1. Introduction

Bi-directional communication, which enhances the effectiveness of education through question-and-answer and/or discussion sessions, is important in asynchronous e-Learning systems. Assignments are also important for e-Learning programs to motivate students and track their learning. However, assigned reports in e-Learning programs are submitted online, and it is very easy for students to plagiarize by pasting text from other reports or web sites.

This chapter introduces intelligent interaction support systems including an intelligent bulletin board system (iBBS) and a plagiarism-detecting assignment management system (pAMS). Both systems provide text-based intelligent functions that apply natural language processing and support smooth communication and a fair evaluation of e-Learning classes.

First, for iBBS, a function for the automated detection of important posting notifications (IPNs) is introduced. The IPN function performs information filtering; however, it is not feasible to estimate the statistics of word occurrences, which are required for conventional information filtering techniques, prior to opening a discussion. Nevertheless, since the subjects of discussion in e-Learning classes are always related to the class topic, the ontology is consistent throughout the discussion. A concept-based vector space model, which can virtually capture ontological information from a collection of documents, is therefore effective for the IPN function.

Second, pAMS, an assignment management system with plagiarism detection, is introduced. Students electronically submit their assigned reports using pAMS and the system stores the submitted documents. The system then compares each of submitted reports to other reports and web pages, and detects identical or similar parts of text in the reports. The system then makes the teacher aware of any potential plagiarism. This chapter describes the method for identifying the similar parts of two reports, even if one has been skillfully modified from the other. In addition, the method of retrieving original source materials is introduced. Finally, the evaluation results using a limited set of reports are described.
2. The Bulletin Board System with Intelligent Communication Support: iBBS

2.1 Requirements for iBBS
Inherently, a BBS for an e-Learning course cannot generate active discussions of the subject. If a large number of articles are posted on the BBS every day, reading every article becomes burdensome to both teachers and students, and may cause some students to drop out of the discussion. Therefore, it is desirable to reduce the workload for discussion participants. To realize this workload reduction, an intelligent communication support function, which is an important IPN function, is embedded in the BBS. We call this system iBBS, and it is achieved with natural language processing techniques.

2.2 The IPN Function in iBBS
The IPN function, illustrated in Figure 1, operates as follows.
1. A participant (student B in the figure) registers his/her words of interest (keywords) into the system.
2. Another participant (student A in the figure) posts an article that includes the content of interest to the student B.
3. The system notifies student B of the posting.

The IPN function must notify the posting of the article by including not only the keywords themselves but also similar words. For example, assume that student B registers an "urgent stop" as a keyword, and student A posts an article including the word "emergency stop." The system must be able to notify student B of this article because the article discusses "urgent stop" even though the poster uses word "emergency stop" to express it.

A message in the BBS has a title line expressing its subject; however, a novice user often uses an irrelevant title, e.g., "question" or "request." Thus, the system has to check the content of the posted message. Consequently, to apply the IPN function, the system must understand the meanings of words.

Although machine-readable dictionaries [3] are commonly used for this purpose, such dictionaries generally consist of words for daily use rather than technical terms used in specialized fields. The authors previously proposed the Concept-based Vector Space Model (CBVSM), which is able to capture the relationships between words used in target texts [14]. CBVSM provides a function that discerns the semantic similarity between words that appear in texts; this capability enables the system to process texts as if the technical terms are understood.

CBVSM and the procedure to construct the concept base are introduced below, and then an implementation of the IPN function of iBBS is described. A performance (accuracy) evaluation of the IPN function is also presented.
2. The Bulletin Board System with Intelligent Communication Support: iBBS

2.1 Requirements for iBBS
Inherently, a BBS for an e-Learning course cannot generate active discussions of the subject. If a large number of articles are posted on the BBS every day, reading every article becomes burdensome to both teachers and students, and may cause some students to drop out of the discussion. Therefore, it is desirable to reduce the workload for discussion participants. To realize this workload reduction, an intelligent communication support function, which is an important IPN function, is embedded in the BBS. We call this system iBBS, and it is achieved with natural language processing techniques.

2.2 The IPN Function in iBBS
The IPN function, illustrated in Figure 1, operates as follows.
1. A participant (student B in the figure) registers his/her words of interest (keywords) into the system.
2. Another participant (student A in the figure) posts an article that includes the content of interest to the student B.
3. The system notifies student B of the posting.

