In Proceedings of the 2003 IEEE/WIC International Conference on Web Intelligence (WI-2003), October 13-16, 2003. Halifax, Canada.

Generic Text Summarization Using Local and Global Properties of Sentences

Chuleerat Jaruskulchai and Canasai Kruengkrai
Intelligent Information Retrieval and Database Laboratory
Department of Computer Science, Faculty of Science
Kasetsart University, Bangkok, Thailand
g4364115,fscichj@ku.ac.th

Abstract

As the enormous amount of on-line text grows on the World-Wide Web, the development of methods for automatically summarizing this text becomes more important. In this paper, we propose an alternative approach for extracting the most salient sentences from the original document to form a summary. The idea of our approach is to exploit both the local and the global properties of sentences. The local property can be considered as clusters of significant words within each sentence, while the global property can be though of as relations of all sentences in a document. These two properties are combined for ranking and extracting summary sentences. Experimental results show that our approach achieves acceptable performance.

1. Introduction

As the enormous amount of on-line text grows on the World-Wide Web (WWW), the development of methods for automatically summarizing this text becomes more important. Basically, text summarization is problem of condensing a source text into a shorter version preserving its information content. It can be broadly classified into two approaches: abstraction and extraction. In contrast to abstraction that requires using heavy machinery from natural language processing (NLP), including grammars and lexicons for parsing and generation [6], extraction can be easily viewed as the process of selecting important excerpts (sentences, paragraphs, etc.) from the original document and concatenating them into a shorter form. Therefore, most of recent works in this area are based on extraction [5]. Text summarization techniques can be of great value for other tasks. A variety of tasks in which text summarization system can be applied ranges from Personal digital assistants (PDAs) [2] to search engines [13].

A comprehensive survey of text summarization approaches can be found in [11]. Luhn [10] proposed a simple but effective approach that uses term frequencies and their related positions to weight sentences that are then extracted to form a summary. Subsequent works have demonstrated the success of Luhn's approach [2][9]. Edmunson [4] proposed the use of other features such as title words, sentence location, and bonus words to improve sentence extraction. Goldstein et al. [5] presented a sentence extraction technique that assigns weighted scores for both statistical and linguistic features in the sentence. Recently, Salton et al. [15] have developed a model for representing a text document by using undirected graphs. Their approach considers vertices as paragraphs and edges as the similarity between two paragraphs. The idea is that the most import paragraphs should be linked to many other paragraphs, which are likely to discuss topic covered in those paragraphs.

Statistical learning approaches have also been studied in text summarization problem. The first known supervised learning algorithm was proposed by Kupiec et al. [8]. Their approach estimates the probability that a sentence should be included in a summary given its feature values based on the independent assumption of Bayes' Rule. Other supervised learning algorithms have already been investigated. Chuang and Yang [3] studied several algorithms for extracting sentence segments, such as decision tree, naive Bayes classifier, and neural network. They also used rhetorical relations for representing features. However, one drawback of the supervised learning algorithms is that they require an annotated corpus to learn.

In this paper, we propose an alternative approach for extracting the most salient sentences from the original document to form a summary. The idea of our approach is to exploit both the local and the global properties of sentences. The local property can be considered as clusters of significant words within each sentence, while the global property can be though of as relations of all sentences in a document. These two properties can be combined and tuned for ranking and extracting sentences. We provide experimental evidence that our approach achieves acceptable performance comparable to a text summarizer (Microsoft Word summarizer). Furthermore, our algorithm does not require the external knowledge other than the document itself, and be able to summarize general text documents.

The remainder of this paper is structured as follows. Section 2 describes our approach in detail, including how to find clusters of significant words, how to discover relations of text spans, and an algorithm for combining these two approaches. In Section 3, we give the experimental setup and performance evaluations. Section 4 presents our experimental results compared with the other text summarizer. We conclude in Section 5 with some directions of future work.

