Information Organization System for Duplicated Information Sources

Takaharu Takeda
Graduate University for Advanced Studies
2-1-2 Hitotsubashi, Chiyoda, Tokyo 101-8430, Japan

Atsuhiro Takasu
National Institute of Informatics
2-1-2 Hitotsubashi, Chiyoda, Tokyo 101-8430, Japan

ABSTRACT

Web becomes an important and global information source. Because it becomes very large, we need a system that helps users to find information from them. Information organization is one way to support users' information finding task. Since the Web contains similar or same information, we propose an information organization system that is designed for those duplicated information sources. The proposed system has functions of crawling, feature extraction, clustering/classification, summarization, and novel information extraction and support users to find necessary information efficiently. This paper evaluates the proposed system using evaluation corpus on text summarization task.

KEYWORDS

Information Filtering, Multiple Text Summarization, Suffix Array

1. INTRODUCTION

As the growth of the Internet, large amount of information is accumulated and disseminated. The Web becomes the global information source, and we can obtain any information from them. However, it is too large and is not organized well, so we sometimes have difficulty to find necessary information. So we need systems that support us to handle the accumulated and disseminated information. Information retrieval techniques have been developed (Baeza-Yates & Ribeiro-Neto 1999) and search engines become an indispensable tool for retrieving information from large amount of web pages. Duplication is one feature of recently disseminated information. Digital information can be easily copied and reassembled. For example, news articles often contain paragraphs and sentences that are same to the previous articles because they are often composed by adding updated information to current article. Spam mails and spam blogs (splog) are other examples that contain duplicated information. Another feature of recent information is its dynamicity. Many types of pages on the Internet are frequently updated and new pages are added.

Several techniques have been developed to handle these features. Text categorization and clustering is a basic tool to extract information from unorganized information source. However, they are usually designed for a static set of documents, and they don't handle dynamically updated information. Topic detection and tracking (TDT) systems detect clusters from document streams (Allan 2002). With them users can keep track of news events that interest them. TDT systems detect the first document describing a new topic and track the following documents describing the same topic. Yang et al. applied a clustering technique for static documents (Yang et al. 2000). Document creation time is important information for handling document stream. Frants and McCarley discuss the effectiveness of the time information for topic tracking (Frants & MCarley 2000). Brants et al. proposed how to use the time information for topic tracking (Brants et al. 2003).

When the cluster becomes large, we need concise description of topics. Text summarization (TS) can be used to make digest of documents describing same topics. Since Web pages contain duplicated information, multiple text summarizations are effective to reduce the amount of text describing the same topic. TS systems usually choose important sentences from documents, reduce the duplicated description, and make summaries from the remaining important sentences. Several multiple text summarization methods are proposed.

Centroid-based summarization (CBS) (Dragomair et al. 2000) uses the centroids of the clusters of news articles produced by standard single-pass clustering systems (CIDR) (Hirao et al. 2004) in order to extract sentences central to the topic. R. Barzilay et al. proposed a method that generates a "concise summary" by identifying and synthesizing the similar elements across related texts from a set of documents (Barzilay et al. 99). This system first determines how to combine propositions into a single sentence, and then it combines each set of propositions into a sentence, maps them from concepts to words, and builds a syntactic structure. The Columbia summarizer (McKeown et al. 2001) uses machine learning and statistical techniques to identify similar sentences across the inputted articles. Novelty detection is another related technique that finds novel parts in documents. Maximal marginal relevance (Carbonell & Goldstein 1998) is a widely used approach for information retrieval and applied to the novelty detection. It ranks documents according to a combined criterion of query relevance and information novelty within a document. Then, it extracts the novel sentences and creates a summary from them.

Although TDT and TS have been studied fairly independently, both technologies are useful to construct information organization system for duplicated, unorganized and dynamic information sources. We are developing a system for handling such information. Document and sentence similarity measurement is the basic part for both TDT and TS. Usually they are regarded as a bag of words and document similarity is measured as a similarity between bags of words. However, documents in duplicated information source often contain phrases and sentences that appear in multiple documents. We use them to integrate topic detection, text summarization, and organization of inside topics. In this paper we propose our system focusing on text summarization technique.

