Building a Chinese Text Summarizer with Phrasal Chunks and Domain Knowledge

Weiquan Liu & Joe Zhou

Intel China Research Center

Page 87 ~ 96

Proceedings of Research on Computational Linguistics

Conference XIII (ROCLING XIII)

Taipei, Taiwan

2000-08-24/2000-08-25

Building A Chinese Text Summarizer with Phrasal Chunks and Domain Knowledge

Weiquan Liu and Joe Zhou

{Lious.Liu; Joe .F.Zhou}@intel.com

Intel China Research Center

601 North Tower, Beijing Kerry Center

#1 Guanghua Road, Beijing 10002, China

Abstract

This paper introduces a Chinese summarizier called *ThemePicker*. Though the system incorporates both statistical and text analysis models, the statistical model plays a major role during the automated process. In addition to word segmentation and proper names identification, phrasal chunk extraction and content density calculation are based on a semantic network pre-constructed for a chosen domain. To improve the readability of the extracted sentences as auto-generated summary, a shallow parsing algorithm is used to eliminate the semantic redundancy.

1 Introduction

Due to the overwhelming amount of textual resources over Internet people find it increasingly difficult to grasp targeted information without any adjunctive tools. One of these tools is automatic summarization and abstraction. When coupled with general search and retrieval systems, text summarization can contribute to alleviating the effort in accessing these abundant information resources. It is capable of condensing the amount of original text, enabling the user to quickly capture the main theme of the text.

Based on the techniques employed (Hovy, 1998), existing summarization systems can be divided into three categories, i.e., word-frequency-based, cohesion-based, or information-extraction-based. Comparing to the other two techniques the first one is statistical oriented, fast and domain independent (Brandow *et al*, 1995). The quality, however, is often questionable. Cohesion-based techniques (or sometimes called as being linguistic oriented) can generate more fluent abstracts, but the sentence-by-sentence computation against the entire raw text is often quite expensive. Even the most advanced part of speech (POS) tagging or syntactic parsing algorithms are unable to handle all the language phenomena emerged from giga-bytes of naturally running text. Summarization based on information extraction relies on the predefined templates. It is domain dependent. The unpredictable textual content over Internet, however, may let the templates suffer from incompletion or intra-contradiction no matter how well they might be predefined.

In this paper we introduces a Chinese summarization system. Though it is a hybrid system incorporating some natural language techniques, considering the speed and efficiency of text processing we still adapted a statistical oriented algorithm and allowed it to play a major role during the automatic process. After pre-processing, the system first extracts phrasal chunks from the input. The phrasal chunks normally refer to meaningful terms and proper names existing in the text that are difficult to capture using simple methods. Then, we use a domain specific concept network to calculate the content density, i.e. measuring the significance score of each individual sentence. Finally, a Chinese dependency grammar applies as a shallow parser to process the extracted sentences into bracketed frames so as to achieve further binding and embellishment for the final output.

2 System Overview

The system, hereafter referred to as *ThemePicker*, works as a plug-in to web browsers. When surfing among some selected Chinese newspaper web sites, *ThemePicker* monitors the content of the browser’s window. When the number of domain words or terms exceeds a pre-defined threshold, it will kick off the summary generation process and display the output in a separate window. Currently, we chose economic news as our specific domain.

The system consists of four components (see *Fig. 1*). The first component is a pre-processor dealing with the layout of the news web pages and removing unnecessary HTML tags while keeping the headline, title and paragraph hierarchy. The retained information will provide the location of the extracted sentences for later manipulation.

The second component performs two tasks in parallel, resolving Chinese word segmentation and identifying and extracting phrasal chunks. As it is known to all, Chinese is an ideographical character based language with no spaces or delimiting symbols between adjacent words. After breaking the input sentence into a chain of separate character strings we use a lexical knowledge base to look up each word and parse the sentence appropriately. Person names and other proper names are also recognized during the segmentation process. Phrasal chunks are lexical units larger than words but not idioms. They are content oriented special terms (Zhou, 1999). We examined hundreds of documents and frequently encountered these phrasal chunks in the text that bear important information about the document. Since the meaning of a phrasal chunk is by no means the simple aggregation of the meanings of all the words in it, the word segmentation can not handle it. *ThemePicker* uses a statistical algorithm for phrasal chunk identification, aiming at the larger lexical unit that consists of two or more words always occurring in the same sequence.