2. Generating Summaries by Extraction

2.1. Finding Clusters of Significant Words

In this section, we first describe an approach for finding clusters of significant words in each text span to calculating the *local clustering score*. Our approach is reminiscent of Luhn's approach [10] but uses the other term weighting scheme instead of the term frequency. Luhn suggested that the frequency of a word occurrence in an article, as well as its relative position determines its significance in that article. More recent works have also employed Luhn's approach as a basis component for extracting summary sentences [2][9]. This approach performs well despite of its simplicity.

Let β be a subset of a continuous sequence of words in a sentence, $\{w_u, \ldots, w_v\}$. The subset β is called a cluster of significant words if the following rules are true.

- The first word w_u and the last word w_v in the sequence are significant words.
- Significant words are separated by not more than a predefined number of insignificant words.

For example, we can partition the continuous sequence of words in a sentence into clusters as shown in Figure 1. In this case, the sentence consists of twelve words. We use the boldface to indicate positions of significant words. Each cluster is enclosed with brackets. In this example, we define that a cluster of significant words is created whereby significant words are separated by not more than three insignificant words. Note that many clusters of significant words can be found in a sentence. The highest score of the clusters found in the sentence is selected to be the sentence score. Thus, the local clustering score for sentence s_i can be calculated as follows:

$$L_{s_i} = \text{argmax}_{\beta} \frac{ns(\beta, s_i)^2}{n(\beta, s_i)} \,, \tag{1}$$

where $ns(\beta, s_i)$ is the number of bracketed significant words, and $n(\beta, s_i)$ is the total number of bracketed words.

$$w_1 \left[\boldsymbol{w_2} \; w_3 \; \boldsymbol{w_4} \right] w_5 \; w_6 \; w_7 \; w_8 \left[\boldsymbol{w_9} \; w_{10} \; \boldsymbol{w_{11}} \; \boldsymbol{w_{12}} \right].$$

Figure 1. Clusters of significant words.

We can see that the first important step in this process is to identify significant words to mark positions for calculating the local clustering score. Our goal is to identify topical words, which are indicative of the topics underlying the document. It is well accepted in information retrieval community that words can be broadly classified into content-bearing and stopwords. According to Luhn's approach, the term frequency within the document is used to weight all the words. The other term weighting technique frequently used is TFIDF (Term Frequency Inverse Document Frequency) [14]. Unfortunately, this technique needs a corpus for computing IDF score, causing the genre-dependent problem for generic text summarization task.

In our work, we decide to use TLTF (Term Length Term Frequency) term weighting technique [1] for scoring words in the document instead of TFIDF. TLTF multiplies a monotonic function of the term length by a monotonic function of the term frequency. The basic idea of TLTF term weighting technique is based on the assumption that words which are used more frequently tend to be shorter. Such words are not strongly indicative of the topics underlying in the document, e.g. stopwords. In contrast, words which are used less frequently tend to be longer. One significant benefit of using TLTF term weighting technique for our task is that it does not require any external resources, only using the information within the document.

2.2. Discovering Relations of Text Spans

We then move on to describe an approach for discovering relations of text spans. Given a document D, we can represent it by a undirected graph $G = (V, E)$, where $V = \{s_1, \ldots, s_m\}$ denotes the set of text spans in that document. Here a text span is considered to be a sentence. An edge (s_i, s_j) is in E if and only if the similarity between paragraphs s_i and s_j is above a certain threshold α. A widely used measure of the similarity is the cosine similarity. A sentence s_i consists of a set of words $\{w_{s_{i,1}}, w_{s_{i,2}}, \ldots, w_{s_{i,t}}\}$, where $w_{s_{i,k}}$ is a word in position k of sentence s_i. The cosine similarity between two sentences can be calculated by the following formula:

$$sim(s_i, s_j) = \frac{\sum_{k=1}^{t} w_{s_{i,k}} w_{s_{j,k}}}{\sqrt{\sum_{k=1}^{t} w_{s_{i,k}}^2 \sum_{k=1}^{t} w_{s_{j,k}}^2}} \,. \tag{2}$$

The graph G is called the text relationship map of D [15]. Let d_{s_i} denotes the degree of node s_i. We then refer to

