International Journal of Scientific Research and Engineering Development-– Volume 3 Issues 2 Mar- Apr 2020

Available at www.ijsred.com

RESEARCH ARTICLE OPEN ACCESS

Single Document Summarization Using Sentence Feature Profile

Mar Mar Soe*

*Faculty of Computer Science, University of Computer Studies (Loikaw)

Email: marmarsoe.m2@gmail.com

----------**********----------

Abstract:

Text summarization is the most challenging task in information retrieval tasks. It is an outcome of electronic document explosion and can be seen as the condensation of the document collection. The use of text summarization allows a user to get a sense of the content of full-text, or to know its information content without reading all sentences within the full-text. This system presents feature profile oriented sentence extraction strategy. Feature profile is generated by considering word weight, sentence position, sentence length, sentence centrality, proper nouns in the sentence and numerical data in the sentence. Sentence score is calculated and ranked in order of importance based on sentence score. Document summarization is the process of taking a textual document, extracting content from it. Englishnewspapers datasetis applied in this system for document summarization.

Keywords **—Text summarization, Feature profile, Sentence score**

----------**********----------

I. INTRODUCTION

Document summarization process reduces information overloading because only a summary needs to be read instead of reading the entire document. Data reduction helps user to find the required information quickly without having to waste time in reading the whole text. This system presents a feature profile based document summarizer based on feature of sentences and word frequency. A number of systems rank sentences based on sentence-level and word-level features. Then, it selects the top ranked sentence first, and then finds the word-level features. Document summarization is one feasible key to handle this information overload problem [1, 2]. An extract summary consists of sentences extracted from document(s) while an abstract summary may contain words and phrases which do not exist in the original document(s).

Document summarization is one feasible key to handle information overload problem.This can comprehensively help user to make out ideal documents within a short time by providing scraps of information. The aim of summarization is to present the most important information with a shorter version of the original text and helps the user to quickly understand large volumes of information.The need for getting maximum information by spending minimum time has led to more efforts being directed to the field of summarization. The objectives of the system are: to reduce information overloading, to generate feature profile for sentences, to extract most important sentences based on feature profile, and To get the summary information in news documents.

This paper is organized with five sections. The first section is introduction of the system. Section 2 explains theory of Text Summarization. Section 3 explains document summary process that includes six sub processes. These important processes are explained in detail of this section. Section 4 describes the system design and implementation of the summary generation.And the next section is conclusion, limitation and further extension of the system.

II. AUTOMATIC SUMMARIZATION

Automatic summarization is one such technique, where a computer summarizes a longer text into a shorter non-redundant form. The development of advanced summarization systems also for smaller languages may unfortunately prove too costly. Nevertheless, there will still be a need for summarization tools for these languages in order to curb the immense flow of digital information. Text Summarization has become a very popular Natural Language Processing (NLP) task in recent years [7]. Due to the vast amount of information, especially since the growth of the Internet, automatic summarization has been developed and improved in order to help users manage all the information available these days.

There are many other NLP tasks, such as Information Retrieval (IR), Information Extraction (IE), Question Answering (QA), Text Categorization (TC) or Textual Entailment (TE), which can interact together with the purpose of improving their performance and obtaining better results [3]. Summarization can be characterized as approaching the problem at the surface, entity, or discourse levels [8]. This approach attempts to build a representation of the text, modeling text entities and their relationships. Similarity when two words share common whose form is similar.

A. Document Preprocessing

Usually document sources are of unstructured format, transforming these unstructured documents to structured format requires some preprocessing steps. This could be a paragraph, a sentence, a phrase or even a clause, although the most common probably is extraction performed on sentence level. These steps are tokenization, stop word removal and stemming [6].