The third component in sequence computes the degrees of sentence content density. The computation assigns a significance score to each sentence. The concept net that contains of more than 2000 concept nodes on economic news domain is used to define the semantic similarities between different sentences and adjust the significance scores of sentences across the input text. Sentences with high scores are selected for the inclusion in the candidate summary.

The fourth component analyzes the candidate sentences using a Chinese dependency grammar. The purpose is to improve the readability of the output summary.

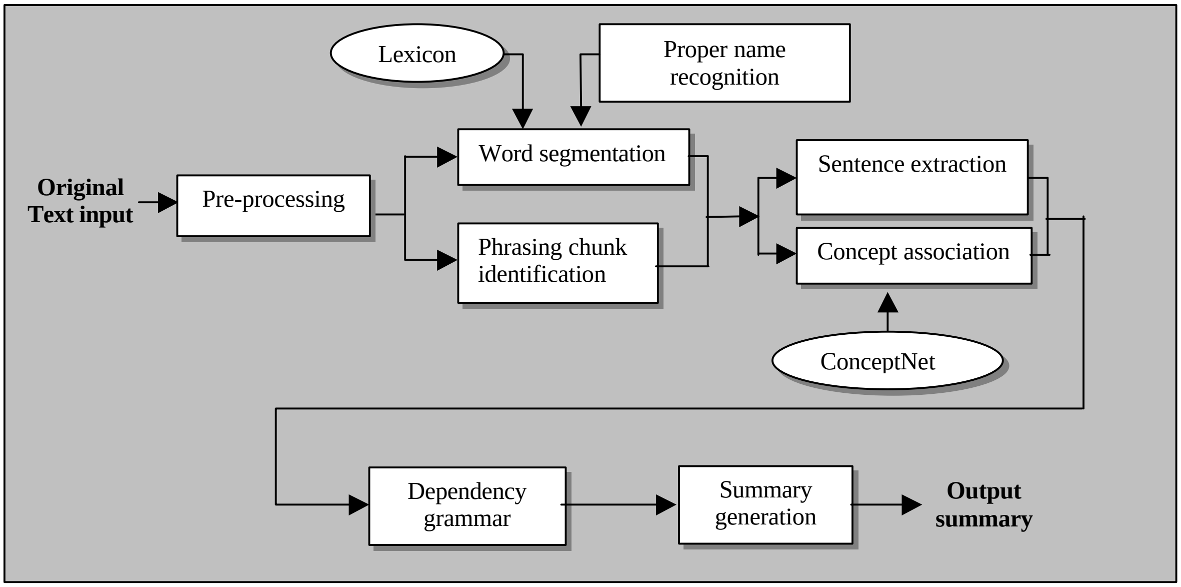

Figure 1: System overview and process flow

In the remaining sections of this paper we will describe in some details the major system components, i.e., word segmentation and proper name identification (Section 3), phrasal chunk extraction (Section 4), domain knowledge for summary generation (Section 5), and the dependency grammar (Section 6). The final section (Section 7) devotes to the system evaluation.

3 Word Segmentation and Proper Name Identification

The segmentation algorithm is a single scan Reverse Maximum Matching (RMM). One major difference from other RMMs is the special lexicon it uses. The lexicon consists of two parts, the indexing pointers and the main body of lexical entries (*see Fig 2*).

