Updateable PAT-Tree Approach to Chinese Key Phrase Extraction using Mutual Information: A Linguistic Foundation for Knowledge Management

Thian-Huat Ong and Hsinchun Chen
Department of Management Information Systems
University of Arizona
Tucson, AZ 85721, U.S.A.
{tong , hchen}@mail.bpa.arizona.edu
http://ai.bpa.arizona.edu/

Abstract

There has been renewed research interest in using the statistical approach to extraction of key phrases from Chinese documents because existing approaches do not allow online frequency updates after phrases have been extracted. This consequently results in inaccurate, partial extraction. In this paper, we present an updateable PAT-tree approach. In our experiment, we compared our approach with that of Lee-Feng Chien with that showed an improvement in recall from 0.19 to 0.43 and in precision from 0.52 to 0.70. This paper also reviews the requirements for a data structure that facilitates implementation of any statistical approaches to key-phrase extraction, including PAT-tree, PAT-array and suffix array with semi-infinite strings.

A. Introduction — From Information Retrieval to Knowledge Management

In this era of the Internet and distributed multimedia computing, new and emerging classes of information technologies have swept into the lives of office workers and everyday people. As such technologies and applications become more overwhelming, pressing, and diverse, solutions for several well-known information technology problems have become an even more urgent need. *Information overload*, a result of the ease of information creation and representation via Internet and WWW, has become more evident (Blair & Maron, 1985) (Chen, Martinez, et al., 1998). Significant variations in database formats and structures, the richness of information media (text, audio, and video), and an abundance of translingual information content also require different information interoperability (Paepcke et al., 1996) (Lesk, 1997).

Several new federal and business initiatives have emerged to attempt to transform our information-glut society into a knowledge-rich society. In the United States NSF Knowledge Networking (KN) initiative, scalable techniques to improve semantic bandwidth and knowledge bandwidth are considered among the priority research areas, as described in the KN report: “The Knowledge Networking initiative focuses on the integration of knowledge from different sources and domains across space and time... KN research aims to move beyond connectivity to achieve new levels of interactivity, increasing the semantic bandwidth, knowledge bandwidth, activity bandwidth, and

Proceedings of the Second Asian Digital Library Conference, November 8-9, 1999, pp. 63-84.

cultural bandwidth among people, organizations, and communities" (Chen, 1998) (Chen & Ng, 1995).

"Knowledge networking," or a more general term, "knowledge management" (KM), has attracted significant attention from academic researchers and even executives in Fortune 500 companies. O'Leary provides the following definition: "Enterprise knowledge management entails formally managing knowledge resources in order to facilitate access and reuse knowledge, typically by means of advanced technology. KM is formal in that knowledge is classified and categorized according to a pre-specified -- but evolving -- ontology into structured and semi-structured data and knowledge bases" (O'Leary, 1998). Knowledge management systems may employ various computational techniques, including linguistics analysis, data mining, machine learning, agents, information retrieval, and human-computer interactions.

The information technology think tank Gartner Group defines KM as: "a discipline that promotes an integrated approach to identifying, capturing, retrieving, sharing and evaluating an enterprise's information assets. These information assets may include databases, documents, policies and procedures as well as the uncaptured tacit expertise and experience resident in individual workers" (Gartner Group, 1998). Gartner Group predicts that KM may become the third wave of the Net, making significant impacts on business practices and the US economy in the next century. Since 1997, 30% of Fortune 500 companies have either added a chief knowledge officer (CKO) position or converted the chief information officer (CIO) position into CKO. Many Fortune 500 and IT companies have considered knowledge sharing their most critical strategic area (Davenport, 1995) (Davenport & Prusak, 1998).

Although it has been variously defined, it is evident that knowledge management exists at the enterprise level (Davenport & Prusak, 1998) and is quite distinct from mere information (Davenport & Prusak, 1998) (Nonaka, 1994) (Teece, 1998). Also apparent in this area are the challenges that knowledge management poses to an organization. In addition to being difficult to manage, knowledge traditionally has been stored on paper or in the minds of people (Davenport, 1995) (O'Leary, 1998). The KM problems facing many firms stem from barriers to access and utilization resulting from the content and format of information (Jones & Jordan, 1998) (Rouse, Thomas, & Boff, 1998). These problems make knowledge management creation and utilization a complex and daunting process. Nevertheless, new knowledge management technologies have started to emerge in a number of different applications and organizations, such as virtual enterprising (Chen, Liao, & Prasad, 1998), joint ventures (Inkpen & Dinur, 1998), aerospace engineering (Jones & Jordan, 1998), and digital libraries (Chen, 1998) (Chen, Houston, et al., 1998).

The just-released PITAC (President's Information Technology Advisory Committee) report concluded that in the United States "the current Federal program is inadequate to start necessary new centers and research programs... The end result is that critical problems are going unsolved and we are endangering the flow of ideas that have fueled the information economy." Among the priorities for research, the PITAC report suggests that the federal program should "support fundamental research in capturing, managing, analyzing, and explaining information and in making it available for its myriad of users" (Schatz & Chen, 1999).

In order to create a "knowledge map" from diverse information sources, Gartner Group has suggested a bottom-up approach that includes data extraction, linguistic analysis, dictionary/thesaurus creation, semantic networks, clustering/categorization, and concept yellowpages (Gartner Group, 1998). These layers of techniques can serve as the foundation for addressing multimedia and translingual interoperability as well. Other related multimedia processing and translingual indexing and machine translation techniques need to be developed to support additional functionality.

B. Prior Research

Phrase extraction, commonly called word segmentation, for the Chinese language means finding the longest phrase in a word string with precise meaning (Kwok, 1997). This is considered the major barrier to text retrieval, especially for Asian languages (Chien & Pu, 1996) (Wu & Tseng, 1993) (Salton, 1989). However, key phrase extraction, commonly known as indexing, goes further, finding the phrases that are representative of a document. Indexing is fundamental to the success of many recent digital library applications and semantic information retrieval techniques (Schatz & Chen, 1999) (Chen, 1998) (Schatz & Chen, 1996) (Lin & Chen, 1996). The need to harness the tremendous amount of information available from online scientific literature and the Internet and to achieve semantic retrieval have progressively prompted more interest and research to advance the state of the art of phrase extraction (Chen, Houston, et al., 1998) (Chen, Chung, et al., 1998) (Chien, 1997).

Prior research in phrase extraction can be categorized into three categories, along with a hybrid approach that combines two or more of these basic approaches.

