The algorithm is consisting of six numbers of steps.

Algorithm: **Single Document Text Summarization Using Frequent Terms and Semantic Similarity**
Input - ***I1.*** Text Data for which Summary is required.
I2. N - for generating top N frequent Terms.
Output - ***O1.*** Summary for the Original Text Data.
O2. Compression Ratio.
O3. Retention Ratio.
Steps:
1. Data Preprocessing
1.a Retrieve data
1.b Eliminate Stop Word
2. Generate Term-Frequency List
2.a Get the N frequent Terms
3. For all N-Frequent Terms
3.a Get the semantic similar words for the terms, add it to the frequent-terms-list
4. Generate Sentences from the Original Data
5. If the sentence consists of term present in frequent-terms-list then add the sentence to summary-sentence-list.
6. Calculate Compression Ratio and Retention Ratio.

4. IMPLEMENTATION & EXPERIMENT

This section discusses about the implementation and experiments done for the proposed summarization technique. In section 4.1 data (text corpus) used for the technique is mentioned, section 4.2 discusses about the evaluation parameters of the summarization and section 4.3 discusses about the results observed and analysis of the result.

4.1 Data

Project TIPSTER SUMMAC provides a corpus of 183 documents from the Computation and Language collection has been marked up in xml and made available as a general resource to the information retrieval, extraction, and summarization communities. The documents are scientific papers which appeared in Association for Computational Linguistics (ACL) sponsored conferences. The markup is based on automatic conversion from latex to xml, and as a result is fairly minimal. The markup includes tags covering core information such as title, author, date, etc., as well as basic structure such as abstract, body, sections, lists, etc. Figures, tables, equations, cross-references and references were all replaced with placeholder tags. The corpus was prepared by The MITRE Corporation and the University of Edinburgh.

4.2 Result Evaluation Parameters

There are two properties of the summary that must be measured while evaluating summaries and summarization systems - the Compression Ratio, i.e. how much shorter the summary is than the original, and the Retention Ratio, i.e. how much of the central information is retained. Retention Ratio is also sometimes referred to as Omission Ratio. The compression ratio and retention ratio can be calculated using equation A and B.

Compression Ratio: CR = (length S) / (length T) (A)

Retention Ratio: *RR* = (info in S) / (info in T) (B)

4.3 Experiments and Result Evaluation

The implementation is done using three popular open source programming API's:

- DISCO API - The open source java based API (Application Programming Interface) called DISCO (extracting DIstributionally related words using CO-occurrences) [2] is used for measuring the semantic similarity of frequent terms. The British National Corpus (BNC) dictionary is used for this purpose in DISCO java API. The size of the dictionary is around 1.6 GB.
- Weka - Weka [18] provides the java API for data mining operations. It was designed by WaikatoUniversity. Weka is a collection of machine learning algorithms for data mining tasks.
- Java - Java [6] is used the programming language for the implementation work for the proposed algorithm. Java is a general-purpose, concurrent, class-based, object-oriented language that is specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere".

Table-1 shows the average compression ratio and average retention ratio for different number of frequent terms chosen. The relationship between the frequent terms and compression ration can be represented using the graph shown in fig. 2 and the relationship between the compression ratio and retention ratio is shown with the help of graph shown in fig. 3.

Table 1. Average Text Compression Ratio and Retention Ratio for Different Frequent Number of Terms.

Number of Terms	1	2	3	4	5	10	15	25
Compression Ratio	21	37	52	67	79	85	91	93.8
Retention Ratio	41	45.7	57.2	64.3	87.5	91	93.5	95

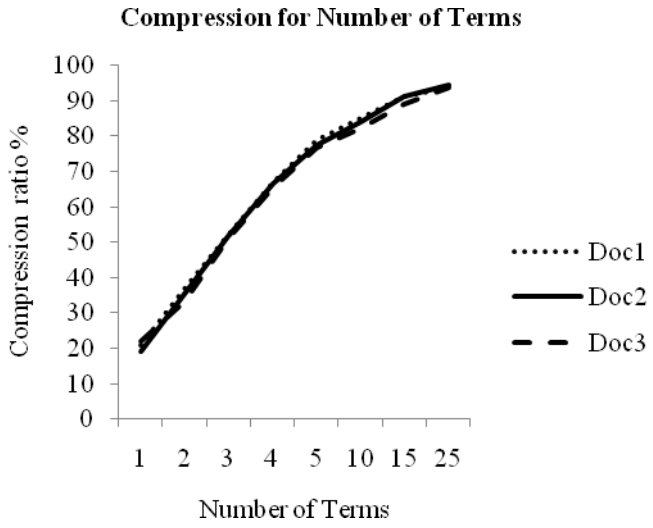

Fig 2: Compression Ratio for Different Number of Frequent Terms.

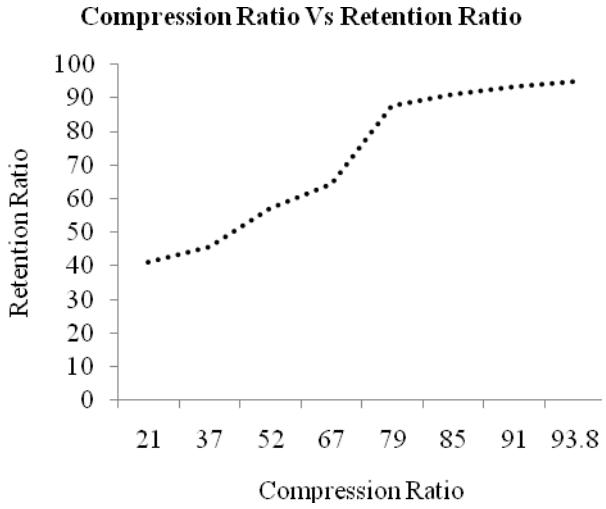

Fig 3: Compression Ratio Vs Retention Ratio for the summarization algorithm.