The IPN function must notify the posting of the article by including not only the keywords themselves but also similar words. For example, assume that student B registers an “urgent stop” as a keyword, and student A posts an article including the word “emergency stop.” The system must be able to notify student B of this article because the article discusses “urgent stop” even though the poster uses word “emergency stop” to express it.

A message in the BBS has a title line expressing its subject; however, a novice user often uses an irrelevant title, e.g., “question” or “request.” Thus, the system has to check the content of the posted message. Consequently, to apply the IPN function, the system must understand the meanings of words. Although machine-readable dictionaries [3] are commonly used for this purpose, such dictionaries generally consist of words for daily use rather than technical terms used in specialized fields. The authors previously proposed the Concept-based Vector Space Model (CBVSM), which is able to capture the relationships between words used in target texts [14]. CBVSM provides a function that discerns the semantic similarity between words that appear in texts; this capability enables the system to process texts as if the technical terms are understood.

CBVSM and the procedure to construct the concept base are introduced below, and then an implementation of the IPN function of iBBS is described. A performance (accuracy) evaluation of the IPN function is also presented.

2.3 Concept-based Vector Space Model

2.3.1 Concept Base and CBVSM
The purpose of CBVSM is the expressing documents and queries as vectors in a multi-dimensional space and calculating the relevance or similarity as a cosine coefficient between two centroid vectors in order to yield a vector space model.

With a basic similarity discernment scheme exploiting the vector space model, a vector of a document is mapped on a hyper-space, in which each word in a set of documents corresponds to an axis, such that the values along the document axes for the documents correspond to the term frequency-inverse document frequency (TFxIDF) values for the words in the documents [7]. This scheme assumes a vector space in which the words directly correspond to the axes. However, synonyms and/or co-occurrences of words are not considered.

Some improved methods for solving the above problem have been proposed. One is Latent Semantic Indexing (LSI) by Deerwester [2]. This method first counts the occurrences of words throughout the documents and then constructs a word frequency matrix. The rank of the matrix is then reduced using Singular Vector Decomposition (SVD), and the reduced-rank matrix is then used as the document vector space.

Another method is the co-occurrence based thesaurus by Schütze [8, 9]. This method obtains a word vector space based on word co-occurrences in close proximity in documents, whereas LSI creates a document vector space based on word frequencies throughout documents. The words that co-occur in a similar manner throughout documents are expected to be placed close to each other in the hyper-space. The authors refer to a database comprised of pairs of a word and its associated vectors as the “concept base”.

Fig. 1. Notification of Important Postings

2.3 Concept-based Vector Space Model

2.3.1 Concept Base and CBVSM
The purpose of CBVSM is the expressing documents and queries as vectors in a multi-dimensional space and calculating the relevance or similarity as a cosine coefficient between two centroid vectors in order to yield a vector space model.

With a basic similarity discernment scheme exploiting the vector space model, a vector of a document is mapped on a hyper-space, in which each word in a set of documents corresponds to an axis, such that the values along the document axes for the documents correspond to the term frequency-inverse document frequency (TFxIDF) values for the words in the documents [7]. This scheme assumes a vector space in which the words directly correspond to the axes. However, synonyms and/or co-occurrences of words are not considered.

Some improved methods for solving the above problem have been proposed. One is Latent Semantic Indexing (LSI) by Deerwester [2]. This method first counts the occurrences of words throughout the documents and then constructs a word frequency matrix. The rank of the matrix is then reduced using Singular Vector Decomposition (SVD), and the reduced-rank matrix is then used as the document vector space.

Another method is the co-occurrence based thesaurus by Schütze [8, 9]. This method obtains a word vector space based on word co-occurrences in close proximity in documents, whereas LSI creates a document vector space based on word frequencies throughout documents. The words that co-occur in a similar manner throughout documents are expected to be placed close to each other in the hyper-space. The authors refer to a database comprised of pairs of a word and its associated vectors as the “concept base”.

www.intechopen.com
The vector for a document is represented as the center of gravity with word vectors constructed from the document. In other words, the document vector is calculated as a normalized summation of the word vectors included in the document, as shown in Figure 2. The similarity between two documents $R(d_1, d_2)$ is calculated as a cosine coefficient between two centroid vectors.

In these methods (LSI and the co-occurrence thesaurus), documents having similar contents provide strong similarities even though the documents do not have the same expressions. This differs from methods based on word occurrences or Boolean full-text searches in that a high similarity degree is obtained only when documents contain similar expressions.