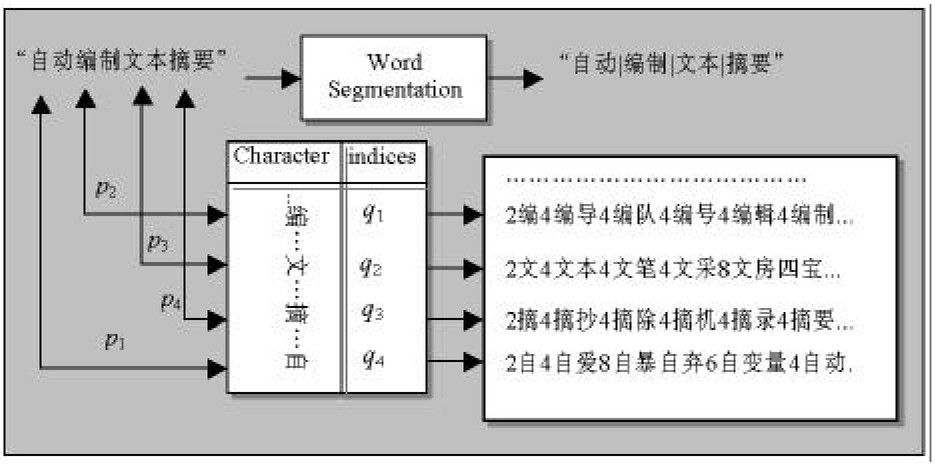
"自动编制文本摘要"
Word Segmentation
"自动|编制|文本|摘要"
Character
indices
p_2
p_3
p_4
p_1
编
文
摘
自
q_1
q_2
q_3
q_4
2编4编导4编队4编号4编辑4编制...
2文4文本4文笔4文采8文房四宝...
2摘4摘抄4摘除4摘机4摘录4摘要...
2自4自爱8自暴自弃6自变量4自动.

Figure 2: Lexicon structure and segmentation process

The algorithm works efficiently. The average number of comparisons needed to segment each word is only 2.89 (Liu *et al*, 1998). The unregistered single characters that are left behind the word segmentation will become the target of proper name recognition.

Proper names in Chinese carry no signals like capitalization, hyphenation, and interpunction in English, to indicate that they are special and different from other noun phrases. Our algorithm currently can handle two types of proper names, people names and organization names. People names include Chinese person names and names of foreign origin (though treated differently). The majority of organization names are company names due to the nature of the selected domain economic news.

To fulfil the task of recognizing Chinese person names we built a surname and a given name databases. Intuitively, any given Chinese person name is formed by a lead surname and followed by 1 or 2 given names. The surname has only one character and rarely has two, therefore the length of each person name ranges from 2 to 4 characters. In the surname and given name databases, each character is given a possibility value that is obtained by calculating its frequency over a large name bank. Our person name recognition algorithm works as follows.

When an unregistered single character word is encountered during the scan of the segmented text, the algorithm will check a) whether the character is a surname, and b) whether the character is followed by one or two single character words. If both conditions are met, these two to three consecutive character string may likely be a person name, denoted as $n=sc_1c_2$. (four-character names are temporarily omitted since they are rare). Here is the calculation of the possibility of n:

$$p(n) = \log p(s)\,p(c_1), \text{if there is a single given name, or}$$
$$p(n) = \log p(s)\,p(c_1)\,p(c_2), \text{if there are double given names.}$$

Thus, n is recognized as a Chinese person name for two character names if $\eta_1 < p(n) < \eta_2$, or for three character names if $\zeta_1 < p(n) < \zeta_2$. Here, η_1, η_2, ζ_1 and ζ_2 are pre-defined thresholds (Sun, 1998).

When calculating the possibilities, the title words, such as *Mr.*, *Mrs*. etc. that immediately before *n* and verbs that follow *n* are also considered heuristically.

The difference between Chinese person name and transliterated foreign name is that the latter uses only a limited set of characters. The number of characters that allow to be used to denote foreign origin names is about 400 to 500 (Sun, 1998). Within this set, a portion of it can only be used as the first character and another subset can only be the tail ones. Using this principle we defined a set of rules to label the margins of foreign names resulting in satisfactory precision and recall.

Company name identification is also statistical and heuristic in nature. Based on the observation and analysis of a large quantity of collected Chinese text, we concluded that most company names can be denoted by the following BNF:

<*Geographical Loc*> + [<*Ordinal Number*>] + {<*Product Name*>|<*Trade Name*>} + <*Appellative Noun*>

Thus, we built a FSM in which heuristic rules are introduced to allow the system capture such text strings as company names.