It is observed from fig. 2 that compression ratio increases as the number of frequent terms increases for sentence filtration. This is because the length of summary document increases if number of frequent terms for filtration increases. In the similar manners as compression ratio increase due to change in summary document length increase, it also increases the retention ratio of the summarized document. At initial levels change in compression ratio slightly increases the retention ratio but at the later phases retention ratio grows mostly in a linear form with the compression ratio.

5. CONCLUSION AND FUTURE SCOPE

In this paper a single document frequent terms based text summarization algorithm is introduced. Semantic similarity is also used in the algorithm. The proposed algorithm is implemented using open source technologies and is verified over the standard text mining corpus. The discovered results are interesting and meaning of the summarized document is also preserved. The future direction for the proposed work is to apply the similar concept in multi text document summarization.

6. REFERENCES

[1] Alkesh Patel, Tanveer Siddiqui, U. S. Tiwary, "A language independent approach to multilingual text summarization", Conference RIAO2007, Pittsburgh PA, U.S.A., (2007).

[2] DISCO (extracting DIstributionally related words using CO-occurrences) - http://www.linguatools.de/disco/disco_en.html the British National Corpus (BNC)

[3] George Giannakopoulos, Vangelis Karkaletsis, George Vouros, "Summarization Evaluation Under an N-Gram Graph Perspective. In View of Combined Evaluation Measures.", TAC2008, (2008).

[4] Goldstein J., Kantrowitz M., MittalV., Carbonell J.: Summarizing Text Documents: Sentence Selection and Evaluation Metrics. Proceedings of the 22th ACM SIGIR, 121–127, (1999).

[5] J. Kupiec, J. Pedersen, and F. Chen, "A trainable document summarizer", In Proceedings of the 18th ACMSIGIR Conference, pages 68-73, (1995).

[6] Java, The programming language - http://www.oracle.com/technetwork/java/index.html

[7] Jing H.: Summary generation through intelligent cutting and pasting of the input document. Technical Report Columbia University, (1998).

[8] Mani, I., Automatic Summarization, John Benjamins Publishing Co. (2001) 1-22.

[9] Massih R. Amini, Nicolas Usunier, and Patrick Gallinari, "Automatic Text Summarization Based onWord-Clusters and Ranking Algorithms", ECIR 2005, LNCS 3408, pp. 142–156, (2005).

[10] Mitra M., Singhal A., Buckley C.: Automatic Text Summarization by Paragraph Extraction. Proceedings of theACL'97/EACL'97Workshop on Intelligent Scalable Text Summarization, pp. 31–36 (1997).

[11] Naresh Kumar Nagwani, Pradeep Singh, "Weight similarity measurement model based, object oriented approach for bug databases mining to detect similar and duplicate bugs", ACM ICAC3 '09 Proceedings of the International Conference on Advances in Computing, Communication and Control, pp. 202-207, (2009).

[12] Rafeeq Al-Hashemi, "Text Summarization Extraction System (TSES) Using Extracted Keywords", International Arab Journal of e-Technology, Vol. 1, No. 4, June, pp. 164-168, (2010).

[13] René Arnulfo García-Hernández, Yulia Ledeneva, "Word Sequence Models for Single Text Summarization", 2009 Second International Conferences on Advances in Computer-Human Interactions, (2009).

[14] Sparck-Jones K.: Discourse modeling for automatic summarizing. Technical Report 29D, Computer Laboratory, university of Cambridge, (1993).

[15] Strzalkowski T., Wang J., Wise B.: A Robust practical text summarization system. Proceedings of the Fifteenth National Conferences on AI pp. 26–30 (1998).

[16] TIPSTER Text Summarization Evaluation Conference (SUMMAC) - http://www-nlpir.nist.gov/related_projects/tipster_summac/cmp_lg.html

[17] Vishal Gupta, Gurpreet Singh Lehal, "A Survey of Text Summarization Extractive Techniques", Journal of Emerging Technologies in Web Intelligence, VOL. 2, NO. 3, pp. 258-268, AUGUST (2010).

[18] Weka (a collection of machine learning algorithms for data mining tasks) - http://www.cs.waikato.ac.nz/~ml/weka.

[19] Wooncheol Jung, Youngjoong Ko, and Jungyun Seo, "Automatic Text Summarization Using Two-Step Sentence Extraction", AIRS 2004, LNCS 3411, pp. 71 – 81, (2005).

[20] Yulia Ledeneva, Alexander Gelbukh, and René Arnulfo García-Hernández, "Terms Derived from Frequent Sequences for Extractive Text Summarization", CICLing 2008, LNCS 4919, pp. 593–604, (2008).

[21] Zechner K.: Fast Generation of Abstracts from General Domain Text Corpora by Extracting Relevant Sentences. COLING, 986–989, (1996).