For CBVSM, words and documents that are different in nature from each other are mapped together in the same multi-dimensional space. This means that the methods provide not only similarity between words, but also similarity between words and documents, as well as between documents.

### 2.3.2 Construction of the Concept Base

The concept base is a knowledge base of words and is comprised of a set of words and their associated vectors. Each word is associated with a high-dimensional vector (a word vector), and the vector is statistically calculated from the target document set. The procedure to construct the concept base is illustrated in Figure 3 and stated in the following steps.

1. List every word that appears in the target documents. Let $N$ be the number of words and $w_i$ be the $i$-th word in the word list.
2. Create an $N \times N$ zero matrix. Let $C$ be the matrix and $C_{ij}$ be the $i$-th row and $j$-th column element in $C$.
3. Count the co-occurrences of words throughout the documents: if word $w_i$ and word $w_j$ co-occur within a specific distance in a sentence, increment
4. Reduce the rank of $C$ to $M$ using SVD, then obtain the reduced-rank matrix $C'(N$ rows $\times M$ columns).
5. $C'$ forms the concept base. The $i$-th row of $C'$ corresponds to the word vector for word $w_i$.

![Concept Base Construction Diagram]

The maximum value of $N$ is limited by the computing resources. In the case of resource constraints, the word list is truncated based on the occurrence count after step 1. Although $M$ can be 1 to $N$ in theory, we use $M = 100$ because it has been reported that this value is appropriate for discerning the similarity between words [9].

2.4 Implementation of the IPN Function

The IPN function is achieved by applying CBVSM to an information filtering technique [12]. The system computes the similarity between keywords and every posted article, then sends out alerts for articles that have a similarity degree that exceeds a specified threshold. By using CBVSM for computing, the similarity degree reflects not only occurrences of the keywords themselves, but also occurrences of similar words in an article. Therefore, the method satisfies the requirement for the IPN function, that is, notification of an article which does not explicitly include the keywords but discusses a topic expressed by the keywords.

The procedure for achieving the IPN function is listed in the following steps (see Figure 4).

**Preparation 1.** Construct the concept base: The concept base has to be generated prior to opening a discussion board. For concept base generation, text documents describing the same areas as the discussion topics are needed. In an e-Learning program, teaching materials include instructional text that can be used for concept base generation.
**Preparation 2.** Register the keywords of the topic of interest: The user inputs into the system the keywords in which he/she is interested. The system computes the query vector referring to the concept base. A user should have several interests and all students taking a course automatically become users of the discussion board; therefore, the system stores a set of keywords and a query vector for each user.

**Step 1.** When someone posts an article, the system extracts nouns and verbs from the article, then computes the document vector based on the extracted nouns and verbs referring to the concept base.

**Step 2.** The following step is iterated while extracting each query vector for each user from query vector storage.

**Step 3.** The system computes the similarity degree between the query vector and the document vector as the cosine coefficient of these vectors. If the similarity degree exceeds the threshold value, which is defined as a system configuration value, the system notifies the user that an important article is posted.

The concept base generation is a heavily loaded process because it requires the count of word co-occurrences throughout the documents and several matrix calculations. However, this process must be done just one time prior to opening the discussion. For every posting, information filtering based on CBVSM is processed and its cost is identical to that of other methods based on the vector space model.

As stated above, using CBVSM for computing the similarity degree reflects not only occurrences of the keywords themselves, but also occurrences of similar words in an article. For example, assume that student B registers "urgent stop" as a keyword, and student A posts an article including the word "emergency stop." The system described here can notify student B of this article because the similarity degree between words "urgent stop" and "emergency stop" is quite high according to the concept base. In other words, the concept base knows that the words "urgent stop" and "emergency stop" are similar.

The mechanism for adaptation to users is also embedded in the system. To adapt the IPN function to users, a user relevance feedback technique is employed. Because CBVSM is an extension of the vector space model, there relevance feedback technique for the conventional vector space model [1] can be applied directly.
Preparation 2. Register the keywords of the topic of interest: The user inputs into the system the keywords in which he/she is interested. The system computes the query vector referring to the concept base. A user should have several interests and all students taking a course automatically become users of the discussion board; therefore, the system stores a set of keywords and a query vector for each user.

Step 1. When someone posts an article, the system extracts nouns and verbs from the article, then computes the document vector based on the extracted nouns and verbs referring to the concept base.

Step 2. The following step is iterated while extracting each query vector for each user from query vector storage.