The rest of this paper is organized as follows. In section 2, we show overview of our information organization system for duplicated, unorganized, and dynamic information sources. Section 3 describes core components of our filtering system focussing on feature extraction. Section 4 presents the experimental results concerning the accuracy of the proposed method. Finally, Section 5 concludes this paper and addresses some future research directions.

2. INFORMATION ORGANIZATION SYSTEM

2.1 System Overview

This section overviews our information organization system. As described before, our system helps users to find information from duplicated, unorganized, and dynamic information source. Currently, our system handles news articles (Takeda & Takasu 2007; Takeda and Takasu 2008b) and blogs (Takeda and Takasu 2008a).

As for news articles, our system detects new topics from online news and manages document clusters corresponding to news topics. These tasks are same as TDT project (Allan 2002). For each cluster of news articles, our system further makes summary, and then extracts novel information from news article in the cluster. Novel information is a set of phrases and sentences that describe the information that first appears in the series of news articles. For example, if the topic is an earthquake, the novel information can be the first news about the magnitude of the earthquake in the series of news articles about the earthquake. By this system, users can choose their interested topics and grasp the topic by reading summary and novel phrases and sentences.

As for blogs, our system filters out the spam blogs. A study about the blogosphere reports that about 22\% of blogs is spam (Takeda and Takasu 2008a). Therefore spam blog (splog) filtering (Kolari et al. 2006; Lin et al. 2007, Narisawa et al. 2007) is an important task for utilizing the blogosphere. Our system gathers blogs, and then, makes two clusters, i.e., blogs and splogs.

2.2 System Components

Figure 1 shows the overview of our information organization system. As shown in the figure, the system consists of a crawler, feature extractor, cluster constructor, summarizer, and novel phrase detector.

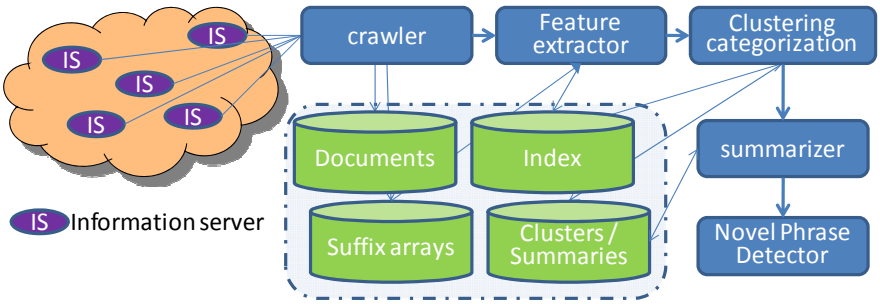

Figure 1. Overview of proposed information organization system.

.

The crawler gathers documents from information servers on the Internet. For news article, the information servers are online news site whereas it is blog service provider for blogs. The crawler polls servers and periodically obtains information concerning any uploaded information in an RSS format. It extracts the contents of news from the html source by removing tags and web advertisements using manually coded rules. In particular, it extracts strings inside the tags of specified ids, such as “main body” or “content”, with an HTML parser. The processed documents are stored in the system database.

This system is designed to filter duplicated information. We assume that the objective information sources contain duplicated phrases and sentences. As for online news article, updated news articles are often made by copying the previous article in the same topic and adding new paragraphs describing updated part of news. Therefore, copied phrases and sentences are important features in the later processes of filtering. As for the splog filtering, splogs are often generated by copying phrases and sentences from other blogs and Web pages. Therefore, copied phrases and sentences are also important features. Because we need to extract them from large amount of documents, efficient and scalable extraction algorithm is required. Our system uses suffix array (Ukkonen 1990) for efficient processing.