Our initial evaluation of some sample text databases indicates that approximately 3% of the original text are proper names of various kinds, among whom the above two categories constitute more than 95%. This means that we would lose 2.85% of the segmentation accuracy if no action were taken to handle these two names. The above procedure now achieves more than 96% in accuracy. The improvement to the segmentation is 2.74%.

As mentioned above, proper names denote critical information in the original document. Their incorporation can make the summary more informative. Improved segmentation helps identify domain words more accurately. The identification of proper names also benefits the shallow parsing and improves the coherence and cohesion of summary output. Though phrasal chunk identification is independent to the segmentation, it is character based not word based.

4 Phrasal Chunk Identification

The phrasal chunk identification algorithm is to locate new terms formed by two or more words that frequently occur in the input text. For the words "香港", "金融" and "改革" found in the input text, if their frequencies all exceed a pre-defined threshold, we can say that they are key words in the original text. But, this does not mean the whole phrasal chunk "香港金融改革" is also a key word. To determine such a long term or a phrase chunk is also a key word we have to prove that these three words or 6 characters frequently appear in exactly the same sequence.

Our phrasal chunk identification algorithm uses a data structure used called Association Tree (A-Tree). A unique A-Tree can be constructed for each individual character using itself as the root of the respective tree.

Fig.3 shows an example of A-Trees. Each node consists of a character and an associated integer shows in parentheses. The integer refers to the number of occurrences of the character in the input text. The integers associated with other child nodes denote the number of occurrences that particular character follows its parent node. An A-Tree is constructed in the following way:

- Scan the input and record the position of each individual character $\boldsymbol{C}$. Define $\psi = \{\boldsymbol{C_i} \mid \boldsymbol{C_i} \in \Sigma\}$ as the set of all possible characters found in the input. $|\boldsymbol{C_i}|$ is the number of occurrence of $\boldsymbol{C_i}$. Delete all $\boldsymbol{C_i}$ when $|\boldsymbol{C_i}| < T$ with T as a predefined threshold

- For each remaining individual character $\boldsymbol{C_i} \in \psi$, create a A-tree and place $\boldsymbol{C_i}$ ***(n)*** at the root of the tree and ***n*** as the associated integer

- Add all the descendants of $\boldsymbol{C_i}$ to the leaf node set $\varphi = \{\boldsymbol{d_j} \mid \boldsymbol{d_j} \in \Sigma\}$. Delete those $\boldsymbol{d_j}$ where $|\boldsymbol{d_i}| < T$ with T as a predefined threshold

- For each node $\boldsymbol{d_j}$ in φ, add its descendant characters as described in step 3 and remove $\boldsymbol{d_j}$ after it gets expanded

- Repeat step 4 until no leaf can be expanded, then the A-Tree of $\boldsymbol{C_i}$ is complete.

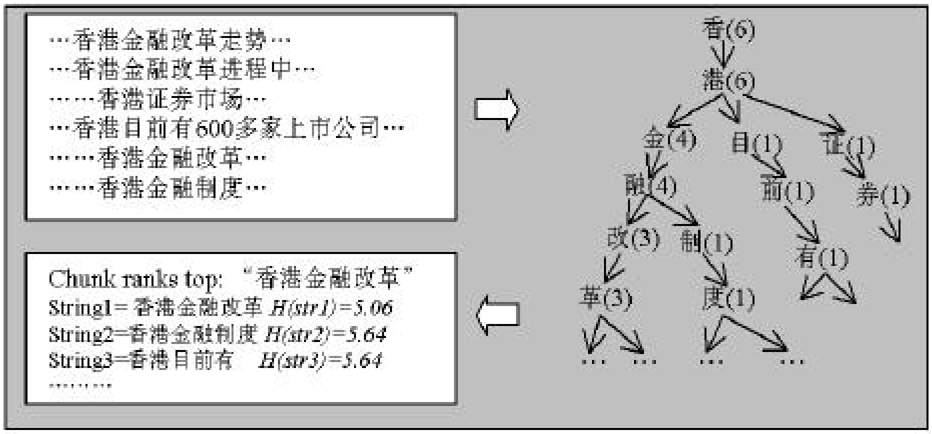

Figure 3: Phrasal chunk identification and an A-Tree

Once all A-Trees are constructed, new phrasal chunks can be extracted using entropy measurement. By tracking from the root node to each leaf node we can get a string of characters. For example, given a string $a_1a_2\cdots a_n b_1 b_2\cdots b_m$ that denotes two sub-strings A=$a_1a_2\cdots a_n$ and B=$b_1b_2\cdots b_m$ with a_1 as the root, the entropy in B given A is: $H(B \mid A) = -\log p(B \mid A)$.