Step 3. The system computes the similarity degree between the query vector and the document vector as the cosine coefficient of these vectors. If the similarity degree exceeds the threshold value, which is defined as a system configuration value, the system notifies the user that an important article is posted.

The concept base generation is a heavily loaded process because it requires the count of word co-occurrences throughout the documents and several matrix calculations. However, this process must be done just one time prior to opening the discussion. For every posting, information filtering based on CBVSM is processed and its cost is identical to that of other methods based on the vector space model.

As stated above, using CBVSM for computing the similarity degree reflects not only occurrences of the keywords themselves, but also occurrences of similar words in an article. For example, assume that student B registers "urgent stop" as a keyword, and student A posts an article including the word "emergency stop." The system described here can notify student B of this article because the similarity degree between words "urgent stop" and "emergency stop" is quite high according to the concept base. In other words, the concept base knows that the words "urgent stop" and "emergency stop" are similar.

The mechanism for adaptation to users is also embedded in the system. To adapt the IPN function to users, a user relevance feedback technique is employed. Because CBVSM is an extension of the vector space model [1], the relevance feedback technique for the conventional vector space model [1] can be applied directly.

Fig. 4. Implementation of the IPN Function

When a user indicates that an article notified by the system is suitable for his/her interest, the system modifies the query vector to be close to the document vector of the article. Conversely, when a user indicates that an article is incorrectly notified, the system modifies the query vector to be far from the document vector. This modification to the query vector is stored and the system uses the modified query vector hereafter.

2.5 Performance Evaluation

Performance of the information retrieval functions including the IPN function is measured by "precision" and "recall" [1]. These measures are evaluated empirically using actual data sets. Precision corresponds to the measure of reliability and recall corresponds to the measure of completeness. Let $N_d$ be the number of retrieved documents and $N_c$ be the number of correct documents of the retrieved documents. Precision $P$ is defined as

$$P = \frac{N_c}{N_d}.$$

In addition, let $N_R$ be the number of correct documents among all documents in the data set; in other words, the number of complete answer sets. Recall $R$ is defined as

$$R = \frac{N_c}{N_R}.$$

2.5.1 Precision and Recall for the IPN function

In this section, the performance of the IPN function is evaluated using articles posted on the BBS of an actual e-Learning class [11]. If the number of articles is not sufficiently large, the evaluation results for a large number of articles posted on an open BBS on the Internet are also reported. The performance is compared with methods based on conventional models:
TF × IDF vector space model [7] and probabilistic model [5]. The performance improvement with user relevance feedback is also shown.

The articles posted on the BBS of the actual e-Learning course "Information Technology Basics" were used for the performance evaluation. As an example, a teacher requested students to discuss Internet protocols, then 38 articles were posted. The topics of the articles included the collision control technique on the Ethernet, message transfer protocols, and so on. For constructing the concept base, a large size (approximately 8,000 words) of instructional text on information technology was used. The text was used for obtaining statistics for conventional models.

Tables 1 and 2 show the results of the evaluation. "CSMA/CD" was assumed as a word of interest for the results in Table 1, and "NNTP" was assumed in Table 2. Complete answer sets can be constructed by reading all discussions in the BBS and judging whether each article is related, if the number of articles is small. For keyword "CSMA/CD," 10 articles discussed that topic, and 3 articles among them did not include the word "CSMA/CD" explicitly. Table 1 shows that the proposed method found all correct articles as important postings. Therefore, the recall rate and precision rate are 1. In contrast, methods based on the conventional models found only 7 correct articles out of 10 and 1 wrong article. One. For keyword "NNTP," 3 articles discussed that topic, and one article among them did not include the keyword explicitly. The proposed method also found all correct articles for this keyword, whereas the conventional methods found only two articles. The results prove that the proposed method has the capability to find related articles which do not include the keywords explicitly, whereas conventional methods cannot find them.

On the performance evaluation above, the number of articles was quite small. Since the IPN function is useful for the BBS on which a large number of articles is posted, a performance evaluation for a large-scale discussion board is desired. However, the authors had no class with a large-scale discussion of these terms. Instead, an evaluation using articles posted on an active BBS on the Internet was conducted. A total of 5946 articles were downloaded from a BBS discussing the Linux operating system and was assumed to be posted on our BBS. Topics "configuration for wireless LAN" and "detection of a hard disk drive" were assumed to be the users' interest, and the keywords were {"wireless," "lan," "configuration"} and {"hdd" and "detection"}, respectively. In this evaluation, the complete answer set could not be determined, because there were large numbers of articles, and it is not feasible for a human to read and judge every article. What a human can do is to judge whether a notified article is correct. Therefore, the precision rate could be obtained but the recall rate could not.