Before any further processing can be done, a text needs to be segmented into words and sentences. This process is called tokenization [10]. Stop words are the words which appear frequently in the document but provide less meaning in identifying the important content of the document[2]. There are over six hundreds and sixty stop words list. Some stop words are: a, an, the, according, again, against, back, become, can, did, don't, effect, else, instead, inward, just, often, once, somebody, try, useful, less, past, suggest, very, quite, we, that, while, might, normal, world, zero, if, etc…

These stemmers attempt to reduce a word to a common root form, often called a stem, so that the words in a document can be represented by one lexical string (term) rather than by the original word forms. The effect is not only that different variants of a term can be conflated to a single representative form, but it also reduces the size of the vocabulary the system need to store representations for.

III. DOCUMENT SUMMARY PROCESS

This system is decomposed into six sub processes:

1. Preprocessing
2. Term_Weight Calculation
3. Feature Profile Generation
4. Sentence Score Calculation based on Feature Profile
5. Sentence Ranking and Ordering
6. Summary Generation

B. Preprocessing

Preprocessing steps are **tokenization, stop words removal** and **stemming**. Often it is necessary to first split the text into words (tokens) in order to correctly identify these boundaries between clauses, phrases or sentences. Tokenization divides the character sequence into words, sentence splitting further divides sequences of words into sentences, and so on. Filtering of the text is done by removing the stop words. The last step is word stemming; word stemming is the process of removing affixes of each word and produces the root word known as the stem. This stemming step is performed by using enhanced Porter Stemming algorithm. Eg., 'connected', 'connecting' and 'connection' would be transformed into 'connect'.

C. Term_Weight Calculation

The document has N number of sentences and collection of terms in the document is denoted as $d = \{t_1, t_2 \ldots t_m\}$[4, 5]. Each term in the document can be represented using a weighting scheme called TF-ISF[4, 5]. TF is the term frequency of word in the

document. The frequency of term occurrences within a document has often been used for calculating the importance of the sentence. The score of a sentence can be calculated as the sum of the score of words in the sentence.

Number of times a term occurred in a sentence is called as the 'Term frequency'. It is represented as "tf". Term Weight is a measure used to calculate weight of term which is scalar product of term frequency and inverse sentence frequency mathematically represented as follows.A commonly used measure to assess the importance of the words in a sentence is the inverse sentence frequency, or ISF, which is defined by the formula:

$$ISF = \log N/n_i \quad (1)$$

where N is the total number of the sentences in the document, and $n_i$is the number of sentences in which word i occurs.

Term weight is calculated as

$$Term\_Weight\ (t_i) = TF(t_i) * ISF(t_i) \quad (2)$$

$$ISF = \log N/n_i$$

where i=1,2…m.

TF = term frequency of the word i in the document

ISF = Inverse Sentence Frequency

N = total number of sentence

n_i = no. of sentences in which word i occurs

D. Feature profile Generation

Feature profile is generated to capture the values of sentence-specific features of all sentences. The system work combines a feature called term feature with five features like sentence position, sentence length, sentence centrality, number of proper nouns in the sentence and number of numerical data in the sentence to generate feature profile. Feature profile is generated to capture the values of sentence-specific features of all sentences [8, 9]. The system work combines a feature called term feature with five features like sentence position, sentence length, sentence centrality, number of proper nouns in the sentence and number of numerical data in the sentence to generate feature profile.All feature score functions are used from standard weights to obtain a suitable combination of feature weights.

For Term Feature, the score of a sentence can be calculated as the sum of the score of terms in the sentence.

Term Feature (*T_F) is defined as*

$$ST\_F_1 = \sum Term\_weight(t).f(t, S_i) \quad (3)$$

where f (t ,s_i) is the frequency of each term t in sentence s_i.

Position information has been proved to be very effective in document summarization; especially in generic summarization. These position features have been proved to be very effective in generic document summarization.The position feature is defined by considering maximum positions of 5. For example, the first sentence in a document has a score value of 5/5, the second sentence has a score 4/5, third sentence has a score value of 3/5, the fourth sentence has a score 2/5, fifth sentence has a score value of 1/5 and other sentences has a score 0/5 [11].

