A neural-based text summarization system

S. P. Yong, A. I. Z. Abidin & Y. Y. Chen
IT/IS Department, Universiti Teknologi PETRONAS, Malaysia

Abstract

The number of electronic documents as a media of business and academic information has increased tremendously after the introduction of the World Wide Web. Ever since, instances where users being overloaded with too much electronic textual information are inevitable. The users may only be interested in shorter versions of text documents but are overloaded with lengthy texts. The objective of the study is to develop a text summarization system that incorporates learning ability by combining a statistical approach, keywords extraction, and neural network with unsupervised learning. The system is able to learn to classify sentences when well trained with sufficient text samples. Users with strong background in writing English summaries have subjectively evaluated the outputs of the text summarization system based on contents. With the average contents score of 83.03%, the system is regarded to have produced an effective summary with most of the important contents of the original text extracted without compromising the summary's readability.
Keywords: keyword extraction, neural network, unsupervised learning.

1 Introduction

The proliferation of electronic documents as a media for business and academic information in the World Wide Web has resulted in users being overloaded by electronic texts. Though users can sort out the documents through various search engines, the engines usually do a poor approximation. The engines only show the initial lines of the document. Users who do not use keywords to search for intended document effectively might come across a vast quantity of hyperlinks.

Text summarization is an emerging field at the intersection of several research areas, including natural language processing, machine learning and information retrieval. It is essential to be able to extract the gist of the electronic documents by having a text summarization system to fully utilize these documents

WIT Transactions on Information and Communication Technologies, Vol 37, © 2006 WIT Press
www.witpress.com, ISSN 1743-3517 (on-line)
doi:10.2495/DATA060191

effectively. Summarization is important in some context to help people understand facts or to gain knowledge.

It is common sense that the main ideas of the original text document should guide the selection of information in producing a summary [1, 2]. There has been a long history of research in text summarization. One of the most popular and successful research on text summarization is Luhn's auto-extract statistical system [4]. In his research, his assumption is that frequency data can be used to extract words and sentences to represent a document. Research on text summarization approaches has been evolving ever since Luhn's ideas were proposed. Other approaches include domain-based system that was inspired by cognitive science theories and domain-independent summarization.

Currently, there are a few text summarization systems that can be seen on the Internet such as *NetSumm* developed by BT Exact, England in 1996, *Pertinence* by Lehman/Bouvet and *Extractor* by Interactive Information Group NRC (National Research Council, Canada). Each of these systems employed different techniques that each have their own merits and limitations.

This research project aims to discover a neural-based approach with unsupervised learning to summarize texts. The objective of this work is to develop a text summarization system, named *TextSum,* which incorporates learning ability by combining statistical approach, keywords extraction and neural network with unsupervised learning. The proposed system should produce a summary of any text documents in English.

2 Related work

Summaries can be created by extraction. Extraction is merely to identify the most important information from a text [3]. The less important information is omitted. The software *NetSumm* is an example of a system using extraction to develop summaries.

A typology of summaries can be made on four sets of parameters: coverage, informative-ness, selectivity, and recipients [5]. Text coverage includes summaries of individual texts or a collection of texts. Selective summaries are made for specific purposes. Summaries can be made for specific groups of recipients. They are possible only when the specific needs of users are predictable. However, summaries can also be undirected, i.e. for use in information system where the background knowledge of users cannot be predicted. The production of informative summaries intended for unspecified needs is probably the most difficult of all.

Joel applies several preprocessing methods to the original documents, namely case folding, stemming, removal of stop words and n-grams [6]. The text summarization algorithm developed in this research combined the 3 steps in text summarization process in which it is based on computing the value of a term frequency-inverse sentence frequency *tf-isf* measured for each word. The output is a summary consisting of all sentences that have high values of *tf-isf*. The system has been evaluated on real-world documents and the result is satisfactory.