<table>
<thead>
<tr>
<th>Method</th>
<th># Notified</th>
<th># Correct</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>CBVSM</td>
<td>3</td>
<td>3</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>TF×IDF</td>
<td>2</td>
<td>2</td>
<td>1.00</td>
<td>0.67</td>
</tr>
<tr>
<td>Probabilistic</td>
<td>2</td>
<td>2</td>
<td>1.00</td>
<td>0.67</td>
</tr>
</tbody>
</table>

Keyword = "NNTP", # correct articles = 3,
# correct articles without word "NNTP" = 1
Table 2. Results for a small number of articles (2)
Table 1 Results for a small number of articles (1)

<table>
<thead>
<tr>
<th>Method</th>
<th># Notified</th>
<th># Correct</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>CBVSM</td>
<td>10</td>
<td>10</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>TFxIDF</td>
<td>8</td>
<td>7</td>
<td>0.87</td>
<td>0.70</td>
</tr>
<tr>
<td>Probabilistic</td>
<td>8</td>
<td>7</td>
<td>0.87</td>
<td>0.70</td>
</tr>
</tbody>
</table>

Keyword = "CSMA/CD", # correct articles = 10,
# correct articles without word "CSMA/CD" = 3
Table 1 Results for a small number of articles (1)

Tables 3 and 4 shows the results. The results suggest that the proposed method performs better than conventional methods even for a large-scale discussion board.

### 2.5.2 Performance Improvement with User Feedback

Performance improvement with user feedback was evaluated using the document set and keywords used above in the large-scale discussion board. Tables 5 and 6 show the results. As shown in the tables, considerable improvement in the precision rate is obtained with user feedback.

Table 3. Results for a large number of articles (1)

<table>
<thead>
<tr>
<th>Method</th>
<th># Notified</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>CBVSM</td>
<td>88</td>
<td>0.75</td>
</tr>
<tr>
<td>TFx IDF</td>
<td>43</td>
<td>0.62</td>
</tr>
<tr>
<td>Probabilistic</td>
<td>60</td>
<td>0.19</td>
</tr>
</tbody>
</table>

Keywords = "wireless lan configuration"

Table 4. Results for a large number of articles (2)

<table>
<thead>
<tr>
<th>Method</th>
<th># Notified</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>CBVSM</td>
<td>37</td>
<td>0.51</td>
</tr>
<tr>
<td>TFx IDF</td>
<td>19</td>
<td>0.42</td>
</tr>
<tr>
<td>Probabilistic</td>
<td>41</td>
<td>0.43</td>
</tr>
</tbody>
</table>

Keywords = "hdd detection"

Table 5. Performance Improvement with User Feedback (1)

<table>
<thead>
<tr>
<th># fb cycles</th>
<th># Notified</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>88</td>
<td>0.75</td>
</tr>
<tr>
<td>1</td>
<td>83</td>
<td>0.80</td>
</tr>
<tr>
<td>2</td>
<td>70</td>
<td>0.86</td>
</tr>
<tr>
<td>3</td>
<td>63</td>
<td>0.90</td>
</tr>
<tr>
<td>4</td>
<td>54</td>
<td>0.93</td>
</tr>
</tbody>
</table>

Keywords = "wireless lan configuration"

Table 5. Performance Improvement with User Feedback (1)

### 2.5.3 Discussions

The evaluation results suggest that the CBVSM-based method provides superior performance compared with conventional filtering methods for a wide range of sizes of
discussion boards.
The IPN function in iBBS for e-Learning differs from a normal information filtering function in that obtaining statistics on articles is difficult prior to opening the discussion board.