Using the extracted features, this system makes clusters. For news articles, each cluster corresponds to a topic described in news articles whereas it corresponds to blogs or splogs for the splog filtering problem. In this system, we define a document similarity depending on the characteristics of the task. For news articles, we use a standard cosine similarity that is often used in information retrieval. As for the splog filtering, we use a metric describing how likely word sequences in the objective blog are copied from other documents.

For each cluster, the summarizer makes summary where it first detects important phrases and sentences. Then, it concatenates the extracted important sentences to make summary. At the same time, the system extracts novel phrases and sentences from the remaining important sentences.

3. INFORMATION ORGANIZATION METHOD

This section describes our information organization method focussing on feature extraction.

3.1 Features for Duplicated Information Source

For information organization, it is important to calculate document similarity that is suitable for objective task. We often adopt the bag-of-word model and calculate document similarity using term and document frequencies of words included in documents. However, documents in duplicated information sources often share longer word sequences. For example, since splogs are often generated by copying phases and sentences, they appear in multiple blogs in the blogosphere. As for news articles, the updated news contains copies of phrases and sentences in the original news. These word sequences are useful for measuring document similarity. Instead of words, we use those word sequences as features of documents. We handle word sequences whose length are longer than predefined *length parameter l*.

In news articles, word sequences describing the same information may be described differently. For example, name of people may be described by his/her title such as Dr. X in one article, whereas he/she may be described by Mr./Ms. X in another article. To handle this descrepancy, we first define the equivalence of

word sequences based on the length and similarity of word sequences. For a pair $\boldsymbol{s}_1$ and $\boldsymbol{s}_2$ of word sequences, let $d(\boldsymbol{s}_1,\boldsymbol{s}_2)$ denote the distance between $\boldsymbol{s}_1$ and $\boldsymbol{s}_2$. The function d should be defined depending on the task. As for the splog filtering, we use the exact match as the distance, i.e., $d(\boldsymbol{s}_1,\boldsymbol{s}_2)$ is 0 if $\boldsymbol{s}_1$ is equal to $\boldsymbol{s}_2$, otherwise it is 1.

On the other hand, we use an edit distance whose editing cost is defined in the following way. For a word w, let $idf(w)$ denote the following inverse document frequency (IDF).

$$idf(w) = \log\left(\frac{|D|}{df(w)+1}\right), \tag{1}$$

where D denotes the set of news article and $df(w)$ denotes the number of document including the word w. We use $idf(w)$ as the importance of the word. The costs of insertion $C_i(w)$ and deletion $C_d(w)$ are defined as

$$C_i(w) = C_d(w) \equiv idf(w), \tag{2}$$

whereas the cost of substitution is defined as

$$c_s(w_1, w_2) \equiv \begin{cases} \dfrac{c^2{}_{\max} + c^2{}_{\min}}{c_{\max}} & w_1 \neq w_2 \\ -2idf(w_1) & w_1 = w_2 \end{cases}, \tag{3}$$

where

$$\begin{aligned} c_{\min} &\equiv \min(idf(w_1), idf(w_2)) \\ c_{\max} &= \max(idf(w_1), idf(w_2)) \end{aligned}. \tag{4}$$

We derived these empirical costs through preliminary experiments. Intuitively, Eq. (2) means that we need higher cost to insert or delete words whose IDF is high. On the other hand, Eq. (3) means that substitution cost is almost $idf(w_1) + idf(w_2)$ if IDFs of words are almost same. Otherwise the substitution cost is almost $C_{\max}$. When $w_1 = w_2$, it reduces the cost proportional to its IDF.

The distance $d(\boldsymbol{s}_1,\boldsymbol{s}_2)$ is defined as the weighted edit distance (Kurtz 1996) according to the costs (2) and (3), i.e., the minimum cost for converting a word sequence $\mathbf{s}_1$ to $\mathbf{s}_2$ by edit operations.

3.2 Phrase Index

We regard that word sequences whose distance is less than predefined parameter t describe the same information. Let us consider a set P of word sequences such that:

- For any word sequence $\boldsymbol{s}$ in P, for some $\boldsymbol{s}'$ in P, inequality $d(\boldsymbol{s},\boldsymbol{s}') < t$ holds, and
- P is maximal.