Taeho Jo shows that using back propagation neural network to extract keywords outperforms the equation in distinguishing keywords [7]. In his paper, he describes how back propagation is applied for the selection of keywords. Sample documents are necessary to determine two main features of each word: inverted document frequency *idf,* and Inverted Term Frequency *itf*. The input features of each word in the given document include Term Frequency *tf*, *idf*, *itf*, Title *t*, First Sentence *fs*, Last Sentence *ls*. The features, *idf* and *itf*, require sample documents to maintain the robustness of the system before computing the value of each word. Output features for each word are the word judged to keyword *K* and the word judged to non-keyword *N*. Both features are represented in binary values.

Many machine-learning approaches for information access require a large amount of supervision in the form of labeled training data. One of the ways to improve a generic document summarization system is by using unsupervised and semi-supervised learning approach [8]. From a machine learning perspective, summarization is typically a task in which a lot of unlabelled data and very few labeled texts so that semi-supervised learning seems well suited for the task.

In general, previous work employed three main steps in text summarization: preprocess text, determine salient sentences and assemble summaries. In the literature, not many previously developed systems employ neural-based approach with unsupervised learning. *NeuralSumm* that was developed in 2003 is one of the systems that applied neural network with unsupervised learning. It uses self-organizing map in summarization task [11]. However, *NeuralSumm* requires user to feed in some training data that may not be convenient for users with no knowledge of the system. Furthermore, it is reported that it performs very poorly for news texts [11].

3 Text summarization: neural-based approach

The system architecture with main modules of the proposed text summarization system *TextSum* is shown in Figure 1. Text preprocessing, keywords extraction, and summary productions are tasks involved in summarizing electronic texts using *TextSum.*

3.1 Text pre-processing subsystem

The system applies two pre-processing methods to the original document: stop words removal and stemming. Stop words are the most frequently appeared words in text. Therefore, they carry little information about the content of a document. For instance, words like “the”, “a”, can”, and “will” are typical stop words. Stemming consist of converting each word to its stem. Eliminating suffixes and prefixes is necessary to get the stem of a word. Porter’s algorithm [10], originally developed for the English Language, is used to pre-process text.

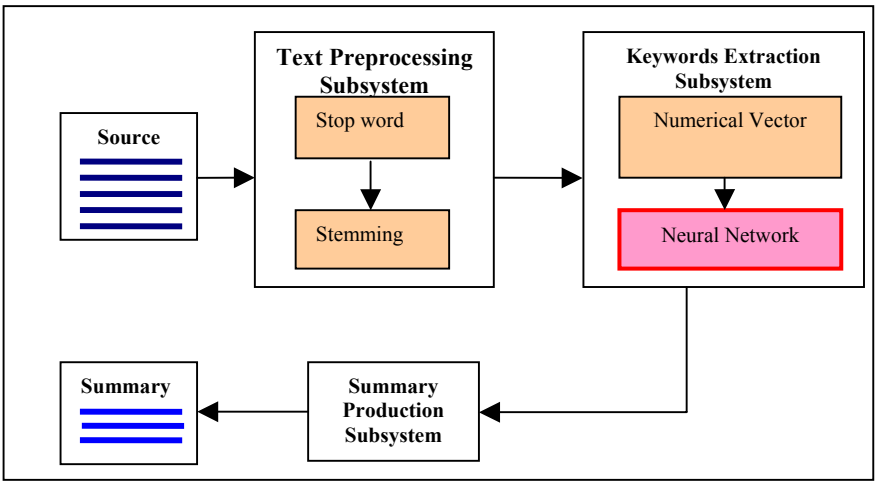

Figure 1: *TextSum* system architecture.

3.2 Keywords extraction subsystem

A group of text documents are taken as a sample. These documents are called sample text documents, which are heterogeneous documents in their content. These sample text documents will be represented into numerical vector. First of all, sentences of a document will be separated. The end of a sentence is defined as a "." followed by a space or new line character. After delimiting them, the main features of each word need to be determined: The features *tf*, and *isf* . *tf-isf(w,s)* is computed by the formula [6] in eqn. (1):

$$tf - isf(w,s) = tf(w,s) * isf(w) \tag{1}$$

where the term frequency *tf(w,s)* is the number of times that the word *w* appears in sentence *s*, and the inverse sentence frequency *isf(w)* is given by the formula [6] in eqn. (2):

$$isf(w) = log(s / sf(w)) \tag{2}$$

where the sentence frequency *sf(w)* is the number of sentences in which the word *w* occurs. The eqn. (1) serves as the input features of each word to the neural network.