Position Feature (P_F) is defined as

$$SP\_F_2 = [(M+1) - Position\ (S_i)]/M \quad (4)$$

Where M = maximum position or no. of sentences in document.

For Sentence Length Feature, Longer sentences usually contain more information about the documents [4, 5]. **The Length Feature (*L_F)*** is defined as

$$SL\_F3 = \frac{Length(S_i)}{no.ofWordOccuringInLongestSentence} \quad (5)$$

For Sentence Centrality Feature, There are two points to clarify in this definition of centrality. First is how to define similarity between two sentences. Second is how to compute the overall centrality of a sentence given its similarity to other sentences [4, 5]. The Sentence Centrality Feature (*C_F)* is defined as

$$SC\_F4 = \frac{words(s_i) \cap words(others)}{words(s_i) \cup words(others)} \quad (6)$$

In general the sentence that contains more proper nouns is an important one and it is most probably included in the document summary. The following formula is used to calculate the inclusion of proper

nouns (*PN_F)* in the sentence. The Sentence with Proper Noun Feature (PN_F) is defined as

$$SP\ \_\ NF\ 5 = \frac{PN\ \_\ count\ (S_i)}{Lenght\ (S_i)} \quad (7)$$

For Numerical Data Feature,the number of numerical data in sentence, sentence that contains numerical data is important and it is most probably included in the document summary. The score for this feature is calculated as the ratio of the number of numerical data that occur in sentence over the sentence length[4, 5].

$$SN\_F_6 = \frac{No.\ Numerical\ data\ in\ Si}{Sentence\ Length\ (Si)} \quad (8)$$

E. Sentence Score Calculation

The score of a sentence is the weighted sum of the scores for all terms in it. Sentence score is a measure which is used to measure how important the sentence is in document. The following formula is used to calculate the score of the sentence where f_w,f_p,f_l,f_c,f_{pn},f_{nd} are weights of word, position, length, centrality, sentence with proper noun and numerical data features. These weights are given in order to normalize the values of sentence specific features such that $f_w+f_p+f_l+f_c+f_{pn}+f_{nd}$ must be 1. Here the values assigned for f_w is 0.3, f_p is 0.1, f_l is 0.2, f_c is 0.2, f_{nd} is 0.1, f_{pn} = 0.1.

$$Sentense\ \_\ Score\ (S_i) = f_w.ST\ \_\ F(S_i) + f_p.SP\ \_\ F(S_i) + f_l.SL\ \_\ F(S_i) + f_c.SC\ \_\ F(S_i) + f_{pn}.SPN\ \_\ F(S_i) + f_{nd}.SND\ \_\ F(S_i) \quad (9)$$

F. Sentence Ranking and Ordering

Sentence score is a measure which is used to measure how important the sentence is in document. Sentences are ranked according to their score values in descending order. After ranking, the system employs the sentence ordering strategy according to their position and chronology of the original documents. A sentence is ranked higher if it is contained ranked higher if it contains many sentences which are more relevant to the given query.

G. Summary Generation

This system takes as input the set of all sentences from the input document along with their important scores. Feature profile is generated by considering word weight that is term weight, sentence position, and sentence length, and sentence centrality, proper noun in the sentence and numerical data in sentence.If these features are sufficiently relevant for the single document summarization task, the sentence extraction techniques compute score for each sentence based on features. It was defined different sentence features that considered important for a generic summary. Therefore, the features score of each sentence that we described in the previous section are used to obtain the significant sentences. In this system, a set of highest score sentences are extracted as document summary based on compression rate. The numbers of sentences are extracted according to 40% compression rate from the source document.

IV. SYSTEM DESIGN AND IMPLEMENTATION

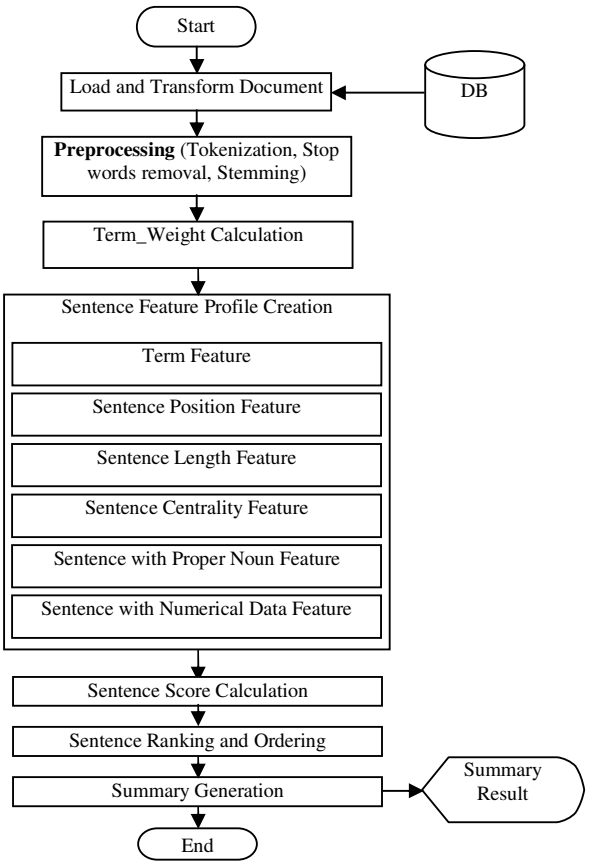

Fig 1. System Flow Diagram

According to generate text summarization based on feature profile, firstly, transform the documents that want to summarize. In single document text summarization, it takes a single document as an input to perform summarization and produce a single output document.In this process, it uses statistical and linguistic features of the sentences to decide the most relevant sentences in the given input document.

Then Tokenization, Stop words Removal, Porter Stemmer algorithm is applied for preprocessing. The six features calculate from each sentence for sentence feature profile creation process. Each sentences of the document is represented by sentence score. Then all document sentences are ranked in a descending order according to their scores. A set of highest score sentences are extracted as document summary. This system is implemented by using C# programming language and MySQL database. This system can be used word files and text files for summarization.

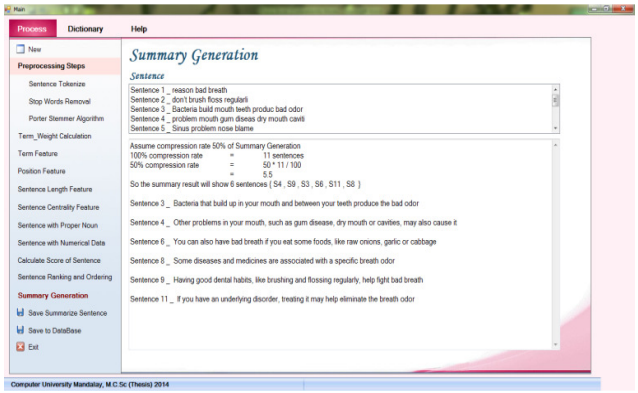

Fig 2. Summary generation for sample document

H. The Evolution of Text Summarization Approaches

For the extraction task we are dealing here, things are a bit easier. We generated the extract of a system with the desired summary at 40% compression ratio and Information retaining is nearly full text. Therefore,we extracted the appropriate number of sentences according to 40% compression rate. It has been proven that the extraction of 40% of sentences from the source document can be as informative as the full text of a document.

I. Data Compression with less Information Loss

How much shorter the summary is than the original. It can be represented as

$$\text{Compression Ratio CR} = \frac{\text{Length\_of\_Summary\_needed}}{\text{Length\_of\_Full\_Text}}$$

How much information is still retained in the summary is represented as

$$\text{Retention Ratio} = \frac{\text{Information\_in\_Summary}}{\text{Information\_in\_Full\_Text}}$$

A good summary is one CR is small (tending to zero) while RR is large (tending to unity).

V. Conclusions

This system discusses about feature profile oriented sentence extraction based summarization of single document and also represents a technique for extracting key sentences from a document in order to use such sentences as a summary of the same. Sentences are selected in order based on the sentence score. In the health news document, user can get summary document by using this system. This system can eliminate the overall document by extracting the summary sentences. Feature extraction include extractingfeatures associated with the sentence (such as sentencenumber, number of words in that sentence and so on)and the features associated with words (such as thenamed entities, the term frequency and so on). Thesummary generation component calculates the score foreach sentence based on the features that were identified the feature extraction module.

J. Advantages of the System

This process reduces information overloading because only a summary needs to be read instead of reading the entire document [3]. The goal of text summarization is to present the most important information in a shorter version of the original text while keeping its main content and helps the user to quickly understand large volumes of information. Because of calculating the sentence and words scores, user can easy to know the important features in sentence of documents and can study sentence feature profile creating approach and sentence score calculation in this system.

K. Limitation and Further Extension

Despite the fact that text summarization has traditionally been focused on text input, the input to the summarization process can also be multimedia information, such as images, video or audio, as well as on-line information or hypertexts. But this system can't summarize these inputs. Document summarization with these inputs will be further extension. We would like to extend this system to be able to use abstraction summary by using relevance and concept of knowledge documents for entity and discourse-level.

The future work includes multi-document summarization system with variousNLP tools. Thus it will improve the processing speed and efficiency of the system. This system can extend by using sentence scoring calculation, and sentence ordering for multi-document text summarization. Another feature research line could consist in adding more knowledge to the system, exploring new approaches based on semantic relationships (for instance, WordNet relations such as synonymy or hyponymy) and graph-based relations.

REFERENCES

[1] Ani Nenkova, Kathleen McKeown, "Automatic Summarization", Foundations and Trends in Information Retrival, Vol. 5, 2011.
[2] Elena Lloret, O´scar Ferra´ndez, Rafael Mun˜oz, and Manuel Palomar, "A Text Summarization Approach under the Influence of Textual Entailment," University of Alicante, Spain.
[3] Gunes Erkan, Dragomir R. Radev, "LexRank: Graph-Based Lexical Centrality as Salience in Text Summarization," University of Michigan, 2004.
[4] J. Goldstain, M. kantrowitz, V. Mittal, and J. Carbonell, "Summarizing text documents: Sentence selection and evaluation metrics", In Proceedings of ACM SIGIR'99, (Berkeley, CA), Aug. 1999.
[5] Ladda Suanmali, Naomie Salim and Mohammed Salem Binwahlan, "Fuzzy Logic Based Method for Improving Text Summarization", (IJCSIS) International Journal of Computer Science and Information Security, Vol.2, No.1, 2009.
[6] Martin Hassel, "Resource Lean and Portable Automatic Text Summarization," KTH Computer Science and Communication, Stockholm, Sweden, 2007.
[7] Meru Brunn, Yllias Chali, Christopher J. Pinchak, "Text Summarization Using Lexical Chains", Department of Mathematics and Computer Science, University of Lethbridge at DUC 2001.
[8] Mohamed Abdel Fattah and Fuji Ren, "Automatic Text Summarization", World Academy of Science, Engineering and Technology, Vol.2, 2008.01.28.
[9] R. Barzilay and M. Elhadad, "Using lexical chains for text summarization," in Proceedings of the Workshop on Intelligent Scalable Text Summarization, (Madrid, Spain), Aug. 1997.
[10] The German Research Foundation, "Chapter (1), Tokenization," Computational Linguistics and Phonetics, Universitat des Saarlandes.
[11] You Ouyang, Wenjie Li, Qin Lu, Renxian Zhang, "A Study on Position Information in Document Summarization", Department of Computing, the Hong Kong Polytechnic University, August. 2010.