We refer to the set of word sequences satisfying these conditions as a *phrase index*. To enumerate phrase indices, we need to enumerate all word sequences and make clusters of them according to the distance function $d(\boldsymbol{s}, \boldsymbol{s}')$. However, computational cost of the enumeration is high, so we developed an efficient approximate algorithm (Takeda & Takasu 2008b). Due to page length limitation, we omit the algorithm.

For each phrase index P, we use a weight of P that is a form of function $W(|P|)$, i.e., the function of the number of word sequences included in P. The weight function should be defined according to the task. As for the splog filtering, we used the following weight function (Takeda & Takasu 2008a):

$$W(|P|) \equiv \log|D| - \log|P|. \tag{5}$$

In splog filtering, the weight represents how likely the word sequence is copied from other documents. Eq. (2) is a sort of IDF. We can use other weight functions. For example, Narisawa et al. (Narisawa et al. 2007) assumed that the frequency of word sequence satisfies Zipf's law and introduced a metric representing how the frequency of word sequence is distant from the Zipf's law. Our system regards that a document is splog if the sum of weights of word sequences included in the document is larger than predefined parameter (Takeda & Takasu 2008a).

As for the summarization, we assume that important phrase indices are repeated frequently in a series of news article describing the same topic. According to this assumption, we use the following weight function:

$$W(|P|) \equiv |P|, \tag{6}$$

that is, we simply use the number of word sequences included in the phrase index.

3.3 Summary Generation

This section describes how to make summary from documents. We select sentences according to the weight of phrase index. For a phrase index P, let $S(P)$ denote the set of sentences that include one of word sequences in P. Then, we extract the representative sentence of $S(P)$ that has the maximum value of sum of IDFs of words included in the sentence, i.e.,

$$R(P) \equiv \arg\max_{w_1 \cdots w_l \in S(P)} \sum_{i=1}^{l} idf(w_i), \tag{7}$$

where $idf(w)$ is defined by Eq. (1).

Let $P_1, P_2, ..., P_n$ be the list of phrase indices ordered by their weights. It is usually required to generate a summary whose length is shorter than the specified length. We choose representative sentences from the phrase indices P_1, P_2, ..., and concatenate them until the summary length exceeds the specified length.

4. EXPERIMENTAL RESULTS

4.1 Data Set

There are many studies about text summarization. To compare the proposed method to those studies and clarify the characteristics of the proposed method, we used the NTCIR4-TSC3 (Hirano04) corpus which is constructed for evaluating summarization systems for multiple documents. The corpus consists of articles from the Mainichi and Yomiuri newspapers (in Japanese) published between 1998 and 1999. The Corpus consists of 30 clusters of news articles. Each cluster corresponds to single event in the news articles. Events are typical topic in news articles. They are also used as topics in the topic detection and tracking task [allan02]. Average number of articles in a cluster is about 10. Hereafter we refer to the cluster as a topic. In the corpus, a set $\{m_1, m_2, \cdots, m_n\}$ of important sentences is manually assigned to each topic for both short and long summaries. The ideal summary should contain the information described by important sentences. Because important sentences may be described in various sentences, for each important sentence m_i, the corpus also provides a set $A_i \equiv \{A_{i,1}, A_{i,2}, \cdots, A_{i,l}\}$ of sentences in the news articles where *equivalent sentence* $A_{i,j}$ is a set of sentences describing the information equivalent to m_i. In addition, the corpus provides manually written short and long summaries.

4.2 Evaluation Metrics

Several kinds of metrics are used to describe the quality of summary. In NTCIR4-TSC3, two kinds of evaluation metrics were prepared. One is subjective, i.e., the scores were given by humans who read the generated summaries. The other is objective metrics called precision and coverage that can be calculated automatically by comparing the summary generated by system with the important sentences prepared manually. In this experiment we used these two objective metrics to compare the quality of summaries.