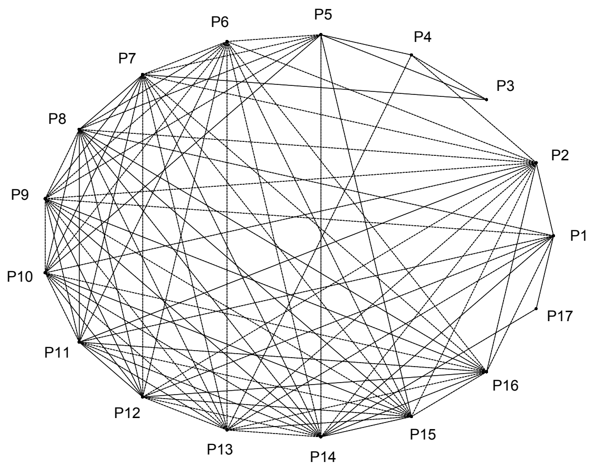

Figure 2. Text relationship map of an article entitled "Tracing Ancestry with MtDNA" using α = 0.10.

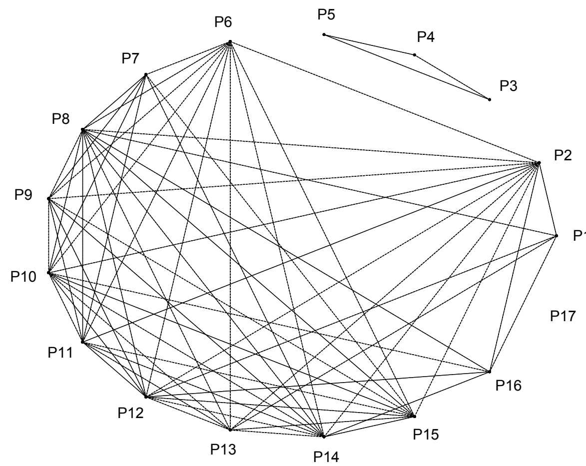

Figure 3. Text relationship map of the same article, but using α = 0.20.

d_{s_i} as the *global connectivity score*. Generating a summary for a given document can be processed by sorting all the nodes with d_{s_i} in decreasing order, and then simply extracting n top-scoring nodes, where n is the targeted number of sentences in the summary.

This idea is based on Salton et al.'s approach that performed extraction at the paragraph level. They suggested that since a highly bushy node is linked to a number of other nodes, it has an overlapping vocabulary with several paragraphs and is likely to discuss topics covered in many other paragraphs. Consequently, such nodes are good candidates for extraction. They then used a global bushy path that is constructed out of n most bushy nodes to form the summary. Their experimental results on encyclopedia articles showed that this approach gave reasonable results.

However, when we directly applied this approach for extracting summaries from a collection of medium-sized documents, such as online articles or newspapers, we found that using only the global connectivity score is inadequate to measure the informativeness of text spans in some case. In order to describe this situation, we consider an example of a text relationship map in Figure 2. The map is constructed from an online article entitled "Tracing Ancestry with MtDNA".[1] In this case, a text span is considered to be a paragraph as performed in [15]. The similarity threshold α is 0.1. Thus, edges with similarity less than 0.1 do not appear on the map. Nodes P2, P7, P8, P9 and P13 obtain the maximum global connectivity score at 13. However, the global connectivity score of nodes P6, P10, P11, P12, P15, and P16 is at 12, which is slightly different. When we increase the threshold α = 0.2, we obtain a text relationship map as shown in Figure 3. Although node P8 now only achieves the maximum global connectivity score at 12, nodes P2, P10, P12, and P14 still get the same score at 11.

From above example, it is hard to determine that node P8 is more relevant than nodes such as P2 or P10, since their scores are only different at 1 point. Our preliminary experiments with many other documents lead to the suggestion that the global score of nodes in the text relation map tends to be slightly different on some document lengths. Furthermore, when nodes are used to represent text spans at the sentence level, the text relationship map is produced with the high redundancy in scores. Given a compression rate (ratio of summary length to source length), if we immediately extract these nodes of sentences, many sentences with the same score are also included in the summary.