<table>
<thead>
<tr>
<th># fb cycles</th>
<th># Notified</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>37</td>
<td>0.51</td>
</tr>
<tr>
<td>1</td>
<td>34</td>
<td>0.59</td>
</tr>
<tr>
<td>2</td>
<td>29</td>
<td>0.64</td>
</tr>
<tr>
<td>3</td>
<td>28</td>
<td>0.60</td>
</tr>
<tr>
<td>4</td>
<td>25</td>
<td>0.68</td>
</tr>
</tbody>
</table>

Keywords = "hdd detection"
Table 6. Performance Improvement with User Feedback (2)

Instead, the ontology for the discussion articles is implicitly contained in the teaching materials. CBVSM can capture the ontology from the teaching materials virtually, and so the proposed method is expected to provide appropriate judgment even if it for the very first article in the discussion board. In contrast, the methods based on conventional models require statistics on the articles; however, these cannot be obtained in a BBS. Only one instructional text is used for constructing the concept base and obtaining actual statistics of the documents; therefore, the IDF value for every word is constantly 1 in the TF × IDF model and a priori document probability is also 1 in the probabilistic model. This degrades the performance of conventional methods. As shown in the results in Tables 1 and 2, the proposed method provides perfect precision whereas conventional methods return incorrect articles.

CBVSM is also able to discern the relevance between a topic and an article even though the article does not include the topic keyword, whereas conventional methods basically only detect articles including the keywords. The results show that the CBVSM-based method has a recall rate of 1.0, in contrast to the conventional methods, which cannot detect articles that do not explicitly include the keywords.

In addition, the results prove that user relevance feedback improves the performance, as expected.

3. The Assignment Management System with Plagiarism Detection: pAMS

3.1 Requirements for pAMS
Assignments are important for e-Learning programs to motivate students and track their learning. Since a student of an e-Learning program writes his/her report in electronic form, it is very easy for him/her to copy text from other reports or web sites. Consequently, it is a very heavy burden for a teacher to check for plagiarism in these reports. Therefore, an assignment management system that has a plagiarism detection function is needed. As an added benefit, if students know that the assignment management system automatically detects plagiarism, they will avoid copying text from others. This leads to the prevention of plagiarism.

The plagiarism detection function must have the following capabilities.
Some students may modify the source text to avoid detection. Therefore, the detection function has to be able to find similar parts in two documents, even if they are skillfully modified and not copied word by word.

- It is not feasible to check the similarity of the target text against every web page on the Internet because of resource limitations. Therefore, the function must be able to narrow down the web pages to suspected ones before checking the similarity.

3.2 System Structure and Plagiarism Detection Subsystem
The overall structure of pAMS is shown in Figure 5. This system consists of two parts: an assignment management subsystem and a plagiarism detection subsystem [13]. The assignment management subsystem is a module that is compatible with a learning management system (LMS) and provides functions that include assignment configuration, report submission, and submitted report references. The plagiarism detection subsystem detects plagiarism by comparing each pair of reports and web pages.

Figure 6 shows the structure of the plagiarism detection subsystem. When a target report is given, it detects any part similar to other reports and web pages by the following sequence.

1. Extracts the query words from the target document; The words are used for retrieving web pages that are possible source documents if the target report copied text from the web.
2. Issues the query to an Internet search service (e.g., Google), then retrieves and stores $N$ web pages found by the query, where $N$ is a configuration parameter (50 to 100 is appropriate).
3. Compares the target report with each of the other reports and stored web
4. When the above step is finished for every document, a list of suspect parts is generated based on the accumulated comparison results.

The query words should include words expressing the content of the report and simultaneously the words that are able to narrow the scope of the search. High-frequency words in the document express the main topic and longer words limit the scope. Therefore, the authors employ a combination of high-frequency words and long words as query words. For the detection of similar parts, a combination of the TF-IDF vector space method with the Smith-Waterman algorithm [10, 4, 6] is employed. The TF-IDF vector space model is used for computing the degree of similarity between one sentence and another. Based on sentence-by-sentence similarity degrees, quasi-contiguous parts of similar sentences are found with the Smith-Waterman algorithm. The TF-IDF vector space model technique can recognize the copied sentences revised with the substitution of a word with a synonym, and the Smith-Waterman algorithm can find plagiarized parts revised with word re-sequencing.

3.3 Performance Evaluation

For the performance evaluation of plagiarism detection, students were employed to write intentionally plagiarized reports. Four subjects, including "embedded systems," "copyright," "open source software" and "derivatives," were given, and the students searched web pages describing these subjects, then plagiarized text from these pages. The web pages used for plagiarism were stored in a source document pool and their URLs were recorded for each plagiarized report. A total of 900 plagiarized reports were written. Some of the plagiarized reports were comprised of "dead" copies of parts from one, two, or three web pages. Others
were revised with several levels of modification. Non-plagiarized reports were also written. The number of these was 20.