Precision is the ratio of how many sentences in the summary generated by system are included in the manually prepared important sentences (Hirano04). Let h be the minimum number of sentences required for making a summary containing all information, and m be the number of important sentences included in the summary generated by system. Then, the precision is defined as h/m

Coverage is an evaluation metric for measuring how close the system output is to the abstract taking into account the redundancy of the summary generated by system. For each important sentence m_i, it is defined as the ratio that denote how many corresponding sentences are included in a generated summary E. Formally, for a set A_i of equivalent sentences for m_i, the coverage of the important sentence m_i is defined as

$$e(i) \equiv \max_{1 \le j \le i} \frac{\left| A_{i,j} \cap E \right|}{\left| A_{i,j} \right|}. \tag{9}$$

Note that *e(i)* is 1 when any $A_{i,j}$ is completely included in the summary E. Using this ratio, the coverage is defined as

$$c(E) = \frac{\sum_{i=1}^{n} e(i)}{n}. \quad (10)$$

As described before, an important sentence m_i can be described by a set of sentences in documents in multiple ways. *Redundancy of important sentence* shows the redundancy of the generated summary. For an important sentence m_i, Let L_i denote the set of sentences related to m_i, i.e.,

$$L_i = \bigcup_j A_{i,j}. \quad (11)$$

Furthermore, let A_i^* denote the minimum set of sentences that covers the important sentence m_i, i.e.,

$$A_i^* = \arg\min_{A \in A_i, A_i \subseteq E} |A|. \quad (12)$$

Then, the redundancy of important sentence is defined as

$$r(E) = \frac{\sum_{i=1}^{n} |E \cap L_i| - |A_i^*|}{n}. \quad (13)$$

As shown in this equation, the redundancy of important sentence is the average number of redundant sentences in the generated summary *E*.

4.2 Performance Evaluation

In this experiment we compared the proposed method with other summarization systems participating in NTCIR4-TSC3. Important information is often described in the first part of document in news article. The baseline summarization method is a summarizer that takes the first part of the news article as important sentences. This method is referred to as "LEAD" in this paper. It picks a sentence one by one from the head of article in order of time for Multi-document summarization (TSC3). We also measured performance when we use simple tfidf term weighting scheme without feature phrases (TFIDF).

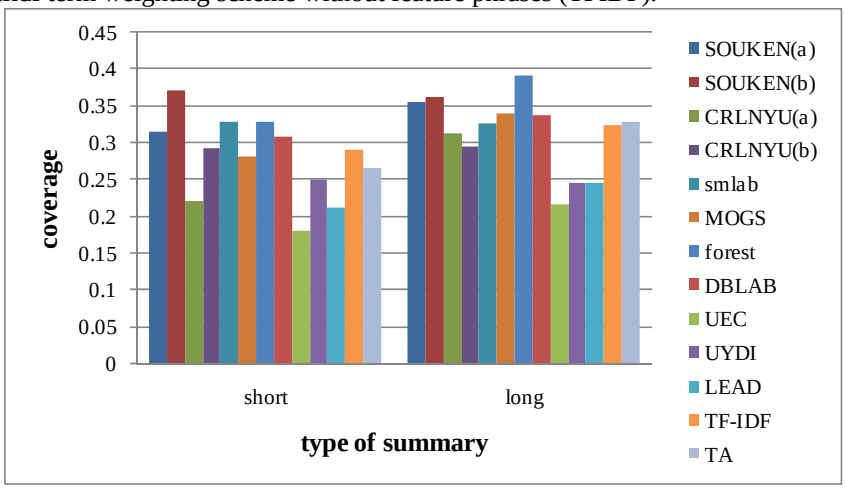

Figure 2. Coverage of summarization systems.

The second groups of compared systems are those that participated in TSC3. There were 10 systems that are referred to as "SOKEN a", "SOKEN b", "CRLNYU a", "CRLNYU b", "smlab" , "MOGS", "forest",

"DBLAB", "UEC" and "UYDI". Many of them used NLP techniques such as stemming, stop word removing or event modeling.