The type of neural network adopted in this work is competitive network. The neurons in a competitive layer distribute themselves to recognize frequently presented input vectors.

The input vector is generated from the previous stage using *tf-isf* formula. The distance function of the proposed competitive network accepts *p* and $IW^{1,1}$ (input weight matrix) and produces a vector having elements that are the negatives of

the distance between the input vector p and input vectors $iIW^{1,1}$ (i = the ith neuron) formed from the rows of the input weight matrix. The net input n^1 of a competitive layer is computed by a formula [9] in eqn. (3):

$$\sqrt{p^2 + IW^2} = \pm\sqrt{Q}$$

taking negative value of Q

$$n^1 = -Q + \sqrt{b} \tag{3}$$

where b = 1.

The maximum net input a neuron can have is 0 if all biases are 0. This occurs when the input vector p equals to that neuron's weight vector. The competitive transfer function accepts a net input vector for a layer and returns neuron outputs of 0 for all neurons except for the winner, the neuron associated with the most positive element of net input n^1. The winner's output is 1. If all biases are 0, then the neuron whose weight vector is closest to the input vector has the least negative net input and, therefore, wins the competition to output a 1.

The weights of the winning neuron are adjusted by using Kohonen's learning rule. For example, suppose the ith neuron wins, the elements of the ith row of the input weight matrix are adjusted using the formula [9] in eqn. (4):

$$iIW^{1,1}(q) = iIW^{1,1}(q-1) + \alpha(p(q) - iIW^{1,1}(q-1)) \tag{4}$$

The Kohonen rule allows the weights of a neuron to learn an input vector. Therefore it is useful in recognition application. The neuron whose weight vector is closest to the input vector is updated to be even closer. As a result, the winning neuron is more likely to win the competition the next time when there is a similar vector and is less likely to win when a different vector is presented. As more and more inputs are presented, each neuron in the layer closest to a group of input vectors soon adjusts its weight vector toward those input vectors.

3.3 Summary production subsystem

The output from the competitive network needs to be decoded into words to identify the keywords. The system chooses sentences that have the keywords as part of the summary. When selecting sentences, there are no stop words in the identified keywords. The resulted summary will not be accurate if the sentences were selected based on stop words. Hence, it is vital to run through another round of stop words checking procedure before selecting sentences.

4 Results and discussion

Each of the modules (text pre-processing, keyword extraction, and summary production) functions as expected. The text preprocessing and the keyword extraction modules work in the background and have no meaningful outputs to users, while the summary production module is the subsystem producing a

readable summary. Since humans potentially benefit from *TextSum*, humans' involvement in the evaluation of the system is very important.

4.1 User evaluation

An intrinsic evaluation (subjective evaluation) is the type of user's evaluation applied for this project. The extraction of important contents is the main criterion in evaluating the quality of the summary. A total of 30 users with strong backgrounds in writing English summaries are each equipped with 5 different text documents. The steps involved in evaluating a report are as follows:

- Each user reads the original text and then highlights the document's important sentences.
- Next, each user reads the summary produced by *TextSum* and compares the content of the summary against the highlighted sentences in the original text.
- Since different documents have different number of important contents, each user then gives 1 mark if an important content is both highlighted in the original text and is extracted by *TextSum*.
- Later, for each report, the average of the number of highlighted contents (*HC*) and the average of the marks for summary contents (*SC*) are calculated.
- Finally, the calculated averages are truncated to produce whole numbers.

The document type, the average number of *HC*, the average number of *SC*, and the accuracy of *TextSum* in extracting important contents are tabulated in Table 1.

Table 1: Contents evaluated by English experts.

No	Documents Type	$A = trunc\left(\frac{\sum_{i=1}^{30} HC_i}{30}\right)$	$B = trunc\left(\frac{\sum_{i=1}^{30} HS_i}{30}\right)$	$\left(\frac{B}{A}\right) \times 100\%$
1	Technical Paper on Artificial Intelligence	11	9	81.81%
2	Technical Paper on Medicine	12	11	91.67%
3	News Article	10	8	80.00%
4	News Article	8	6	75.00%
5	Product Description	15	13	86.67%

Based on the experiment, *TextSum's* average contents score is 83.03%. The grading scale of Universiti Teknologi Petronas (UTP) is used as a benckmark to measure *TextSum's* performance. At UTP, a student obtains a grade of A for a score $S >= 85$ and a grade of A- for $80 <= S < 85$. With the average contents score of 83.03%, *TextSum* can be regarded to have produced a fairly good summary without compromising the readability.

In general, users perceive *TextSum* to have an ability to extract most of the important contents from the original text. The users feel that busy people would benefit from the system by getting an overall picture of bulky content before actually reading the whole document. Users may not need to spend too much time reading lengthy text documents.

In recent research on text summarization, many developed systems use statistical approaches based on Luhn's work [4], linguistic or neural-based with supervised learning approach, such as Taeho Jo's work. However, in our work, statistical, keyword, and neural-based methods are combined to extract the important sentences to put as part of a summary.

5 Conclusion

In general, very few previously developed systems employ neural-based approach, specifically unsupervised learning. *TextSum* is designed to incorporate intelligence, allowing the system to learn on how to classify keywords. Using *TextSum*, users may not need to feed in some training data as in using *NeuralSumm*.

The proposed competitive network in *TextSum* serves as the heart of the system. The architecture of the competitive network in *TextSum* has been carefully designed, as it will directly affect the system's output. The evaluation results on the system are satisfactory. Overall, the experimental results show that *TextSum* is regarded to have produced fairly good summaries with an average score on content of 83.03% for 5 different reports (2 news articles, 2 technical papers, and 1 product description).

The performance of *TextSum* can be improved by training the competitive network with multiple sample documents (e.g. > 100) in similar fields to increase robustness of the network. Currently, the network is trained with three report types: news articles, technical papers and product descriptions. For future enhancement, it is recommended that the network be trained with other report types such as legal documents and financial news to allow users from different professions to utilize the system.

References

[1] Marcu, D. The Theory and Practice of Discourse Parsing and Summarization. Cambridge, MA. The MIT Press (2000).

[2] Mani, I. Automatic Summarization. John Benjamins Publishing Co., Amsterdam (2001).

[3] Hovey, Eduard and Chin Y.L. <http://www.isi.edu/natural-language/projects/SUMMARIST.html>. Accessed in June 2004.
[4] C. J. van Rijsbergen. 1979 <http://www.dcs.gla.ac.uk/Keith/pdf/Chapter2.pdf>. Accessed in June 2004.
[5] Dangstuhl, S and Hutchins, J. December 1993. <http://ourworld.compuserve.com/homepages/WJHutchins/Dagstuhl.htm>. Accessed in June 2004.
[6] Joel, L. <http://www.ppgia.pucpr.br/~alex/pub_papers.dir/PADD2000.ps>. Accessed in June and July 2004.
[7] Taeho Jo. 2000. <http://www.site.uottawa.ca/~tjo018/Publication/ic2003_06_02.pdf>. Accessed in September 2004.
[8] Amini, Massih-Reza and Gallinari, Patrick. 2001. <http://portal.acm.org/citation.cfm?id=645805.670138>. Accessed in August 2004.
[9] Dr. Thang, K.F. 2004, Introduction to MATLAB & Neural Network Toolbox.
[10] Porter,M.F. <http://www.tartarus.org/~martin/PorterStemmer/>. Accessed in August 2004.
[11] NeuralSumm: Neural Network for Summarization <http://www.nilc.icmc.usp.br/~thiago/NeuralSumm.html>. Accessed in December 2004.