2.3. Combining Local and Global Properties

In this section, we present an algorithm that takes advantage of both the local and the global properties of sentences for generating extractive summaries. From previous sections, we describe two different approaches that can be used to extract summary sentences. However, these extraction schemes are based on different views and concepts. The local clustering score only capture the content of information within sentences, while the global connectivity score consider mainly the structural aspect of the document to evaluate the informativeness of sentences. This leads to our motivation for unifying good aspects of these two properties. We can consider the local clustering score as the local property of sentences, and the global connectivity score as the global property. Here we propose an algorithm that combines the

[1] The article is available at: `http://www.pbs.org/wgbh/nova/neanderthals/mtdna.html`

local clustering score with the global connectivity score to get a single measure of the informativeness of sentences, which can be tuned according to the relative importance of properties.

Our algorithm proceeds as follows. Given a document, we start by eliminating stopwords and extracting all unique words in the document. These unique words are used to be the document vocabulary. Thus, we can represent a sentence s_i as a vector. We then compute the similarity between all the sentence vectors using equation (2), and eliminate edges with similarities less than a threshold in order to build the text relationship map. This process automatically yields the global connectivity score. Next, we weight each word in the document vocabulary using TLTF term weighting. All the words are sorted by their TLTF scores, and top r words are selected to be significant words. We mark positions of significant words in each sentence to calculate the local clustering score. After obtaining both scores, we can compute, for each sentence s_i, the combination score by using the following ranking function:

$$F(s_i) = \lambda G' + (1 - \lambda) L' \,, \tag{3}$$

where G' is the normalized global connectivity score, and L' is the normalized local clustering score. The normalized global connectivity score G' can be calculated as follows:

$$G' = \frac{d_{s_i}}{d_{max}} \,, \tag{4}$$

where d_{max} is the degree of the node that has the maximum edges using for normalization, resulting the score in the range of $[0, 1]$. Using equation (1), L' is given by:

$$L' = \frac{L_{s_i}}{L_{max}} \,, \tag{5}$$

where L_{max} is the maximum local clustering score using for normalization. Similarly, it results this score in the range of $[0, 1]$. The parameter λ is varied depending on the relative importance of the components G' and L'. As a result, we can rank all the sentences according to their combination scores in decreasing order. Finally, we extract n top-scoring sentences corresponding to the compression rate, and rearrange them in chronological order to form the output summary.

3. Experiments

We now describe in detail the data set, experimental setup, and performance evaluations. The typical approach for testing a summarization system is to create an "ideal" summary, either by professional abstractors or merging summaries provided by multiple human subjects using methods such as majority opinion, union, or intersection [7]. This approach is known as intrinsic method. However, identifying important units in documents by humans is a time-consuming process. There have been several works presenting algorithms for automatic producing gold standard sentences [1] [12]. These gold standard sentences called *extracts* can be used for evaluating the summarization systems.

3.1. Data set and Experimental setup

In our experiments, we selected 10 articles taken from a corpus, the Ziff-Davis. This data set is a collection of newspaper articles announcing computer products, marking important units in each document by Marcu [12]. The annotation scheme is based on aligning clauses in a handwritten abstract with clauses in that text. To determine these clauses, the rhetorical structure theory is applied. The idea is to delete the clauses from the text and also preserve the correlation between the abstract and the text based on cosine similarity measure. This process converges when it can no longer delete the clauses without decreasing the similarity of the current set of extracts with the abstract. It finally cleans up the generated extracts using the rhetorical status of the clauses. Due to this annotation scheme, important units in a document are marked at either the clause level or the sentence level as shown below.

> On Microsoft SQL Server <I> , if a new form is created, DataEase will create the table on the back end according the fields defined in the form. </I>

> <I> PC Week Labs found no facility to call externally written program routines. </I>

After removing unnecessary markers and tags, we used an individual sentence that contains the important unit as an extract to evaluate the performance of the summarization system. In order to compare our algorithm with the other text summarizer, we chose the Microsoft Word summarizer of Microsoft Office 2000 as the baseline approach.