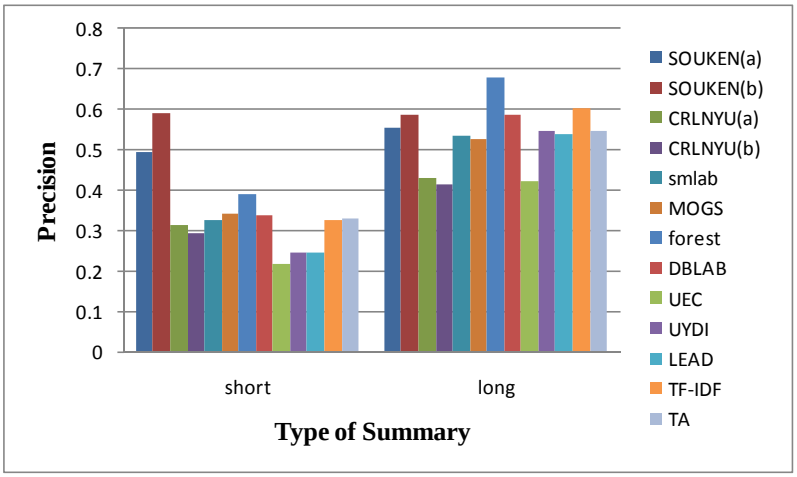

Figure 3. Precision of summarization systems.

Figure 2 and Figure 3 show the result of summarization in terms of coverage and precision, respectively. The results of the second group, i.e., systems participating in the TSC3, are drawn from the article [hirano04]. Figure 4 shows the result of summarization in terms of redundancy of important sentences. This metric was not used at TSC3, we got the generated summaries from three teams participating in TSC3 and measured the performance. Note that high precision and coverage mean better summarization whereas low redundancy of important sentences means better summarization.

First, the proposed method (TA) outperforms the baseline method LEAD.

Second, compared with the systems participating in the TSC3, however, the proposed method could not achieve top performance. Especially, the proposed method has lower performance for the short summary. This means that the proposed method is lower ranking ability of the important sentences compared to the second group systems. Many of the second group systems utilize language feature to extract the important sentences whereas the proposed method extracts important sentences based on the frequency of similar sentences. So the proposed method should incorporate more language feature for ranking clusters of important sentences.

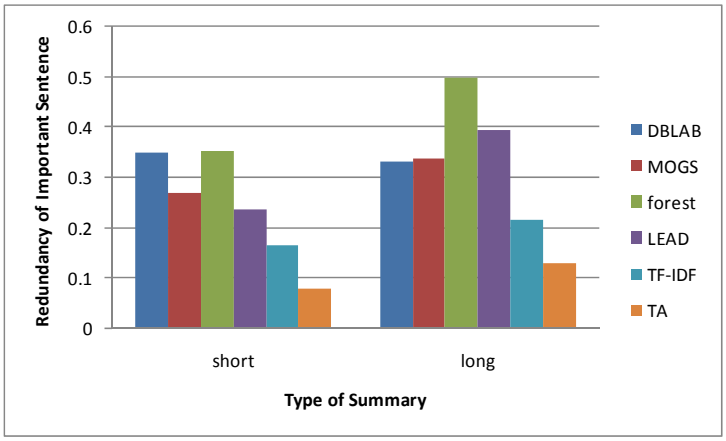

Figure 4. Redundancy of important sentences of summarization systems.

Third, the proposed method can generate summaries with less redundancy (see Figure 4). Many of the second group systems handle document as bag of words whereas the proposed method uses both word and phrase features. The latter feature seems to be effective to remove the redundancy.