1. Dictionary approach:

This approach uses a human-generated dictionary to break out known phrases in the dictionary that have more than one character. The major advantage is quick and simple implementation. A lot of systems are still using this approach (Bian & Chen, 1998) (Li & Xing, 1998) (Chen, 1997), especially when new words are not a concern or the dictionary covers the target documents. Because there usually is a partial matching dilemma, several strategies have been devised to improve the matching process, including maximum-matching, minimum-matching, forward-matching, backward matching, bi-directional, and other heuristics. However, the effectiveness of this approach is largely limited by the comprehensiveness of the dictionary, which cannot effectively deal with proper nouns such as names and places and new terms constantly being created in special domains.

2. Linguistic approach:

An alternative approach uses syntactic and/or semantic knowledge bases, heuristics, or rules to extract phrases (Wu & Tseng, 1995). From the natural language processing research in English, it has been demonstrated to be able to achieve higher precision, depending on the system design (Church, 1988) (Brill, 1995) (Voutilainen, 1997). However, although a number of articles have suggested the possibility, there has been no extensive computational linguistic research for the Chinese language, and implementing this approach remains a difficult task (Wong & Li, 1998). The major drawback is that building complete syntactic and semantic knowledge bases for large and complex domains has proved daunting.

3. Statistical approach:

This approach generally learns valuable statistical information from a usually large corpus to extract possible phrases. For a thorough review, readers are referred to (Su, Chaing, & Chang, 1996). This approach was shown to generate good performance in extracting key phrases (Chien, 1998) (Yang et al., 1998) (Chien, 1997) (Chen, 1997) and it has a traditional tie to the n-gram approach with a small number of n, such as 2, 3, or 4. However, when Chien first used PAT-tree for Chinese, the n became unrestricted. Despite its extensive computation, the technique does not require any labor-intensive creation of dictionary or knowledge base and is capable of capturing emerging terminology in the corpus. One major drawback is that it is not able to extract valid phrases that do not occur sufficiently frequently.

In order to achieve key phrase extraction, we need to apply additional processing. Although perhaps not as precise as the linguistic approach, the statistical approach is still a good choice for extracting key phrases. Since it is based on the probable occurrences in the collection, low-frequency phrases, which are more likely not to be key phrases, will be removed automatically. Therefore, a statistical approach is very well suited for extracting key phrases, especially when combined with our proposed updateable data structure.

However, there are some challenges for the statistical approach. Kenneth Church has pointed out that some Bible literature has repeated patterns with up to 400 words (Church, 1997) and constitutes a challenge for the n-gram technique because, without removal of the pattern every sub-pattern in the 400-word sentence could be extracted. Removal of a pattern from the corpus affects the frequency distribution of the corpus, especially repetitive removals of many sub-patterns. An attempt to figure out the new frequency without updating was tried but failed. Different researchers have tried different heuristics to extract better phrases and avoid partial phrases.

Our proposed approach is an extension of Lee-feng Chien's PAT-Tree-Based Keyword Extraction for Chinese Information Retrieval (Chien, 1997). Chien's approach can be considered a generic model for n-gram, because it does not restrict the value of n to any fixed number. The key distinctions between these two approaches are their extracting methods and the use of stop words.

The first and most important distinction is the heuristics to extract phrases. Chien's approach examines from the longest possible patterns to the smallest patterns, each time using three iterative inclusion-exclusion heuristics, which backtrack to the longer patterns if necessary. The backtracking heuristic tries to avoid extracting incorrect partial phrases, but it does not always guarantee the outcome.

Our approach is much simpler: starting from the longest possible patterns to the smallest patterns, and each time uses a frequency and mutual information threshold to decide whether to extract and thus *remove* the phrase (update the frequency). The heuristic is shown as pseudo-code as

```
for len:=max_len down to 2
  repeat until no more changes
    for pat := all patterns of length len
      if freq(pat)>TH_f and mutual_info(pat)>TH_mi
        output pat as an extract phrase
        remove pat from PAT-tree
```

Here, we illustrate how online frequency updating after removal of extracted key phrases helps to extract more precise phrases. However, since a more real-life and elaborate example would obscure the discussion, we create a hypothetical scenario in which a corpus has only 50 人工智慧 (artificial intelligence). The algorithm starts with the longest pattern, which is 4. Since both the frequency (50) and mutual information (1.0) are high, this phrase is extracted. Without removal of this newly extracted phrase, the incorrect partial pattern 人工智 or 工智慧 will have a chance of being extracted later. On the other hand, if we remove the newly extracted phrase from this corpus (i.e., update the frequency), then there is nothing left in this corpus, and subsequent partial extractions will be impossible. It might be asked, "What about the correct partial phrases 人工 (artificial) and 智慧 (intelligence)?" Of course, they will not be extracted in this hypothetical scenario, but in a real-life corpus, if the smaller phrases 人工 and 智慧 happen independently of the longer phrase 人工智慧 frequently enough (i.e., are important by themselves), they will be extracted too. Yet in a corpus that talks mostly about MIS and artificial intelligence, the smaller phrases 人工 and 智慧 are not useful and so are not extracted.

In our updateable approach, we therefore do not need a backtracking heuristic to avoid partial extraction, but we need to develop a new data structure that supports consistent on-line frequency update after removing patterns that we have already extracted. In this case, the 400-word pattern in the Bible will now be extracted only once and its sub-patterns will not be extracted. Detailed discussion of implementation is deferred to a later section titled Online Frequency Update.

A second difference is that Chien's approach does not call for stop words, whereas our preliminary experience suggests that the use of stop words could increase the accuracy of the results. In order to isolate the effect of stop words, the experiment also includes our approach with and without stop words, so that we can identify the improvements of our approach without stop words over Chien's approach, and then that for our approach with stop words.

C. Mutual Information

A number of mutual information metrics are being used in recent statistical approaches. *Mutual information* is one commonly used to measure how consistently two patterns occur together in a corpus. Traditionally, *MI(A,B)* is defined as the log-likelihood ratio of the joint probability of pattern *A* and *B* over the probability of *A* and *B*:

$$MI(A,B) = \log_2\left(\frac{P(A,B)}{P(A)P(B)}\right)$$

In Chinese phrase extraction, let *c* be a pattern of interest (e.g., 人工智慧 meaning artificial intelligence), and let *left* and *right* be its sub-pattern:

c = 人工智慧 "artificial intelligence"
left = 人工智 (partial word, no meaning)
right = 工智慧 (partial word, no meaning)

Then, we can obtain the mutual information for the pattern *c*,

$$MI_c = \log_2\left(\frac{P(c)}{P(left)P(right)}\right)$$

The intuition arises from the independent event where $P(c)=P(left)P(right)$, in which $MI_c=0$; if $MI_c>>0$ and the patterns co-occur very well. The probability of pattern c is the ratio of the frequency f_c over the total frequency F:

$$MI_c = \log_2\left(\frac{f_c/F}{f_{left}/F \cdot f_{right}/F}\right) = \log_2\left(\frac{Ff_c}{f_{left} \cdot f_{right}}\right)$$