3.2. Performance Evaluations

We evaluate results of summarization by using the standard recall, precision, and F_1. Let J be the number of extracts in the summary, K be the number of selected sentences in the summary, and M be the number of extracts in the test document. We then refer to precision (P) of the algorithm as the fraction between the number of extracts in the summary and the number of selected sentences in the

Text	MS Word summarizer			Local			Global			Local+Global		
	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
T.1	0.700	0.778	0.737	0.500	0.667	0.571	0.364	0.444	0.400	0.556	0.556	0.556
T.2	0.250	0.300	0.273	0.455	0.500	0.476	0.400	0.400	0.400	0.455	0.500	0.476
T.3	0.214	0.231	0.222	0.375	0.462	0.414	0.429	0.462	0.444	0.357	0.385	0.370
T.4	0.125	0.111	0.118	0.357	0.556	0.435	0.500	0.556	0.526	0.500	0.444	0.471
T.5	0.600	0.300	0.400	0.571	0.400	0.471	0.714	0.500	0.588	0.833	0.500	0.625
T.6	0.500	0.273	0.353	0.333	0.273	0.300	0.300	0.273	0.286	0.429	0.273	0.333
T.7	0.444	0.308	0.364	0.375	0.231	0.286	0.556	0.385	0.455	0.500	0.308	0.381
T.8	0.375	0.250	0.300	0.500	0.417	0.455	0.600	0.500	0.545	0.714	0.417	0.526
T.9	0.250	0.200	0.222	0.667	0.400	0.500	0.750	0.600	0.667	1.000	0.600	0.750
T.10	0.429	0.333	0.375	0.571	0.444	0.500	0.429	0.333	0.375	0.667	0.444	0.533
Avg	0.389	0.308	0.336	0.470	0.435	0.441	0.504	0.445	0.469	0.601	0.443	0.502

Table 1. Comparisons of precision (P), recall (R), and F_1 using different methods.

summary:

$$P = \frac{J}{K}\,, \tag{6}$$

recall (R) as the fraction between the number of extracts in the summary and the number of extracts in the test document:

$$R = \frac{J}{M}\,. \tag{7}$$

Finally, F_1, a combination of recall and precision, can be calculated as follows:

$$F_1 = \frac{2 \cdot P \cdot R}{P + R}\,. \tag{8}$$

4. Results and Discussions

In this section, we provide experimental evidence that our algorithm gives acceptable performance. The compression rate of sentence extraction to form a summary is 20%. This rate yields the number of extracts in the summary relative to the number of actual extracts in a given test document. The threshold α of the cosine similarity is 0.2. For the distance between significant words in a cluster, we set that significant words are separated by not more than three insignificant words. We also examined our algorithm by using an individual property, and combining both the local and the global properties with λ = 0.5.

Table 1 shows a summary of precision, recall, and F_1 for each approach. Local and Global columns are produced by simply varying λ = 0 and λ = 1, respectively. The Microsoft Word summarizer reaches an average of 0.39 precision, 0.31 recall, and 0.34 F_1. Our algorithm using only the local property achieves an average of 0.47 precision, 0.435 recall, and 0.44 F_1, while using only the global property achieves an average of 0.50 precision, 0.445 recall, and 0.47 F_1. Interestingly, the algorithm gives the best results when it uses both properties. Combining both properties yield results of 0.60 precision, 0.44 recall, and 0.50 F_1.

We also see that average recall values of our algorithm are slightly different, but average precision values significantly improve using the combination of properties. Since using only the local or the global property tends to select more sentences from the document, it increases the chance that the selected sentences will be matched with the target extracts. On the other hand, it also selects irrelevant sentences to be included in the summary, so precision can decrease. However, these experiments involve just one data set. Further experiments are needed to determine the performance of our algorithm. Figure 4 illustrates an example of key words and summary sentences extracted from the document ZF109-628-509 on the Ziff-Davis corpus. The implementation of our algorithm is available for user testing at `http://mickey.sci.ku.ac.th/~TextSumm/index.html`.

5. Conclusions and Future Work

In this paper, we have presented an alternative approach for extracting summary sentences from the original document to form a summary. Our approach takes advantage of both the local and the global properties of sentences. The algorithm that combines these two properties for ranking and extracting sentences is given. Experimental results show that our algorithm is very promising. Furthermore, the algorithm does not require the external knowledge other than the document itself, and be able to summarize general text documents.