For an A-Tree the ratio $|b_m| / |a_n|$ is an estimation of $p(B|A)$. The smaller the H value the closer the relationship between these two sub-strings. A zero value means B always follows A, suggesting that AB is a meaningful phrasal chunk.

For a string G=$C_0C_1C_2\cdots C_n$, the entropy in C_1 given C_0 is H_{C1}= -log $P(C_1|C_0)$. Given C_0C_1, entropy in C_2 is H_{C2}= - log $p(C_2|C_0C_1)$. Thus, the total entropy measurement of G is defined as:

$$H_\Gamma = \sum_{i=0}^{n} H_{Ci} = -\log p(C_0 ... C_n), \quad \text{where } H_{C0} = -\log p(C_0)$$

As shown in Fig. 3 there are three phrasal chunks that have been listed with their respective H values with the first one bearing the lowest. The chunk identification algorithm will collect all the phrasal chunks with H value less than a certain threshold among all the A-Trees built from the input text. These phrasal chunks are larger than a word and likely express the key content of the input.

5 Sentence Extraction Using Domain Knowledge

The significance score of a sentence is determined based on the sum of two measurements, the density of domain concepts and the density of phrasal chunks.

Suppose a sentence denoted as $S=U_1U_2U_3\cdots U_L$, $U_i \in [F \mid W \mid K]$, $1<i<L$ (here F: function words, W: domain concept words and K: phrasal chunks), for those U_i that belong to F, no contribution will be made to the significance score. For other U_i that belong to W, their contribution to the significance score is gained from the domain knowledge contained in a *ConceptNet*. The *ConceptNet* is a graphic network constructed semi-automatically with nodes as various concepts and arcs as relations between concepts. The current version of our *ConceptNet* contains more than 2,000 nodes all collected from a large economic news database (see Fig. 4). The relations between concepts are of several types, such as a-kind-of, a-part-of, abbreviation-of, product-of, member-of, etc. The density of domain concepts α_w is calculated as follows:

$$a_w = \sum_{U_i \in W} g_i (1 - \sum_{U_j \in W} R(U_i, U_j)) / |U|, \; g_i \text{ is a heuristic coefficient.}$$

$R(w_1,w_2)$ is a function that determines the semantic relations between w_1 and w_2.

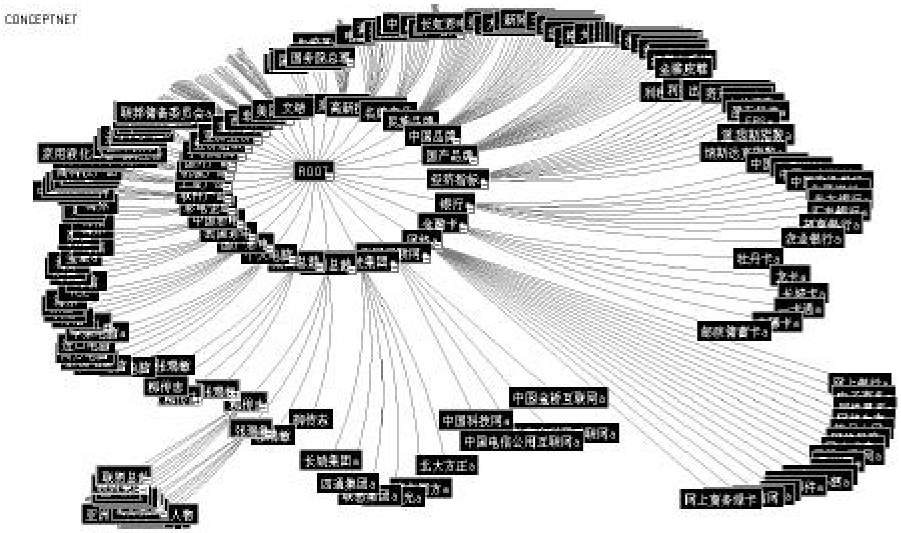

Figure 4: A partial snapshot of *ConceptNet* for economic news domain

For those U_i that belong to K, their contribution to the significance score is calculated as