If we are only interested in deciding whether the mutual information exceeds a certain threshold TH such that $MI_c \geq TH$, then we can simplify the calculation without involving the slower logarithmic function, as in

$$\frac{f_c}{f_{left} \cdot f_{right}} \geq TH', \quad where\ TH' = 2^{TH}/F$$

On the other hand, Chien defined an alternate mutual information metric, called significance estimation function, which is the ratio of the probability of the

$$MI'_c = \frac{P(c)}{P(subpatterns\ of\ c)}$$

$$= \frac{P(c)}{P(left) + P(right) - P(c)}$$

Intuitively, when pattern c and its sub-pattern co-occur very well, MI_c' is high and close to 1, so the pattern c is more likely to form a phrase than its left and right sub-patterns alone. On the other hand, if MI_c' is low and close to 0, the pattern c is not likely to form a phrase. The calculation can be simplified using frequency:

$$MI'_c = \frac{f_c/F}{f_{left}/F + f_{right}/F - f_c/F}$$

$$= \frac{f_c}{f_{left_c} + f_{right_c} - f_c}$$

In fact, this turned out to be the same as the Jaccard measure, which is also commonly used in Information Retrieval and defined as $Jaccard(A,B) = |A \cap B| / |A \cup B|$.

Therefore, both mutual information metrics are equally easy to calculate for threshold cut-off, but no experiment has been conducted to determine which metric performs better.

D. Required Data Structure

Since all the mutual-information-based approaches need to access all the possible patterns and find their frequency of occurrence, a highly efficient searching algorithm, both in terms of space and time, must be employed, or else it becomes too large or too slow for practical use. There has been a tremendous amount of research in this area, in particular in the field of fast text searching (Beaza-Yates & Gonnet, 1996) (Manber & Myers, 1993). Basically, we have a static text corpus that can be preprocessed to increase the performance of the search.

1. Space Efficiency — Semi-Infinite Strings

In general, a string of length N has $N(N+1)/2$ sub-patterns. For example, there are 10 sub-patterns for 'abcd': a, b, c, d, ab, bc, cd, abc, bcd, and abcd. Even if we keep all sub-patterns as null-terminated strings in a flat sequence without additional tree structure, the number of bytes required is still

$$\sum_{k=1}^{N}(N-k+1)(k+1) = N(N+1)(N+5)/6 = O(N^3)$$

In the example, we need 30 bytes for 'abcd'. However, a typical 1MB corpus will require 166,666TB of memory, which is too demanding even for supercomputers. Furthermore, an $O(N^2)$ space data structure is no better, because an 1MB corpus still would will take 1TB. Without a better data structure to store all the sub-patterns, all n-gram techniques will become impractical for bigger N or unrestricted N. Therefore, it has been common for earlier attempts to be restricted to small N such as 2 to 4. In conclusion, we must look for an $O(N)$ space data structure.

We can view the training corpus as a static text, which is a single string padded at its right end with an infinite number of null (or any special) characters. A *semi-infinite string* (*sistring*) (Knuth, 1973) (Baeza-Yates & Gonnet, 1996) is the sequence of characters starting at any position of the text and continuing to the right. For example, the text of

```
abcd.......
```

has the following set of sistrings:

```
abcd.......
bcd.......
cd.......
d.......
```

By using a unique end-of-text symbol that appears nowhere else, we can guarantee that no one semi-infinite string is a prefix of another. Hence, semi-infinite strings can be unambiguously identified by their starting position. Therefore, the space requirement for this structure is only $O(N)$, for the text itself and the starting positions for the semi-infinite strings. Now, searching for a pattern simply becomes searching for a prefix in any of the semi-infinite strings. For example, we know 'bc' is a pattern, because it is found to be a prefix to the semi-infinite string 'bc……'.

For extracting Chinese phrases, Chien suggested using a punctuation mark, such as comma, period, and so on, as a boundary to process the text at the sentence level (Chien 1997), because a Chinese phrase will not contain any punctuation marks. For example, the string "詞彙自動抽取,減化索引困難" (phrase automatic extraction, alleviating indexing difficulty) generates the following semi-infinite strings:

```
詞彙自動抽取00000...
彙自動抽取0000000...
自動抽取000000000...
動抽取00000000000...
抽取0000000000000...
取000000000000000...
減化索引困難00000...
化索引困難0000000...
```

索引困難000000000...
引困難00000000000...
困難0000000000000...
難000000000000000...

2. Fast Searching — PAT-Tree, PAT-Array, Suffix Array

After we have created a data structure to store all the sub-patterns efficiently, we must ensure that it allows us to search prefixes very efficiently. The fastest searching algorithm is $O(1)$ hashing, but hashing is not suitable for long, variable-length keys and prefix searching. Therefore, we need to look for the next best solution at $O(\log N)$. There has been a lot of research in the field of fast text searching that has tackled this problem. For a review of the state of the art and various optimizations, readers are referred to (Manber & Myers, 1993).

Trie-based search is highly efficient, but it has two drawbacks: space wasted by one-way branching and complicated codes resulting from two different types of nodes (Sedgewick, 1998). In 1968, Morrison discovered *patricia* ("practical algorithm to retrieve information coded in alphanumeric") that avoids these two problems (Morrison, 1968). A patricia trie requires only N nodes to be constructed for N search keys, and it requires only about $\log_2 N$ bit comparisons and one full key comparison per search, $O(\log N + P)$. Patricia is fast, because it immediately jumps to the bits that matter for comparison, so the number of bit comparisons is never more than the length of the key. In fast text searching, a *PAT-tree* is a patricia trie for semi-infinite strings. Chien used a PAT-tree to perform fast searching during phrase extraction. For implementation details and example codes, readers are referred to (Sedgewick, 1998) (Gonnet & Baeza-Yates, 1991) (Morrison, 1968).

An alternate yet simple but equally fast solution is to use *PAT-array* or *suffix array* (Gonnet & Baeza-Yates, 1991), which basically is a sorted list of semi-infinite strings, and sorting can be done in $O(N \log N)$ using any standard sorting algorithm such as QuickSort. Then the search time for a prefix is similar to a binary search and thus can be done in $O(P \log N)$, where P is the length of the search key and n is the size of the corpus. Manber and Myers improved the search time to $O(P + \log N)$ in a new suffix array coupled with information about the longest common prefixes, which can be constructed in time $O(N)$ (Manber & Myers, 1993). The advantage of using a linear structure is that all the semi-infinite strings with the prefix that we are looking for are going to be in a consecutive block. Therefore, it will be much easier to figure out the frequency of a pattern (prefix) of interest.

In summary, the research in fast text searching such as PAT-tree, PAT-array or suffix array enables us to perform key phrase extraction in an efficient manner, in terms of both space and time.

E. Online Frequency Update

All the fast text-searching algorithms assume that the text is fixed and is never updated, thereby enabling us to preprocess the text for much faster subsequent searching. However, the ability to update is a desirable property, because deleting the bigger phrases after we extract them will improve the subsequent extraction of smaller phrases. However, removing a pattern from the text will affect the entire

fast text searching data structure, which cannot be easily updated without reconstruction, a costly process.

Two basic constraints must be observed to make updates possible: the text cannot be changed and the impact on the searching should be minimized. If the text cannot be changed, we need to introduce some flags to denote updates. However, a careless use of flagging will greatly degrade search performance, so we must consider the characteristics of the corpus. We have observed that the number of extracted phrases is much smaller than the number of all possible patterns. Therefore, we need to have a way to expedite counting when there has been no update.

We can use two flags for every character to accomplish this. Although implementation by representing them as two bits in a single byte to save space is possible, we make them separate to make discussion of them clear. The first flag is for deletion (D), and it is turned on when the corresponding segment of sub-pattern is extracted as a phrase. When a sub-pattern has this flag, it should not be counted toward the frequency. The second flag is for modification (M), and it is turned on for the entire sentence of which a portion has been extracted. This will expedite the counting of sentences not updated.

For example, the corpus has a pattern "詞彙自動抽取,減化索引困難" (phrase automatic extraction, alleviating indexing difficulty) and we have not extracted any phrases. The flags are empty ("-" to denote empty flag):

詞彙自動抽取,減化索引困難
- - - - - - - - - - - - - (for deletion)
- - - - - - - - - - - - - (for modification)

When we extract the phrase 自動抽取 (automatic extraction), the flags update to

詞彙自動抽取,減化索引困難
- - D D D D - - - - - - - (for deletion)
M M M M M M - - - - - - - (for modification)

When we see no flags in a pattern, we know no modification has taken place, and we can count the frequency quickly. When we see the flag M on a pattern, we need to proceed to a more careful and slower counting process to check whether that pattern has in fact been partially deleted by the D flag. Therefore, we are able to quickly and consistently count the updated frequency affected by those extracted phrases.

F. Architecture

Figure 1 shows the overall architecture of our approach, which is similar to (Chien, 1997), except for the updateable heuristics for extracting phrases from PAT-tree and stop wording. The process starts with a corpus having a sequence of Chinese characters. The input is a collection of Chinese text. In preprocessing, the text is broken down by punctuation marks and English words, a process which results in only chunks of Chinese characters.

1. Apply stop wording if needed.
2. Construct the PAT-tree and extract the key phrases. The entire process proceeds from the longest possible patterns to patterns with a length of two, because we are not interested in single characters. For each pattern, we use a frequency threshold to

decide whether further action is needed, to reduce unnecessary computations. If passed, the co-occurrence metric is calculated, and if it exceeds another threshold, then this pattern is extracted as a phrase and removed from the corpus, so the frequency distribution will be adjusted. Note that there is no need to backtrack to a longer pattern using this heuristic.

3. Filter the results with another PAT-tree constructed by a general corpus can be used to get rid of phrases that commonly exist in daily usage, such as newspapers.
4. Filter by examining the frequency distribution of the extracted phrases.

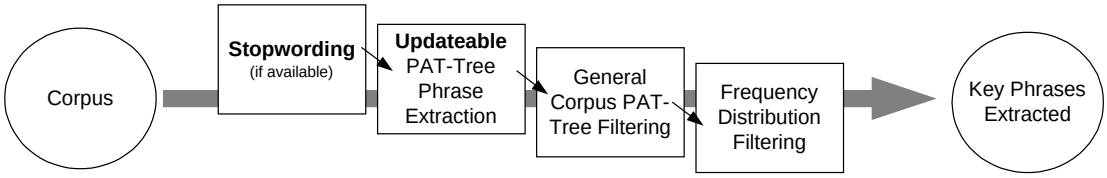

Figure 1. Proposed key phrase extraction architecture, with two key differences: the ability to update the PAT-tree after removal and the use of stop words.

G. Experiment Design

An experiment was conducted to compare the extraction results of our approach with those of the published approach by Chien. Since we could not obtain the author's original codes, we tried to recreate the algorithm described in (Chien, 1997). However, since Chien's approach did not use stop words, we experimented with two versions of our approach, one without stop words (AZ) for direct comparison and another with stop words (AZ-S) for isolating the effects of the use of stop words.

There are still a few other differences between our implementation of Chien's approach and the original one. First, the test bed is a partial collection of journal abstracts from Science and Technology Information Center (STIC, http://www.stic.gov.tw), focused in the field of MIS with 1700 items or 1MB of journal abstracts. The nature of a more specialized test-bed domain than the books or news articles used in the original paper may have caused a difference in the quality of extraction. Second, we built the general PAT-tree from a 3MB collection of news articles from http://www.sina.com during January to June of 1999. Third, we use the same set of about 500 stop words as the common-word lexicon. Fourth, the thresholds chosen for the extraction and frequency filtering process probably were different from those in the original experiment. However, all three algorithms use the same threshold values to make comparisons meaningful.

Our experiment was controlled to limit differences to the ability to update, in combination with use of stop words and without such use. Hence, although our implementation of Chien's algorithm may not be optimal, it does provide a viable and direct comparison of the effects of the ability to update and the use of stop words.

In order to avoid bias of subjects toward any particular approach, the Java experiment interface combined and showed the extracted phrases from all three approaches with those from a single interface underlined (see Figure 2). Therefore, the subject would not be able to tell whether an extracted phrase came from one particular approach, but because the system knew which extracted phrases came from which approach, it could correctly calculate the values of recall and precision for each approach.

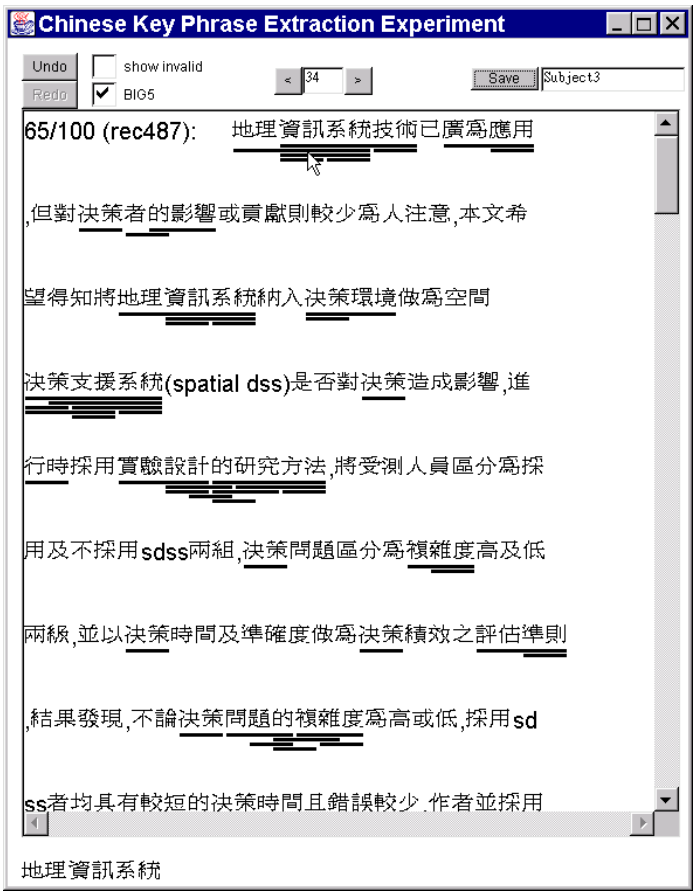

Figure 2. Experiment interface underlining of the extraction results from all three algorithms. The experiment subject was to remove the wrong segmentations and then pick the relevant key phrases without bias. The system then calculated recall and precision are for each algorithm.

This experiment involved four subjects who were native Chinese speakers and proficient in the field of MIS. Each subject was presented 40 randomly selected journal abstracts. Therefore, there was a total of 160 sample data for each approach and a total of 480 sample data (n=480) for the entire experiment. First of all, the subject was asked to remove extracted phrases that had been wrongly segmented, such as 策支援系統 and 的問題 (see Figure 2). Then, the subject was asked to pick out the good key phrases that he or she judged to be relevant and descriptive of the journal abstract at hand. For example, both 地理資訊系統 (geographic information system) and 決策支援系統 (decision support system) could be considered good key phrases.

H. Experiment Results and Analysis

Figure 3 shows the key phrase extraction results from all three approaches, where Chien represents our implementation of Chien's approach, AZ our approach without stop words, and AZ-S our approach with stop words. Both of our approaches produced fewer wrong segmentations than Chien's approach (3.3% and 0.5% compared to 19.8%). Our approach without stop words provided a direct head-to-head comparison with Chien's approach, and was able to produce more key phrases (3,169 compared with 2,326), while

the error rate of wrong segmentation still remained much lower (3.3% compared to 19.8%).

Comparing the effects of the presence of stop words, we found that the use of stop words cut down the number of extracted phrases by one third and kept the percentage of wrong segmentation even lower, at 0.5%.

Approach	Number of key phrases extracted	Percent of wrong segmentation
Chien	2,326	19.8%
AZ	3,169	3.3%
AZ-S	1,921	0.5%

Figure 3. The key phrase extraction results from three different approaches: Chien, AZ (without stopwords), and AZ-S (with stopwords).

The distribution of the mutual information value also changed when we introduced the ability to update, as shown in Figure 4. The values of the co-occurrence metric for the extracted key phrases in our approaches tended to be higher, whereas Chien's extracted key phrases were distributed more evenly between the threshold and the maximum value. This suggests that adding the ability to update confirmed our belief that deletion of extracted phrases tends to increase the success of subsequent phrase extraction.

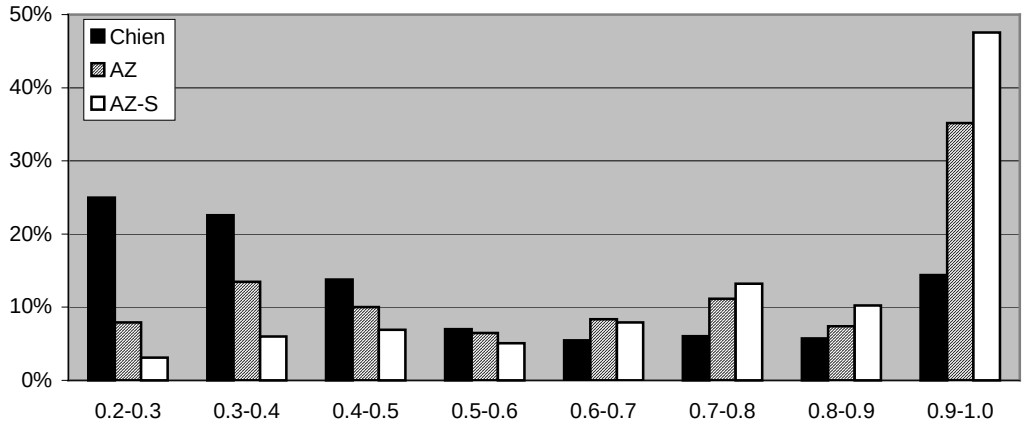

Figure 4. Comparison of distribution histograms of mutual information values of phrases extracted by different approaches.

Recall and precision, which are commonly used in information retrieval, were used to evaluate the experiment results. For a journal abstract, each approach extracted a set *R* of key phrases. Let |*R*| be the number of key phrases in this set. For this abstract, there is an answer set *A* of key phrases, which best describes the abstract. However, since the answer set is very subjective to the subject, we shall use all the key phrases that the subject has chosen. Then, *Ra* represents the intersection of the sets *R* and *A*. Figure 5 illustrates the relationship between these sets. The recall and precision measures are defined as follows.

- **Recall** is the fraction of relevant key phrases (the set *R*) extracted

$$Recall = \frac{|Ra|}{|R|}$$

- **Precision** is the fraction of the extracted key phrases (the set *A*) that are relevant

$$Precision = \frac{|Ra|}{|A|}$$

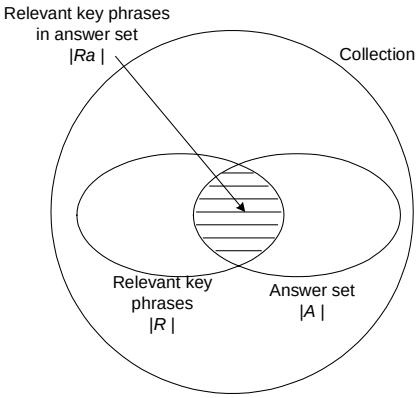

Figure 5. Precision and recall for a given example of key phrase extraction for a journal abstract.

For comparison, we used the average value of recall and precision, and the results are tabulated in Figure 6 and Figure 7. Both of our approaches, with and without stop words, showed significant improvement over Chien's approach, at a .05 confidence level in both recall and precision. Although there was no statistical difference in precision between our approach without stop words (AZ) and with stop words (AZ-S), there was a slight improvement in recall with stop words (AZ-S).

```
Source      DF        SS        MS        F        P
ALG          2    8.7865    4.3932    81.34    0.000
Error      477   25.7623    0.0540
Total      479   34.5488

                                    Individual 95% CIs For Mean
                                    Based on Pooled StDev
Level        N      Mean     StDev  --------+---------+---------+--------
Chien      160    0.1915    0.1219   (--*--)
AZ         160    0.4345    0.2524                    (--*--)
AZ-S       160    0.5082    0.2889                          (--*--)
                                    --------+---------+---------+--------
Pooled StDev =    0.2324          0.24      0.36      0.48
```

Figure 6. Analysis of variance for recall.

```
Source      DF        SS        MS         F        P
ALG          2    3.4238    1.7119    50.81    0.000
Error      477   16.0707    0.0337
Total      479   19.4945
                                      Individual 95% CIs For Mean
                                      Based on Pooled StDev
Level        N      Mean     StDev  ----------+---------+---------+------
Chien      160    0.5214    0.1884  (---*----)
AZ         160    0.7003    0.1840                          (---*---)
AZ-S       160    0.7009    0.1781                          (---*---)
                                     ----------+---------+---------+------
Pooled StDev =    0.1836             0.560     0.630     0.700
```

Figure 7. Analysis of variance for precision

I. Discussion

First of all, we need to reiterate that the Chien's approach used in this experiment was our own implementation to our best optimization, as pointed out in the Experiment Design section. Chien reported average recall and precision as 0.43 and 0.30 for the experiment on book indexing, which was a test bed different from the journal abstracts used in this experiment. By comparing statistics that were not very meaningful because of different test beds, we found that our approaches still equaled Chien's in recall and surpassed his in precision.

In addition, the actual value of precision should be lower, because we constructed the answer set solely on the basis of the subjects' judgment, and the subjects were not allowed to add more phrases to the answer set, even if they felt it was necessary. However, this does not affect the relative comparison, especially when the margin is substantial, because a bigger answer set will move the values of precision in the same direction.

All three approaches are able to extract very long phrases, such as 策略性資訊系統規劃 (strategic information system planning) and 自然語言處理系統 (natural language processing systems), which illustrate the potential of a statistical approach to identifying long, new emerging terminology from corpus. Although the experiment showed encouraging results from our approaches, the experiment subjects would have liked the system also to have captured English names, phrases and acronyms, such as *Intel*, *neural networks,* and *ATM*, which are very useful key phrases, especially in the scientific literature.

All the programs were coded in C/C++ and finished processing in between 4 minutes for the non-updateable approach and 10 minutes for the updateable approach on a Pentium-II 350MHz with 256MB RAM. Therefore, these data structures and algorithms introduced were shown to be a speedy statistical approach to key phrase extraction.

J. Future Directions

The ability to generate key phrases is very important in the application of newer-generation knowledge management tools, because longer key phrases rather than the single character word are able to give more precise meaning and thus represent content more precisely.

1. Chinese SOM Category Map

A category map is the result of performing Self-Organizing Map (SOM), a neural network-based clustering, of similar documents and automatic category labeling. Documents that are similar (in key phrase) to each other are grouped together in a neighborhood on a two-dimensional display. Figure 8 shows a preliminary SOM category map generated by using the keywords extracted in this experiment. Each colored region represents a unique topic that contains similar documents. Topics that are more important often occupy larger regions. Examination of the category 專家系統 (expert systems) shows that the SOM actually relates it to other relevant categories such as 神經網 (neural networks) and 模糊 (fuzzy). By clicking on each region, a searcher can browse documents grouped in the region. Based on our recent experiments, we found such a 2D graphical display of hierarchical structure to be promising for aiding large-scale and dynamic user browsing and searching. The Kohonen SOM algorithm for classifying textual documents requires outputs from a good key phrase extraction process.

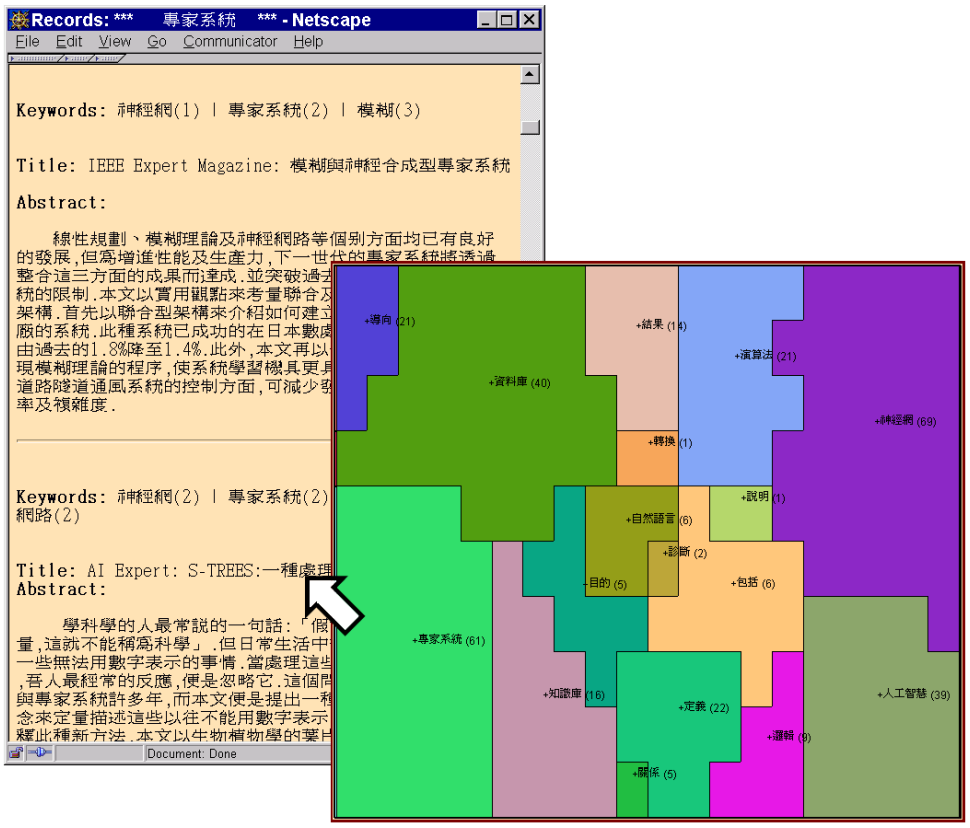

Figure 8. Extracted key phrases become features for the SOM, a neural network automatic content summarization technique. A user can pick a category and view its corresponding content.

2. Chinese CI Spider

The concept of using agents is an outgrowth of the past 40 years' research on artificial intelligence and robotics. It has become particularly relevant in the context of Internet research. The idea of a software entity that could perform tasks on behalf of a user was well established by the mid-1970s (Caglayan, 1997). The research aims to create software

to support reasoning, knowledge representation, and learning. Practical applications of agents have attracted significant attention in the 1990s. Many software agents have emerged to support Internet searches, data mining, and collaborative computing applications.

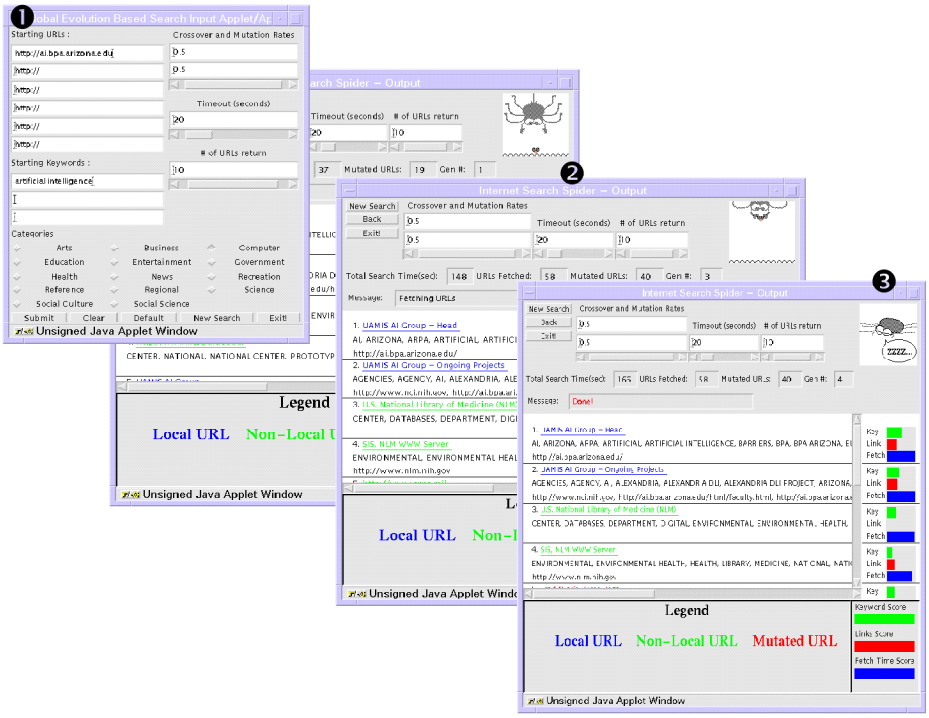

Figure 9. Itsy Bitsy Spider at work: (1) The user enters starting URLs. (2) The spider is collecting URLs that are similar to the starting URLs. (3) Spider "sleeps" after collecting web pages.

An agent-based spider for Internet searches was developed in a previous NSF-funded Internet project (Chen, Chung, et al., 1998). The Itsy Bitsy Spider explored the use of genetic algorithms for searching for Internet homepages based on search profiles supplied by users. Figure 9 (available at http://ai.bpa.arizona.edu) is a sample spider system. Submission of a search spawns agents, which then perform searches from the starting URLs and find new homepages that match the starting URLs and keywords. In addition, a user can limit the number of URLs returned, the search time for each new homepage, and can exercise other search engine parameters (e.g., crossover and mutation rates, to be discussed later). As a customized, cooperative agent, most parameters can be changed and re-used to affect search results, making the agent a truly dynamic, personalized Web search assistant. With the new Chinese key-phrase extraction technique we have just developed, we will be able to expand our spider to analyze Chinese web pages.

3. Extension to Other Oriental Languages

Chinese shares similar characteristics with many other oriental languages, especially Taiwanese, Japanese, and Korean. It is very likely that we can scale this technique across these languages. We hope to expand it in the near future.

K. Conclusions

We have presented our approach to extracting Chinese key phrases using an updateable PAT-tree. The experiment showed that the ability to update frequency after key phrases have been extracted is a valuable feature that increases the accuracy of subsequent extraction. This was validated empirically by the improvement in obtained recall from 0.19 to 0.43 and precision from 0.52 to 0.70. However, the effect of the use of stop words improved recall slightly but made no change in precision. On the other hand, the use of stop words resulted in a reduction of one third in the number of key phrases extracted.

This paper also reviewed the data structures required to implement any statistical approach efficiently, both in terms of space and time. A new data structure was introduced to add the ability to update frequency after key phrases had been extracted.

L. Acknowledgement

This research is sponsored in part by the NSF DLI2 (Digital Libraries Initiatives Phase 2, http://www.dli2.nsf.gov) grant #9817473. In addition, we greatly appreciate STIC (Science and Technology Information Center, http://www.stic.gov.tw) in Taiwan for providing the scientific journal abstracts for use in the experiment.

M. References

Baeza-Yates, R., & Gonnet, G. (1996). Fast Text Searching for Regular Expressions or Automaton Searching on Tries, *Journal of the ACM*, 43 (6), pp. 915-936.

Bian, G-W & Chen, H-H (1998). A New Hybrid Approach for Chinese-English Query Translation. *Proceedings of the First Asia Digital Library Workshop*, pp. 156-167.

Blair, D. C., and Maron, M. E. (1985). An evaluation of retrieval effectiveness for a full-text document-retrieval system. *Communications of the ACM*, 28(3), 289-299.

Brill, E. (1995). *Transformation-Based Error-Driven Learning and Natural Language Processing*. Computational Linguistics, 21 (4), 543-565.

Caglayan, A., Harrison., C. (1997). *Agent Sourcebook, A Complete Guide to Desktop, Internet, and Intranet Agents*.

Chen, A. et al. (1997). Chinese Text Retrieval without Using a Dictionary. *in Proceedings of the ACM SIGIR 97*, pp. 42-49.

Chen, H. (1998). The Illinois Digital Library Initiative Project: Federating Repositories and Semantic Research. *Proceedings of the First Asia Digital Library Workshop*, pp. 13-23.

Chen, H., Chung, Y., Ramsey, M., & Yang, C. (1998). "A Smart Itsy Bitsy Spider for the Web," *Journal of the American Society for Information Science*, 49 (7), Pages 604-618.

Chen, H., Houston, A. L., Sewell, R. R. & Schatz, B. R. (1998), "Internet Browsing and Searching: User Evaluation of Category Map and Concept Space Techniques," *Journal of the American Society for Information Science*, 49 (7), pp. 582-603.

Chen, H., J. Martinez, D. T. Ng, and B. R. Schatz. (1997). A Concept Space Approach to Addressing the Vocabulary Problem in Scientific Information Retrieval: An

Experiment on the Worm Community System. *Journal of the American Society for Information Science* 48 (1), pp. 17-31.

Chen, H., & Ng, D. T. (1995). An algorithmic approach to concept exploration in large knowledge network (automatic thesaurus consultation): Symbolic branch-and-bound vs. connectionist Hopfield net activation. *Journal of the American Society of Information Science*, 46(5), pp. 348-369.

Chen, Y. M., Liao, C. C., & Prasad, B. (1998). A systematic approach to virtual enterprising through knowledge management techniques. *Concurrent Engineering-Research and Applications*, 6(3), 225-244.

Chien, L-F and Pu, H-T (1996). Important issues on Chinese information retrieval. *Computational Linguistics and Chinese Language Processing, 1* (1), pp. 205-221.

Chien, L-F (1997). PAT-Tree-Based Keyword Extraction for Chinese Information Retrieval. *Proceedings of the 1997 ACM SIGIR*, Philadelphia, PA, USA, pp. 50-58.

Chien, L-F (1998). PAT-Tree-Based Adaptive Keyphrase Extraction for Intelligent Chinese Information Retrieval. in special issue on Information Retreival with Asian Languages, *Information Processing and Management*, Elsevier Press.

Church, K. (1988). *A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text.* Proceedings of the Second Annual Conference on Applied Natural Language Parsing ACL, Austin, TX.

Church, K. (1997). Ngrams. *Proceedings of the ACL-95,* Cambridge, MA, USA.

Davenport, T. H. (1995). Business process reengineering: Where it's been, where it's going. In V. Grover & W. Kettinger (Eds.), *Business Process Change: Reengineering Concepts, Methods and Technologies* (pp. 1-13). Middletown, PA: Idea Publishing.

Davenport, T. H., & Prusak, L. (1998). *Working Knowledge: How Organizations Manage What They Know*. Boston, MA: *Harvard Business School Press*.

Gartner Group, Summer Knowledge Management Workshop Report, Summer, 1998.

Gonnet, G. H. & Baeza-Yates, R. (1991*). Handbook of Algorithms and Data Structures in Pascal and C,* 2nd Ed.

Inkpen, A. C., & Dinur, A. (1998). Knowledge management processes and international joint ventures. *Organization Scienc*e, 9(4), 454-468.

Jones, P., & Jordan, J. (1998). Knowledge orientations and team effectiveness. International Journal of Technology Management, 16 (1-3), 152-161.

Kwok K. L. (1997). Comparing Representations in Chinese Information Retrieval. in *Proceedings of ACM SIGIR'97*, pp. 34-41

Knuth, D. E. (1973). *The Art of Computer Programming: Sorting and searching*, Vol. 3. Addison-Wesley, Mass.

Lesk, M. (1997). *Practical Digital Libraries*, Morgan Kauffmann, Los Altos, CA.

Li, Z. & Xing, L. (1998). Search the Chinese Web — Design and the Operation of Net-Compass. *Proceedings of the First Asia Digital Library Workshop*, pp. 42-46.

Lin, C. & Chen, H. (1996). *An Automatic Indexing and Neural Network Approach to Concept Retrieval and Classification of Multilingual (Chinese-English) Documents*. IEEE Transactions on Systems, Man, and Cybernetics, 26 (1), pp. 1-14.

Manber, U., & Myers, G. (1993). Suffix arrays: a new method for on-line string searches. SIAM-Journal-on-Computing, 22 (5), pp. 935-48.

Morrison, D. R. (1968). PATRICIA — Practical Algorithm to Retrieve Information Coded in Alphanumeric. *Journal of the Association for Computing Machinery*, 15 (4), pp. 514-534.

Nonaka, I. (1994). A dynamic thery of organizational knowledge creation. *Organization Science*, 5 (1), 14-37.

O'Leary, D. E. (1998). Enterprise knowledge management. *IEEE Computer*, 31 (3), 54-62.

Orwig, R., Chen, H., and Nunamaker, J. F. (1997). A graphical, self-organizing approach to classifying electronic meeting output. *Journal of the American Society for Information Science*, 48 (2), pp. 157-170.

Paepcke, A., S. B. Cousins, H. Garcia-Molino, S. W. Hasson, S. P. Ketcxhpel, M. Roscheisen, and T. Winograd (1996). Using distributed objects for digital library interoperability, *IEEE COMPUTER*, 29(5), pp. 61-69.

Rouse, W. B., Thomas, B. S., & Boff, K. R. (1998). Knowledge maps for knowledge mining: Application to R&D/technology manangement. *IEEE Transaction on Systems, Man and Cybernetics: Part C- Applications and Reviews*, 28(3), pp. 309-317.

Salton, G. (1989). *Automatic Text Processing*. Reading, Addison-Wesley,**(City?)** MA.

Schatz, B. & Chen, H. (1996). Building Large-Scale Digital Libraries, *IEEE Computers*, Special Issue on "Building Large-Scale Digital Libraries," 29 (5), pp. 22-27, May 1996.

Schatz, B. R. & Chen, H. (1999). Digital libraries: technological advancements and social impacts. *IEEE Computer*, 31(2), 45-50.

Sedgewick, R. (1998). *Algorithms, 3rd Ed.* Addison-Wesley.

Su, K-Y, Chaing, T-H, & Chang, J-S (1996). An Overview of Corpus-Based Statistics-Oriented (CBSO) Techniques for Natual Language Processing. *Computational Linguistics and Chinese Language Processing, 1* (1), pp. 101-157.

Teece, D. J. (1998). Research directions for knowledge management. *California Management Review*, 40(3), 289-292.

Voutilainen, A. (1997) *A Short Introduction to NPTool*, www.lingsoft.fi/doc/nptool/intro/

Wong, K-F & Li, W. (1998). Intelligent Chinese Information Retrieval — Why is it so Difficult? *Proceedings of the First Asia Digital Library Workshop*, pp. 47-56.

Wu Z. & Tseng G. (1993). "Chinese Text Segmentation for Text Retrieval: Achievements and Problems," Journal of the American Society for Information Sciences, 44, pp. 532-542.

Wu Z. & Tseng G. (1995). ACTS: An Automatic Chinese Text Segmentation System for Full Text Retrieval. *Journal of the American Society for Information Sciences*, 46, pp. 83- 96.

Yang, C. C., Yen, J., Yung, S. K., & Chung, K. L. (1998). Chinese Indexing using Mutual Information. *Proceedings of the First Asia Digital Library Workshop*, pp. 57-64.