Study of Automatic Abstracting Based on Corpus and Hierarchical Dictionary*

SONG Jin ZHAO Dong-yan

(Department of Computer Science and Technology Beijing University Beijing 100871)
E-mail: songjin@ailab.pku.edu.cn

Abstract The study of automatic abstracting is a vital and practical information processing task in natural language processing, and becomes an important problem in domains such as Internet information retrieval. An approach based on corpus proposed by this paper provides an integration of the advantages of linguistic analysis based methods and those based on statistics. In essence, the basic idea of corpus-based method is at the expense of the cost of analysis outside the system to gain the efficiency of the algorithm inside the system. The algorithm given by the paper implements both keywording and abstracting while the former is based on a hierarchical dictionary and the latter on the corpus.

Key words Automatic abstracting, corpus, keywording, hierarchical dictionary.

The study of Automatic Abstracting is a practical but difficult branch in Natural Language Processing, and becomes an important problem in domains such as Internet information retrieval. Its aim is to cover the full 'source text' of a document and generate a brief and hopefully intelligible statement from it[1].

In fact, the research of automatic abstracting began in as early as 1950s. The first experiment on automatic abstracting was reported in a paper by H.P. Luhn published in 1958[2]. All subsequent work reported up to about 1970 was based on Luhn's ideas and relied mainly on the word-frequency methods[3,4] while the later work was based on the clue phrase methods[5,6] and methods using domain-knowledge[7~9] or based on natural language understanding[10].

Obviously, abstracting methods cannot rely simply on gross statistical evidence, and it is gradually desired that they can take into account the syntactic and semantic characteristics of the language and the text. Unfortunately, abstracting method based on linguistic analysis is usually time-consuming and then not suitable to be used by practical systems at present.

The method based on corpus proposed by this paper provides an integration of the advantages of the methods based on linguistic analysis and those based on statistics. First, the processing of the corpus is separated from the run time of the system, and then the syntactic and semantic analysis can be conducted deeply without impairing the performance of the system. Secondly, the statistical evidence extracted from the analysis of the corpus supports the succinctness and efficiency of the abstracting algorithm. Thirdly, Natural Language Processing Engineering including automatic abstracting needs the support from the analysis of realistic language

* **SONG Jin** was born in 1972. She is a lecturer of Department of Computer Science and Technology, Beijing University. She received a Ph.D. degree from the Institute of Computing Technology, the Chinese Academy of Sciences in 1998. Her current research interests are artificial intelligence, multi-agent systems, and natural language processing. **ZHAO Dong-yan** was born in 1969. He is a Ph.D. candidate at Institute of Computer, Beijing University. His research interests are electronic publishing systems, multi-agent systems and computer network.

Manuscript received 1997-12-24, accepted 1998-10-06.

material.

In essence, the basic idea of corpus-based abstracting method is at the expense of the cost of analysis outside the system to gain the efficiency of the algorithm inside the system.

1 Overview of Automatic Abstracting

1.1 Extraction-based abstracting

Research of automatic abstracting up to about 1970 was concerned with methods for producing extracts, that is, sets of sentences selected to provide a good indication of the subject of the document. The general approach is to examine each sentence, looking for clues to its importance; to compute a score for the sentence based on the clues found; and then either to select all sentences whose score exceeds some threshold, or to select the highest scoring sentences, up to a certain total. The sentences are then printed in their order of occurrence in the original text.

Many distinct types of clue to sentence significance have been tried, for example, the frequency of keywords, the title keywords, the location of sentences, syntactic criteria, the cue in the sentences, the relation criteria, and so on[11].

Reliable automatic abstracting methods, however, must take into account the syntactic and semantic characteristics of the language and the text. They can not rely simply on gross clue evidence of sentences.

1.2 Artificial intelligence based abstracting

Artificial intelligence based or linguistic approach performs a full syntactic (even semantic) analysis (a parse) in order to join the individual words of a sentence into phrases, phrases into clauses, and so on. In principle, a compact representation of a document could be achieved using a combination of syntactic and semantic criteria.

A full analysis generally requires extensive or detailed semantic knowledge to be handled explicitly. For instance, DeJong's FRUMP system[5] analyses news articles by instantiating slots in one frame of a set of predefined frames. When the analysis is completed, a script is used to generate a summary of the information held in the relevant frame. In Rau's SCISOR system[9], detailed linguistic analysis of a text (or indeed of several interrelated texts) results in the construction of a semantic graph, which is convenient for intermediate storage. A natural language generator may then produce summaries from the stored material.

It must be understood that these abstracting systems are only capable of processing texts within a very narrow domain, whose characteristics are predictable and well understood.

2 Design of the Algorithm

The algorithm given by the paper implements both keywording and abstracting while the former is based on a hierarchical dictionary and the latter on the corpus.

2.1 Keywording

The basic method of keywording in our system is using a hierarchical dictionary to classify the document T within a domain D, and then selecting the key words of the document depending on their importance to the domain D and the document T.

2.1.1 Hierarchical dictionary

In the hierarchical dictionary used in our method, the classification of a word w within all the domains D_1, $D_2,\ldots,D_n$ is described as follows:

$$w:\{(D_1,g_1),(D_2,g_2),\ldots,(D_n,g_n)\}, \tag{1}$$

where D_i denoting a domain is a real number whose integer part denotes the super hierarchy and decimal part denotes the sub-hierarchy. For example, the domain computer network is 11.12 where 11 represents the domain computer. g_i denotes the belief of the classification of w within the corresponding domain and

$$g_i = \frac{F_w(D_i)}{F(w)}, \tag{2}$$

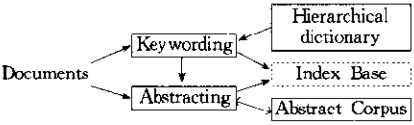

Fig. 1 General structure of the algorithm

where $F_w(D_i)$ is the frequency of w within the documents of domain D_i, and $F(w)$ is the frequency of w within all documents.

Or g_i can be determined by the occurrence of w in corresponding dictionary D_i:

$$g_i = \begin{cases} 0, & \text{if } w \text{ doesn't occur in the dic. } D_i; \\ 1, & \text{if } w \text{ occurs in the dic. } D_i \text{ only}; \\ \alpha, & \text{if } w \text{ occurs occurs not only in the dic. } D_i. \end{cases} \tag{3}$$

Here $0 < \alpha \leqslant 1$.

2.1.2 Classifying the document

Assuming the document T is a set of words occurring within it, that is

$$T = \{w_1, w_2, \ldots, w_m\}, \tag{4}$$

then the classification of T into domain D_j is determined by the classification of w_i into D_j, and is described by G_j:

$$G_j = \frac{\sum_{i=1}^{m}(g_{w_i}(D_j) \times N(w_i))}{\sum_{i=1}^{m} N(w_i)}, \tag{5}$$

where $N(w_i)$ is the number of occurrences of word w_i within the document T.

2.1.3 Weighing the words

As Luhn stated in one of his early papers[3], the frequency of word occurrence in an article furnishes a useful measurement of word significance. It is further proposed that frequency data can be used to extract words to represent a document. In our method, the information of domain is also considered besides the frequency to determine the weight of words.

Therefore if the domain of the document is D, the weight of word $w_i (0 < i \leqslant m)$ is described by $Wg(w_i)$:

$$Wg(w_i) = g_{w_i}(D) \times N(w_i), \tag{6}$$

where $g_{w_i}(D)$ is the classification of w_i in D.

Finally select the words with n-highest values of $Wg(w_i)$ as the keywords of T.

2.2 Abstracting

There are three main steps in abstracting: ① analyzing materials in the corpus and extracting the statistic information, ② determining the weights of sentences in the document, and ③ generating the abstract. The process is shown in Fig. 2.

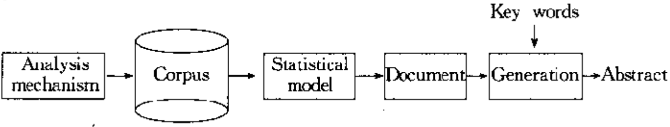

Fig. 2 Corpus-based abstracting

In fact, step one is separated from the algorithm and is included here to describe the algorithm clearly.

2.2.1 Analysis of materials in the corpus

The corpus used includes 2 000 abstracts (and corresponding documents) of technique reports or papers, totally about 280 000 words, and 140 words per abstract. The analysis of the corpus consists of three steps:

(1) Extracting 'abstract words' (depending on their occurrence frequency mainly). Abstract words including abstract verb, noun, adjective, adverb and phrase are those having nothing to do with the specific subject of the document, but commonly accompanying explicit statements about the subject.

Table 1 Occurrence times of several common abstract words

Abstract words	Attribute	Occurrence times
describe	verb	596
include	verb	369
address	verb	94
goal	noun	110
approach	noun	207
effort	noun	86
in detail	phrase	24
based on	phrase	222
be compared with	phrase	64

Abstract verb: There are about 100 abstract verbs, including some verb phrases such as account for, attempt to, consist of and so on. The verbs with high frequency include describe, include, show, present, introduce, review, propose etc.

Abstract noun: These nouns consist of paper, report, thesis, project, effort, etc. which denote the document itself, and purpose, goal, result, feature, advantage, etc. which denote the features of the document.

Abstract adjective (adverb): New, major, important, brief, and rapidly, particularly, significantly and so on.

Abstract phrase: For example, in contrast to, in order to, in detail, in particular etc.

(2) Forming 'abstract items'. This step is changing abstract words depending on their syntactic and semantic characteristics and then forming the corresponding abstract items. For example,

① abstract word 'include' must be changed to abstract items 'include (s)(d)', 'be included' and 'including' for these different items have different weights;

② abstract word 'give' is changed to 'be given' only and deleted with 'giving', 'gave' etc. for the latter ones are semantically less important to the abstract than the former one;

③ abstract word 'describe' directly forms the item 'describe' without any change for any forms of 'describe' are semantically important to the abstracts.

(3) Determining the weight of the abstract items.

Let the weight of abstract item w be W, then

$$W=\frac{N_w(A)}{N_w(D)}, \tag{7}$$

where $N_w(A)$ and $N_w(D)$ respectively denote the occurrence times of w in all abstracts and all documents.

Several common abstract items are listed in Table 2 to show that the weight of an item is decided by its occurrence times in both the abstracts and documents.

Table 2 Comparison of weights of several common abstract items

Abstract items	$N_w(A)$	$N_w(D)$	Weight of items
describe#	596	655	0.91
be given	287	338	0.85
be included	63	80	0.79
include (*s*)(*d*)	199	398	0.50
including	107	497	0.21
be presented	598	665	0.90
*propose**	180	202	0.89

In Table 2, $verb^* = \{verb, verb\text{-}ing, verb\text{-}s, verb\text{-}ed, verb\text{-}en\}$, $verb^\# = verb^* - \bigcup\{verb\text{—}noun\}$. For example, propose* includes propose, proposed, proposes and proposing while describe# includes describe, describes, described, describing and description.

2.2.2 Weight determination for sentences

The aim of determining the weight of sentences is to select the 'key sentences' from the document. The key sentences can be defined as the sentences reflecting the main subject of the document, or with a close relation to the main content of the document. The former ones usually include abstract items while the latter ones include keywords of the document.

Therefore, the weight *WS* of a sentence s_i in the document *T* can be described as follows:

$$WS(s_i)=\sum_{i=1}^{m}W(w_i)+\sum_{j=1}^{n}W_g(k_j), \tag{8}$$

where $W(w_i)$ and $Wg(k_j)$ denote respectively the weight of abstract item w_i and that of key word k_j in the sentence.

Based on above calculation, a key sentence is defined as the sentence whose weight is not 0.

In fact, the location of the sentence also influences its weight. In view of the essential difference between the location weight and the word weight of a sentence, the influence of location over the weight of the sentence is reflected within the process of abstract generation.

2.2.3 Generation of the abstract

Generally speaking, the sentences within text parts like 'introduction' or 'conclusion' are usually more important to the subject of the document than others. And the first or the last sentence within a paragraph is usually the most central to the theme of the document. In our system, the information of the relative location of the sentence is reflected in the generation of the abstract.

First, if the document includes the parts like 'introduction' or 'conclusion', then select the sentence with the highest weight in turn within these parts until the length of the abstract *L* is reached.

Else calculate the average *p* of the weights of all key sentences, and select the first or the last sentence within every paragraph with a weight more than *p* until the length of the abstract *L* is reached. If the length of the result abstract *LA* is still less than *L*, then select the *L-LA* key sentences with highest weights.

All selected sentences are then printed in their order of occurrence in the original document. The length of the abstract is decided by the compression ratio and specified by the system or the user.

2.3 Algorithm

The algorithm for abstracting (including keywording) is given in Table 3.

Table 3 Algorithm for keywording and abstracting

```
    input: the document $T$, the length of the abstract $L$;
    output: the set of keywords $Keywords$, the abstract $Abs$;
1.  for every domain $D_i$, calculating the classification $G_i$ of $T$ in $D_i$;
2.  for two highest values $G_i$ and $G_j$, if $G_i \gg G_j$, then $Domain\ (T):=D_i$;
3.  else if $|D_i|=|D_j|$ then $Domain\ (T):=|D_i|$;
4.  else $Domain\ (T):=\{D_i, D_j\}$;
5.  for every word $w_i$, calculate the weight $Wg(w_i)$ depending on the $Domain\ (T)$;
6.  $Keywords\ :=\{w_i \mid Wg(w_i)>threshold\}$;
7.  if $T$ includes the text part '$introduction$' or '$conclusion$', then $Scope:=\{introduction, conclusion\}$,
    for every sentence $s_i$ in $Scope$, calculate the weight $WS(s_i)$, and go to (12);
8.  else $Scope\ :=T$, for every sentence $s_i$ in $Scope$, calculate the weight $WS(s_i)$;
9.  for key sentences $ks_1, ks_2, \ldots, ks_p$, $(WS(ks_i)\neq 0,\ 1\leqslant i\leqslant p)$, calculate the average $Ave$ of $ks_i (1\leqslant i\leqslant p)$;
10. for every sentence $ls_i$ being the first or the last one of paragraphs; if $WS(ls_i)>Ave$, then $Abs:=Abs\cup\{ls_i\}$,
    $Scope:=Scope-Abs$; if $Length\ (Abs)<L$ and $Scope\neq\varphi$, then go to (11), else go to (12);
11. $Abs:=Abs\cup\{s_i \mid WS(s_i)=Max(WS(s_i)),\ s_i\in Scope\}$, $Scope:=Scope-\{s_i\}$;
    if $Length(Abs)<L$ and $Scope\neq\varphi$, then go to (11), else go to (12);
12. for all sentences in $Abs$, print them out in their order of occurrence in $T$.
```

3 Conclusions

In fact, the automatic abstracting system described by the paper is one part of the system MIIRPS whose main aim is recognizing, classifying, abstracting, translting and managing the Internet information. Within MIIRPS, the Information Recognition System transfers the HTML files into pure texts, so the abstracting system is concerned with the pure text (sometimes including the location) information only.

The algorithm given in the paper implements both keywording and abstracting while the former is based on a hierarchical dictionary and the latter on the corpus. It is hoped that the work described here can contribute a little to the research of Automatic Abstracting, Internet IR, Database Managing, and Library Digitizing.

References

1 Paice C D. Extracting the essence. ILASH SEMINARS, 1996.1 http://www.dcs.shef.ac.uk/research/ilash/seminars/paice.html

2 Luhn H P. An experiment in auto-abstracting. In: International Conference on Scientific Information. Washington D.C., 1958

3 Luhn H P. The automatic creation of literature abstracts. In: Schultz ed. H. P. Luhn: Pioneer of Information Science. Spartan, 1968

4 Skorochodko E F. Adaptive method of automatic abstracting and indexing. Information Processing 71, 1971. 1179～1182

5 Paice C D. The automatic generation of literature abstracts: an approach based on the identification of self-indicating phrases. In: Oddy R N, Robertson S E, Rijsbergen C J eds. Information Retrieval Research. Butterworths, 1981

6 Rush J E, Salvador R, Zamora A. Automatic abstracting and indexing production of indicative abstracts by application of contextual inference and syntactic coherence criteria. Journal of the American Society for Information Science, 1971,7:260～274

7 Tait J I. Automatic summarizing of English texts[Ph.D. Thesis]. Computer Laboratory, University of Cambridge, 1983

8 DeJong G F. An overview of the FRUMP system. In: Lehnert and Ringle eds. Strategies for Natural Language Processing. Erlbaum, 1982

9 Rau L F. Conceptual information extraction and retrieval from natural language input. User-oriented Content-based Text and Image Handling. Proceedings of RIAO'88 Conference. MIT, 1988

10　Wang Jian-bo, Du Chun-ling, Wang Kai-zhu. Study of automatic abstraction system based on natural language understanding. Journal of Chinese Information Processing, 1995,9(3):33～42

11　Paice C D. Constructing literature abstracts by computer: techniques and prospects. Information Processing and Management, 1990,26(1):171～186

基于语料库与层次词典的自动文摘研究

宋今　赵东岩

(北京大学计算机科学与技术系　北京　100871)

摘要　自动文摘研究作为自然语言处理研究的一个重要且实用的分支,目前逐渐成为Internet信息检索等应用领域的重要研究课题之一.该文提出的基于语料库的文摘试图将传统的基地语言学分析的文摘方法和基于统计的文摘方法的优点结合在一起.基于语料库的文摘方法的实质即以系统外的分析代价换取系统内的算法效率.该文描述的算法给出了基于层次词典的关键字提取和基于语料库的自动文摘的实现.

关键词　自动文摘,语料库,关键字提取,层次词典.

中图法分类号　TP391