$a_K = \sum_{U_i \in K} g_i H(U_i) / |U|$ (referring to the previous section on the calculation of $H(U_i)$, the entropy of U_i.). Thus, the final significance score for the sentence S is:

$$a_S = l_S(b_1 a_W + b_2(1 - a_T)).$$

Conceptually, we give special treatment to domain concept words and phrasal chunks that appear in the title and headline. Some cue words or phrases are also detected that may bring positive or negative contributions to the significance score depending on their properties. b_1 and b_2 are balance factors for a_w and a_K. ls is determined by the location of S in the paragraph.

After all the input sentences receive the significance scores, those having values greater than a pre-defined threshold are chosen for the possible inclusion in the generated summary. The default length of the output summary is within 10~20% of the original text.

6 Dependency Grammar

Though they receive higher significance scores, the extracted sentences cannot be treated as the abstract of the original text. The readability is low even if they are strung together in the order as they occur in the input. The duplication in meaning and the appearance of improper conjunction words often make readers confused. Anaphora without contextual reference also poses difficulty in comprehension.

To bind and embellish the output summary we employed a Chinese dependency grammar to parse the extracted sentence into Dependency Relation Tree (DRT). Based on the methodology introduced in Liu *et al*, 1998, DRT can further be bracketed into cells. One of the cells is called the core with others being dominated by the core. There exist unique mappings between dependency relations in DRT and the dominating relations among cells. Fig. 5 illustrates such an example.

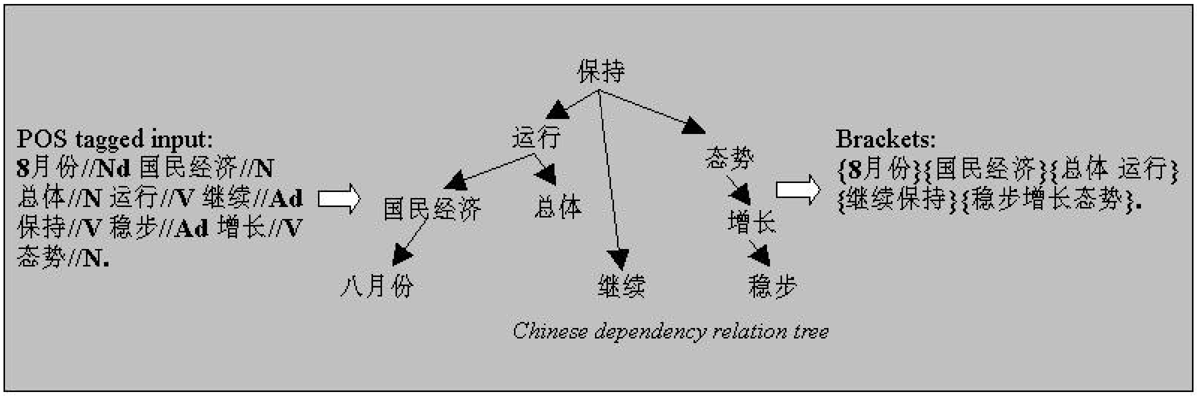
POS tagged input:
8月份//Nd 国民经济//N 总体//N 运行//V 继续//Ad 保持//V 稳步//Ad 增长//V 态势//N.
保持
运行
态势
国民经济
总体
增长
八月份
继续
稳步
Chinese dependency relation tree
Brackets:
{8月份}{国民经济}{总体 运行}{继续保持}{稳步增长态势}.

Figure 5: A sample sentence and its DRT

To eliminate the redundancy between two extracted sentences, we defined a semantic distance between them. Suppose that the bracketed cells of sentence *S* are represented as: $Core(S):=[slot_1(S), slot_2(S),\cdots slot_n(S)]$, then we can define the semantic distance between S_1 and S_2 as $D(S_1, S_2)$:

$$D(S_1, S_2) = \sum_i diff\,(slot_i(S_1), slot_i(S_2))$$

If $Core(S_1)$ and $Core(S_2)$ are different, $D(S_1, S_2)$ is indefinite. If $Core(S_1)$ and $Core(S_2)$ are the same, $diff(\cdot)$ is used to denote the semantic similarities between $slot_i(S_1)$ and $slot_i(S_2)$. The more similar the contents in the two slots, the smaller the value of $diff(\cdot)$, thus the smaller the distance $D(S_1, S_2)$.

A special case of the semantic distance is $D(S_1, S_2)=0$, that means S_1 and S_2 are basically identical in meaning, so one of them can be deleted. In most cases, $D(S_1, S_2)$ is greater than zero. A distance threshold is pre-defined in order to determine which extracted sentence can be eliminated. After the redundancy elimination the remaining portion of extracted sentences is reorganized to assemble the final output summary.

7 Performance Evaluation

In this paper we introduced a Chinese summarizier called *ThemePicker*. It is a hybrid system incorporating both statistical and text analysis models. For the sake of speed and efficiency, the algorithm was implemented in a way that allows the statistical model to take the major role during the automated process. We built a semantic network (ConceptNet), a knowledge base that contains more than 2000 concept nodes with arcs indicating the conceptual relationships between or across nodes. Our experiments have showed that the content density measured based on ConceptNet can be more valid than an algorithm purely based on key terms. To achieve higher degrees of readability of the auto-generated summary, we adapted a shallow parsing algorithm to eliminate the semantic redundancy between the extracted sentences. While enhancing the summary cohesion and coherence, the computational overhead is restricted.

As pointed out in the literature, due to the lack of the evaluation standards for auto summaries, it remains to be an open research topic regarding how to compare the performance of a text summarizer with any concrete and solid measurement (Paice, 1990). We conducted a preliminary system evaluation against the database that contains 2800 news articles (2.4M words in total) on the economic domain. First, two human analysts manually screened 1200 articles and identifies 80 specific topics like *Euro, Fortune Forum, RMB won't be depreciated,* etc. Then, they manually generated summaries for several selected documents from each of the 40 topics. After that, they compared the automatically generated summaries with those they manually composed. The benchmark uses three grading scales, comparing to the manually generated summary the auto counterpart was assigned as either, *good* or *acceptable* or *non-acceptable*. The results indicated that the total documents that received either good or acceptable grades constitute more than two-thirds of the total documents evaluated. Evaluation using more rigid methodology will be performed in the future.

8 References

(Brandow *et al*, 1995) Brandow R. Mitze K. and Rau L F. *Automatic Condensation of Electronic Publication by Sentence Selection.* Information Processing & Management, 31(5): 675-68, 1995

(Cohen, 1995) Cohen J D. *Highlights: Language and Domain Independent Automatic Indexing Terms for Abstracting*. Journal of the American Society for Information Science, 46(3): 162-174, 1995

(Hovy, 1998) Hovy E. and Marcu D. *Automatic Text Summarization*. Tutorial of CONLING/ACL'98. 1998

(Liu *et al*, 1998) Liu W. Wang M. and Zhong Y. *Implementation of a Field Non-specific Hybrid Automatic Abstracting System*, in the Proceedings of 2nd Intl Conf on Information Infrastructure (ICOII’98), Beijing pp275-278

(Paice, 1990) Paice C D. *Constructing Literature Abstracts by Computer: Techniques and Prospects*. Information Processing & Management, 26(1):171-186, 1990

(Sun, 1998) Sun, M S *et al*. *Identifying Chinese names in Unrestricted texts*. Journal of Chinese Information Processing 9(2):16-27, 1998

(Zhou, 1999) Zhou F J. *Phrasal Terms in Real-world IR Applications*. In Strzalkowski T. *eds* Natural Language Information Retrieval, pp215-260. Kluwer Academic Publishers, 1999