In future work, we intend to fully conduct experiments with an Asian language documents, Thai. It is a challenge to summarize these documents, since they are known to be one of agglutinative languages similar to Chinese or Japanese, which means that there are no boundaries between adjoin-

Key words:

fastcache-sx, microway, performance, expansion card, pc week labs, computer, floating-point, computers, installation, original, programs

Summary result at 20%:

The MicroWay FastCache-SX operates on the same principle as the other two evaluated boards: Swap the 286 processor for a 386SX processor, and get 386 capabilities and improved computer performance.

One difference between the FastCache-SX and the Cumulus and 3Est boards is that MicroWay uses a standard expansion card to mount its circuitry.

MicroWay makes very good use of the additional space provided by the three-quarter-length expansion card.

MicroWay simplifies the FastCache-SX installation procedure by providing special tools to extract the old 286 chip.

With the cache turned on, the FastCache-SX delivers the performance expected from a 20MHz 386SX-based computer. During PC Week Labs' tests, the MicroWay FastCache-SX performed at least twice as fast as the original AT.

The FastCache-SX's time, however, is half that of the times turned in by the other two reviewed products.

Figure 4. An example of key words and summary sentences extracted from the document ZF109-628-509 at 20%.

ing words, and also no explicit sentences boundaries in a document. We plan to make a corpus of Thai documents with extracts to experimentally evaluate our algorithm.

6. Acknowledgment

We are indebted to Daniel Marcu for providing us the Ziff-Davis data set.

References

[1] M. Banko, V. Mittal, M. Kantrowitz, and J. Goldstein. Generating extraction-based summaries from hand-written summaries by aligning text spans. *Proceedings of PACLING '99*, 1999.

[2] O. Buyukkokten, H. Garcia-Molina, and A. Paepcke. Seeing the whole in parts: Text summarization for web browsing on handheld devices. *WWW10*, May 2-5 2001.

[3] W. T. Chuang and J. Yang. Extracting sentence segments for text summarization: A machine learning approach. *Proceedings of the 23rd ACM SIGIR*, pages 152–159, 2000.

[4] H. P. Edmundson. New methods in automatic extraction. *Journal of the ACM*, 16(2):264–285, 1969.

[5] J. Goldstein, M. Kantrowitz, V. Mittal, and J. Carbonell. Summarizing text documents: Sentence selection and evaluation metrics. *Proceedings of the 22nd ACM SIGIR*, pages 121–128, 1999.

[6] U. Hahn and I. Mani. The challenges of automatic summarization. *IEEE Computer*, 33(11):29–35, 2000.

[7] H. Jing, R. Barzilay, K. McKeown, and M. Elhadad. Summarization evaluation methods: Experiments and analysis. *AAAI Intelligent Text Summarization Workshop*, pages 60–68, 1998.

[8] J. Kupiec, J. Pedersen, and F. Chen. A trainable document summarizer. *Proceedings of the 18th ACM SIGIR*, pages 68–73, 1995.

[9] A. M. Lam-Adesina and G. J. F. Jones. Applying summarization techniques for term selection in relevance feedback. *Proceedings of the 24th ACM SIGIR*, pages 1–9, 2001.

[10] H. P. Luhn. The automatic creation of literature abstracts. *IBM Journal of Research and Development*, pages 159–165, 1959.

[11] I. Mani and M. T. Maybury. *Advances in actomatic text summarization.* MIT Press, 1999.

[12] D. Marcu. The automatic construction of large-scale corpora for summarization research. *Proceedings of the 22nd ACM SIGIR*, pages 137–144, 1999.

[13] D. R. Radev and W. Fan. Automatic summarization of search engine hit lists. *In Proceedings, ACL Workshop on Recent Advances in NLP and IR. MIT Press*, 2000.

[14] G. Salton and C. Buckley. Term weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513–523, 1988.

[15] G. Salton, A. Singhal, M. Mitra, and C. Buckley. Automatic text structuring and summarization. *In I. Mani, M. Maybury (Eds.), Advances in automatic text summarization. MIT Press*, 1999.