First, to evaluate the accuracy of the plagiarism judgment based on the similar part detection, the web pages used for plagiarism were assumed to be retrieved perfectly. Each of the reports in the test set was input into the system, then the system checked it against web pages in the source document pool and indicated whether it was plagiarized. The precision for plagiarism judgment \( P_i \) is defined as

\[
P_i = \frac{N_C}{N},
\]

where \( N \) is the number of reports in the test set and \( N_C \) is the number of reports for which the system with the source document pool gives a correct judgment.

Next, the accuracy of retrieving possible source web pages was evaluated. Each of the plagiarized reports was input into the system, then the system extracted query words for the reports, submitted the query to the Internet search service, and retrieved 100 web pages that matched the query. The precision for suspected web page retrieval \( P_r \) is defined as

\[
P_r = \frac{N_R}{N_P},
\]

where \( N_P \) is the number of plagiarized reports, and \( N_R \) is the number of reports for which at least one of the source web pages is included in the retrieved web pages.

Finally, the overall accuracy of the system was evaluated. Precision \( P_o \) is expressed as

\[
P_o = \frac{N_C}{N},
\]

where \( N_C \) is the number of the reports for which the system retrieves web pages and eventually gives a correct judgment.

Table 7 shows the results. Precision \( P_i \) demonstrates that if the source web pages can be correctly retrieved, the system can accurately detect plagiarism from the web pages. However, retrieving the source web pages is not sufficiently accurate. This fact decreases the overall precision.

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>( P_i )</td>
<td>0.93</td>
</tr>
<tr>
<td>( P_r )</td>
<td>0.54</td>
</tr>
<tr>
<td>( P_o )</td>
<td>0.49</td>
</tr>
</tbody>
</table>

Table 7. Evaluation Results

4. Summary and Future Work

This chapter describes an intelligent interaction support system for e-Learning systems. The support system includes an intelligent bulletin board system (iBBS) and a plagiarism-detecting assignment management system (pAMS) that enable smooth communication and fair evaluation of e-Learning classes.

The functional requirement of the iBBS is clarified, the method using the concept-based vector space model (CBVSM) for the important posting notification (IPN) function is proposed, and evaluation results are reported. The results show that the proposed method provides superior precision and recall rates compared with conventional methods based on the TF×IDF vector model and probabilistic model.
The system structure, pAMS, comprised of an assignment management subsystem and a plagiarism detection subsystem is introduced. The structure of the plagiarism detection subsystem is introduced as a combination of the retrieval of suspected web pages and the detection of similar parts of text. Similarity discernment between sentences based on the TF-IDF vector space model and quasi-contiguous part recognition with the Smith-Waterman algorithm are employed for the similar part detection. A performance evaluation with a limited scale set of intentionally plagiarized reports was conducted. The results show that the overall precision is approximately 50% because of insufficient accuracy in the retrieval of the source web pages.

Improvement of the source web page retrieval capability and performance evaluations with a large-scale set of real plagiarized reports are tasks for future study.

5. References

The system structure, pAMS, comprised of an assignment management subsystem and a plagiarism detection subsystem is introduced. The structure of the plagiarism detection subsystem is introduced as a combination of the retrieval of suspected web pages and the detection of similar parts of text. Similarity discernment between sentences based on the TF-IDF vector space model and quasi-contiguous part recognition with the Smith-Waterman algorithm are employed for the similar part detection. A performance evaluation with a limited scale set of intentionally plagiarized reports was conducted. The results show that the overall precision is approximately 50% because of insufficient accuracy in the retrieval of the source web pages.

Improvement of the source web page retrieval capability and performance evaluations with a large-scale set of real plagiarized reports are tasks for future study.

5. References


This book is consisting of 24 chapters which are focusing on the basic and applied research regarding e-learning systems. Authors made efforts to provide theoretical as well as practical approaches to solve open problems through their elite research work. This book increases knowledge in the following topics such as e-learning, e-Government, Data mining in e-learning based systems, LMS systems, security in e-learning based systems, surveys regarding teachers to use e-learning systems, analysis of intelligent agents using e-learning, assessment methods for e-learning and barriers to use of effective e-learning systems in education. Basically this book is an open platform for creative discussion for future e-learning based systems which are essential to understand for the students, researchers, academic personalas and industry related people to enhance their capabilities to capture new ideas and provides valuable solution to an international community.

How to reference
In order to correctly reference this scholarly work, feel free to copy and paste the following:
