Text Summarization Extraction System (TSES) Using Extracted Keywords

Rafeeq Al-Hashemi
Faculty of Information Technology, Al-Hussein Bin Talal University,Jordan

Abstract *A new technique to produce a summary of an original text investigated in this paper. The system develops many approaches to solve this problem that gave a high quality result. The 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 keyphrases in the text by implementing a new algorithm through ranking the candidate words. The system uses the extracted keywords/keyphrases to select the important sentence. Each sentence ranked depending on many features such as the existence of the keywords/keyphrase 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. This stage reduced the amount of the candidate sentences in the summary in order to produce a qualitative summary using KFIDF measurement.*

Keywords: *Text Summarization, Keyphrase Extraction, Text mining, Data Mining, Text compression .*

Received October 2, 2009; Accepted March 13, 2010

1. Introduction

Text mining can be described as the process of identifying novel information from a collection of texts. By novel information we mean associations, hypothesis that are not explicitly present in the text source being analyzed [9].

In [4] Hearst makes one of the first attempts to clearly establish what constitutes text mining and distinguishes it from information retrieval and data mining. In [4] Hearst paper, metaphorically describes text mining as the process of mining precious nuggets of ore from a mountain of otherwise worthless rock. She calls text mining the process of discovering heretofore unknown information from a text source.

For example, suppose that a document establishes a relationship between topics A and B and another document establishes a relationship between topics B and C. These two documents jointly establish the possibility of a novel (because no document explicitly relates A and C) relationship between A and C.

Automatic summarization involves reducing a text document or a larger corpus of multiple documents into a short set of words or paragraph that conveys the main meaning of the text. Two methods for automatic text summarization they are Extractive and Abstractive. Extractive methods work by selecting a subset of existing words, phrases, or sentences in the original text to form the summary. In contrast, abstractive methods build an internal semantic representation and then use natural language generation techniques to create a summary that is closer to what a human might generate. Such a summary might contain words not explicitly present in the original. The abstractive methods are still weak, so most research focused on extractive methods, and this is what we will cover.

Two particular types of summarization often addressed in the literature. keyphrase extraction, where the goal is to select individual words or phrases to “tag” a document, and document summarization, where the goal is to select whole sentences to create a short paragraph summary [9, 2, 7].

Our project uses an extractive method to solve the problem with the idea of extracting the keywords, even if it is not existed explicitly within the text. One of the main contributions of the proposed project is the design of the keyword extraction subsystem that helps to select the good sentences to be in the summary.

2. Related Work

The study of text summarization [3] proposed an automatic summarization method combining conventional sentence extraction and trainable classifier based on Support Vector Machine. The study [3] introduces a sentence segmentation process method to make the extraction unit smaller than the original sentence extraction. The evaluation results show that the system achieves closer to the human constructed summaries (upper bound) at 20% summary rate. On the other hand, the system needs to improve readability of its summary output.

In the study of [6] proposed to generate synthetic summaries of input documents. These approaches, though similar to human summarization of texts, are limited in the sense that synthesizing the information requires modelling the latent discourse of documents which in some cases is prohibitive. A simple approach to this problem is to extract relevant sentences with respect to the main idea of documents. In this case, sentences are represented with some numerical features indicating the position of sentences within each document, their length (in terms of words, they contain), their similarity with respect to the document title and some binary features indicating if sentences contain some cue-terms or acronyms found to be relevant for the summarization task. These characteristics are then combined and the first p% of sentences having highest scores is returned as the document summary. The first learning summarizers have been developed under the classification framework where the goal is to learn the combination weights in order to separate summary sentences from the other ones.

In [2] presents a sentence reduction system for automatically removing extraneous phrases from sentences that are extracted from a document for summarization purpose. The system uses multiple sources of knowledge to decide which phrases in an extracted sentence can be removed, including syntactic knowledge, context information, and statistics computed from a corpus which consists of examples written by human professionals. Reduction can significantly improve the conciseness of automatic summaries.

3. The proposed System Architecture

The following diagram figure.1 represents the proposed system:

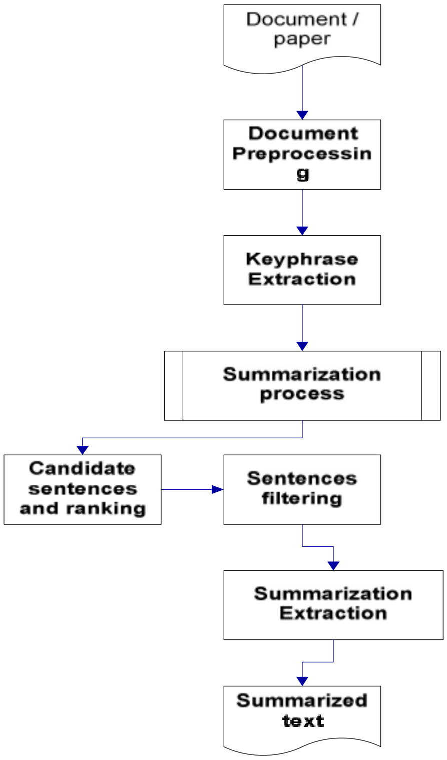


Figure 1. System architecture.

The model consists of the following stages:

4. Preprocessing

The pre- processing is a primary step to load the text into the proposed system, and make some processes such as case-folding that transfer the text into the lower case state that improve the accuracy of the system to distinguish similar words. The pre-processing steps are:

4.1. Stop Word Removal

The procedure is to create a filter for those words that remove them from the text. Using the stop list has the advantage of reducing the size of the candidate keywords.

4.2. Word Tagging

Word tagging is the process of assigning P.O.S) like (noun, verb, and pronoun, Etc.) to each word in a sentence to give word class. The input to a tagging algorithm is a set of words in a natural language and specified tag to each. The first step in any tagging process is to look for the token in a lookup dictionary. The dictionary that created in the proposed system consists of 230,000 words in order to assign words to its right tag. The dictionary had partitioned into tables for each tag type (class) such as table for (noun, verb, Etc.) based on each P.O.S category.
The system searches the tag of the word in the tables and selects the correct tag (if there alternatives) depending on the tags of the previous and next words in the sentence.

4.3. Stemming

Removing suffixes by automatic means is an operation which is especially useful in keyword extraction and information retrieval.

The proposed system employs the Porter stemming [10] algorithm with some improvements on its rules for stem.
Terms with a common stem will usually have similar meanings, for example:
(CONNECT, CONNECTED, CONNECTING, CONNECTION, CONNECTIONS)

Frequently, the performance of a keyword extraction system will be improved if term groups such as these are conflated into a single term. This may be done by removal of the various suffixes -ED, -ING, -ION, IONS to leave the single term CONNECT. In addition, the suffix stripping process will reduce the number of terms in the system, and hence reduce the size and complexity of the data in the system, which is always advantageous.

5. Keyphrase Features

The system uses the following features to distinguish relevant word or phrase (Keywords):

- Term frequency
- Inverse Document Frequency
- Existence in the document title and font type.

- Part of speech approach.

5.1. Inverse Document Frequency (IDF)

Terms that occur in only a few documents are after more valuable than ones that occur in many. In other words, it is important to know in how many document of the collection a certain word exists since a word which is common in a document but also common in most documents is less useful when it comes to differentiating that document from other documents [10]. IDF measures the information content of a word.

The inverse document frequency is calculated with the following formula [8]:

$$Idfi=tf * \log(N/ni) \qquad (1)$$

Where N denotes the number of documents in the collection, and ni is the number of documents in which term i occurs.

5.2. Existence in the Document Title and Font Type

Existence in the document title and font type is another feature to gain more score for candidate keywords. Since the proposed system gives more weight to the words that exists in the document title because of its importance and indication of relevance. Capital letters and font type can show the importance of the word so the system takes this into account.

5.3. Part of Speech Approach

After testing the keywords that extracted manually by the authors of articles in field computer science we noted that those keywords fill in one of the following patterns as displayed in table (1). The proposed system improves this approach by discover a new set of patterns about (21 rule) that frequently used in computer science. This linguistic approach extracts the phrases match any of these patterns that used to extract the candidate keywords. These patterns are the most frequent patterns of the keywords found when we do experiments.

5.4. Keyphrase Weight Calculation

The proposed system computes the weight for each candidate keyphrase using all the features mentioned earlier. The weight represents the strength of the keyphrase, the more weight value the more likely to be a good keyword (keyphrase). We use these results of the extracted keyphrases to be input to the next stage of the text summarization.

The range of scores depends on the input text. The system selects N keywords with the highest values.

Table 1. P.O.S. Patterns.

no	POS Patterns
1.	<adj> <noun>
2.	<noun> <noun>
3.	<noun>
4.	<noun> <noun> <noun>
5.	<adj> <adj> <noun>
6.	<adj>
7.	<adj> <adj> <noun> <noun>
8.	<noun> <verb>
9.	<noun> <noun> <noun> <noun>
10.	<noun> <verb> <noun>
11.	<noun> <adj> <noun>
12.	<prep> <adj> <noun>
13.	<adj> <adj> <adj> <noun pl>
14.	<noun> <adj> <noun pl>
15.	<adj> <adj> <adj> <noun>
16.	<noun pl> <noun>
17.	<adj> <propern>
18.	<adj> <noun> <verb>
19.	<adj> <adj>
20.	<adj> <noun> <noun>
21.	<noun> <noun> <verb>

5.5. Classification

The proposed system tries to improve the efficiency of the system by categorizing the document by trying to assign a document to one or multiple predefined categories and to find the similarity to other existing documents in the training set based on their contents.

The proposed system applies classical supervised machine learning for document classification, by depends on the candidate keywords that are extracted till this step and categorize the current document to one of the pre defined classes. This process has two benefits one for document classification and the second for feed backing this result to filtering the extracted keywords and to increase the accuracy of the system by discarding the candidate keywords that are irrelevant to the processed document field, since the proposed system is a domain specific. Instance-based learning method is the classification method that the proposed system implements. First it computes the similarity between a new document and the training documents that calculates the similarity score for each category, and finds out the category with the highest category score.

The documents classified according to the following equation (2) base on the probability of document membership to each class:

$$\left[P(C_k) = \frac{\text{count of word i in class } C_k \text{ documents}}{\text{count of words in class } C_k \text{ document}} \right] \qquad (2)$$

$$\text{Doc. Class} = \text{Max } P(Ck) \qquad (3)$$

First, the system is learned by training the system with example documents; second, it is evaluated and tested by using the test example documents. Algorithm (1) is the classification algorithm. The corpus we used is in general computer science and categories of database, Image processing, AI. The size of training set is 90 documents and tested by 20 documents.

Algorithm (1): Text classification

```
Input: candidate keywords table, Doc;
Output: document class;
Begin
    For C=1 to 3 {no of classes =3}
        K=K+1
        For I= 1 to n
            $P_C = P_C + cont(Wi)/cont(W)$
        Next
        $prob(C_k) = P_C$
    Next
    For j=1 to 2
    For s=j+1 to 3
        If prob(j) > prob(s) then
            Class(Doc)= max(prob(j))
    Next
    Next
End.
```

6. Sentences Selection Features

- Sentence position in the document and in the paragraph.
- Keyphrase existence.
- Existence of indicated words.
- Sentence length.
- Sentence similarity to the Document class.

6.1. Existence of Headings Words

Sentences occurring under certain headings are positively relevant; and topic sentences tend to occur very early or very late in a document and its paragraphs.

6.2. Existence of Indicated Words

By indicated words, we mean that the existence of information that helps to extract important statements. The following is a list of these words:

Purpose: Information indicating whether the author's principal intent is to offer original research findings, to survey or evaluate the work performed by the others, to present a speculative or theoretical discussion.

Methods: Information indicating the methods used in conducting the research. Such statement may refer to experimental procedures, mathematical techniques.

[Conclusions or findings Generalization or Implications]: Information indicating the significance of the research and its bearing on broader technical problems or theory such as [Recommendations or suggestions].

6.3. Sentence Length Cut-off feature

Short sentences tend not to be included in summaries. Given a threshold, the feature is true for all sentences longer than the threshold and false otherwise.

7. Post Processing

The system makes filtering on the generated summary to reduce the number of the sentences, and to give more compressed summary. The system at first removes redundant sentences; second the system removes the sentence that has a similar to another one more than 65%. This is necessary because authors often repeat the idea using the same or similar sentences in both introduction and conclusion sections. The similarity value is calculated as the vector similarity between two sentences represented as vectors. That is, the more common words in two sentences, the more similarity they are. If the similarity value of two sentences is greater than a threshold, we eliminate one whose rank based on the features is lower that of the other. For the threshold value, we used 65% in the current implementation.

The system implements a filter that replaces incomplete keyphrase by complete one selected from the input text which is in a suitable form such as scientific terms. This filter depends on the KFIDF measurement as in the equation (4), [1]. The filter selects the term that is more trivial and used in the scientific language:

(E.g. neural →neural network , neural network →artificial neural network). The KFIDF computed for each keyword, the more trivial keyword and frequently used in the class gains more value of KFIDF. Again to the example above if the candidate keyword is neural the filter finds phrases within the system near the word neural then it will select the one which has more KFIDF value (e.g. the keyword is “neural” and the phrases found are “neural network” and “neural weights” by applying KFIDF measurement the first phrase will have greater value depending on the trained documents existing in the class.

$$\text{KDIDF(w,cat)= docs(w,cat)} \times LOG\left(\frac{n \times |cats\,|}{cats\,(word\,)} + 1\right) \qquad (4)$$

docs(w,cat)= number of documents in the category cat

containing the word w

n- smoothing factor

cats(word) = the number of categories in which the word occurs

8. Measurements of Evaluation

Using the generated abstract by the author as the standard against which the algorithm-derived abstract. The results evaluated by Precision and Recall measurements.

Precision (P) and Recall (R) are the standard metrics for retrieval effectiveness in information retrieval. They calculated as follows:

$$P = tp / (tp + fp) \qquad (5)$$

$$R = tp / (tp + fn) \qquad (6)$$

Where tp = sentences in the algorithm-derived list also found in the author list; fp = sentences in the algorithm-derived list not found in the author list; fn = sentences in the author list not found in the algorithm derived list. They stand for true positive, false positive and false negative, respectively. In other words, in the case of sentence extraction, the proportion of automatically selected sentences that are also manually assigned sentences is called precision. Recall is the proportion of manually assigned sentences found by the automatic method [11, 12].

9. Results

Documents from the test set have been selected, and the selected sentences to be in the summary presented in table 2 below:

Table 2. Experiment results.

Text #	Automatic selected	Manual selected	Matched	Precision
1	4	3	3	75%
2	9	8	6	67%
3	8	8	6	75%
4	10	8	6	60%
5	10	10	9	90%
6	14	10	9	64%
7	13	12	10	77%
8	24	22	19	79%
9	30	26	22	73%
10	31	21	13	42%
Overall Precision				70%

10. Conclusion

The work presented her depends on the keyphrases extracted by the system and many other features extracted from the document to get the text summary as a result. This gave the advance of finding the most related sentences to be added to the summary text. The system gave good results in comparison to manual summarization extraction. The system can give the most compressed summary with high quality. The main applications of this work are Web search Engines, text compression and word processor.

References

[1] Feiyu Xu, et. al.; "A Domain Adaptive Approach to Automatic Acquisition of Domain Relevant Terms and their Relations with Bootstrapping"; *In Proceedings of the 3rd International Conference on Language Resources an Evaluation (LREC'02)*, May 29-31, Las Palmas, Canary Islands, Spain, 2002.

[2] Hongyan Jing; "Sentence Reduction for Automatic Text Summarization"; Proceedings *of the sixth conference on Applied natural language processing*, Seattle, Washington, pp.310 – 315, 2000.

[3] Kai ISHIKAWA et. al.; "Trainable Automatic Text Summarization Using Segmentation of Sentence"; Multimedia Research Laboratories, NEC Corporation 4-1-1 Miyazaki Miyamae-ku Kawasaki-shi Kanagawa 216-8555, 2003.

[4] M. Hearst; "Untangling text data mining", *In Proceedings of ACL'99*: the 37th Annual Meeting of the Association for Computational Linguistics, 1999.

[5] M.F.Porter; "An algorithm for Suffix Stripping"; originally published *in \Program*, *14*\ no. 3, pp 130-137, July 1980.

[6] Massih Amini; "Two Learning models for Text Summarization"; Canada, Colloquium Series, 2007-2008 Series.

[7] Nguyen Minh Le, et. al.; "A Sentence Reduction Using Syntax Control". *Proceedings of the sixth international workshop on Information retrieval with Asian languages* Volume 11, 2003.

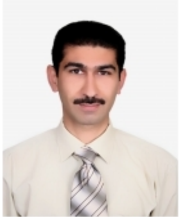
[8] Otterbacher, et. al.; "Revisions that improve cohesion in multi- document summaries"; *in ACL workshop on text summarization*, Philadelphia, 2002.

[9] Sholom M. weiss. Nitim Indurkhya, Tong Zhag, fred J. Damerau, *Text mining predication methods for analyzing unstructured information*. Spring, USA, 2005.

[10] Susanne viestan, "Three methods for keyword extraction", MSc. Department of linguistics, Uppsds University, 2000.

[11] U. Y. Nahm and R. J. Mooney. "Text mining with information extraction". *In AAAI 2002 Spring Symposium on Mining Answers from Texts and Knowledge Bases*, 2002.

[1] William B. Cavnar, and John M. Trenkle; "NGram- Based Text Categorization"; *In Proc. of 3rd Annual Symposium on Document Analysis and Information Retrieval*, pp.161-169, 1994.



Rafeeq Al-Hashemi obtained his Ph.D. in Computer Science from the University of Technology. He is currently a professor of Computer Science at Al-Hussein Bin Talal University in Jordan. His research interests include Text mining, Data mining, and image compression. Dr. Rafeeq is also the Chairman of Computer Science in the University.