5. CONCLUSION

This paper proposes an information filtering system. It utilizes duplicated phrases and sentences to make clusters and summaries. Experimental results show that the proposed method can significantly reduce the redundancy of summaries; however, it requires improvement in detecting important sentences.
We plan to improve the ability to detect important sentences by incorporating more language features such as named entities. The advantage of the proposed method is that it can generate concise summary. So we apply the proposed method to applications for mobile devices whose display area is very limited.

REFERENCES

Allan, J. 2002 *Topic Detection and Tracking: Event-based Information Organization.* Kluwer Academic Press.

Baeza-Yates, R. and Ribeiro-Neto, B. 1999. *Modern Information Retrieval.* ACM Press.

Barzilay, R. et al. 1999. Information fusion in the context of multi-document summarization. *Proceedings of ACL*.

Brants, T. et al. 2003. A System for New Event Detection. *Proceedings of International ACM SIGIR Conference on Research and Development in Information Retrieval.* pp. 330-337.

Carbonell, J. and Goldstein, J. 1998. The use of MMR, diversity-based reranking for reordering documents and producing summaries. Proceedings of *International ACM SIGIR Conference on Research and Development in Information Retrieval*, Melbourne, Australia.

Dragomir R. et al. 2000. Centroid-based summarization of multiple documents: sentence extraction, utility-based evaluation, and user studies. *Proceedings of the ANLP/NAACL Workshop on Automatic Summarization*.

Frants, M. and MCarley, J. S. 2000. Unsupervised and Supervised Clustering for Topic Tracking. *Proceedings of International ACM SIGIR Conference on Research and Development in Information Retrieval.* pp. 310-317.

Hatzivassiloglou, M. et al. 2001. SIMFINDER: A flexible clustering tool for summarization. *Proceedings of NAACL Workshop on Automatic Summarization.* pp. 41-49.

Hirao, T. et al. 2004. SIMFINDER: Text Summarization Challenge 3 –Text Summarization Evaluation at NTCIR Workshop4. *Working Notes of the Fourth NTCIR Workshop Meeting.*

Julie D. et al, 1994. CLUSTAL W: Improving the Sensitivity of Progressive Multiple Sequence Alignment through Sequence Weighting, Position-specific Gap Penalties and Weight Matrix Choice. *In Nucleic Acids Research*, Vol. 22,,pp 4673-4680.

Kolari, P. et al. 2006. Detecting Spam Blogs: A Machine Learning Approach. *Proceedings of 21st National Conference on Artificial Intelligence.*

Kurz, S. 1996. Approximate String Searching under Weighted Edit Distance. *Proceedings of South American Workshop on String Processing*, pp. 156-170.

Lin, Y.R. et al. 2007. Splog Detection Using Self-similarity Analysis on Blog Temporal Dynamics. *Proceedings of International Workshop on Adversarial Information Retrieval on the Web*. pp.1-8.

McKeown, K. R. et al. 2002. Tracking and summarizing news on a daily basis with Columbia's newsblaster. *Proceedings of the Human Language Technology Conference*.

Narisawa, K. et al. 2007. Efficient Computation of Substring Equivalence Classes with Suffix Arrays. *Proceedings of Annual Symposium on Combinatorial Pattern Matching*. pp.340-351.

Takeda, T. and Takasu, A.. 2007. UpdateNews: A News Clustering and Summarization System Using Efficient Text Processing. *Proceedings of Joint Conference on Digital Libraries.* pp.438-439.

Takeda, T. and Takasu, A.. 2008a. A Splog Filtering Method Based on String Copy Detection. *Proceedings of International Conference on the Applications of Digital Information and Web Technologies*. pp.543-548.

Takeda, T. and Takasu, A.. 2008b. News Aggregation System with Automatic Summarization Based on Local Multiple Alignment. *Proceedings of International Conference on Computers and Informatics. NLP65-73.*

Ukkonen, E. 1985. Finding Approximate Patterns in Strings. *Journal of Algorithms*. pp.132-137.

Yang, Y. et al. 2000. Improving Text Categorization Methods for Event Tracking. *Proceedings of International ACM SIGIR Conference on Research and Development in Information Retrieval.* pp. 65-72.