The Application of Natural Language Processing in Compiler Principle System

[1]Yujia Zhai, [1]Lizhen Liu*, [1]Wei Song, [1]Chao Du, [2]Xinlei Zhao
[1]Information and Engineering College, Capital Normal University, Beijing, 100048,P.R.China
[2]College of Foreign Languages, Capital Normal University, Beijing, 100048,P.R.China
Email: zhai_yj1102@163.com, lzliu_cnu@sina.com

Abstract—**Compiling principle is an important course of computer science major, which mainly introduces general principles and basic methods of the construction of compiling programs mainly. Due to high demands of the logic analysis ability, the course bring abstract and unintelligible experience to many students. Thus it is quite difficult for students to master the main points of this course within the limited class time. Based on the requirement above, this paper mainly proposed a method of making use of natural language processing in the research and application of compiling process, which utilizes Maximum Probability Word Segmentation algorithm during the process of lexical analysis and syntax analysis, to offer more effective interface between human and computer. The proposed method can provide students with intuitive and profound knowledge concept in the process of learning how to compile, makes it easier and quicker for students to understand the principle of computer compiling.**

Keywords—natural language processing; compiling Principles; Maximum Probability Word Segmentation algorithm

I. INTRODUCTION

The course of Compiling Principle consists of lexical analysis, syntax analysis, intermediate code generation and semantic analysis. This course is of high abstraction degree, which involves many concepts and methods that are quite unfamiliar to students. And the complex process of lexical analysis and syntax analysis requires high logic analysis ability. What's more, the difficulty of the course's experiment lessons require students' deep insight and analysis of the process of the compiling, thus makes it easier for students to separate theory from practice [1]. The application of the Maximum Probabilistic Segmentation algorithm for the compilation process presented in this paper can be used to further understand the compilation process from different aspects to realize the personalized learning [2] of the students.

Natural language processing is an important branch [3,4] in the domain of computer science and artificial intelligence, which is an integration of linguistics, computer science and mathematics. The main purpose is to develop an effective computer system, especially software system, which can achieve interaction between human and computer using natural language. Interaction with computers using natural language is long-term pursuit of human-beings. It is difficult for computers to understand natural language, which is mainly caused by the difficulty of eliminating the ambiguity that exists at all levels of texts or dialogues of natural languages. In consideration of the course of compiling principle, the task of lexical analysis is to scan the codes which is made up of strings from left to right, decompose them into different parts, recognize the grammatical components with independent meaning one by one according to some rules, and then transmit the results to the process syntax analysis. In order to eliminate the ambiguity in the process of analysis, a lot of knowledge and reasoning are required.

In this paper, we proposed a method that combines the lexical analysis and syntactic analysis [5,6] of the Compilation Principle Course with the Maximum Probability Word Segmentation Algorithm to solve the problem of ambiguity in the word segmentation process, which also gained a lift in the accuracy of word segmentation. In our method, the related parameters can be customized by students. Compared with other word segmentation methods, even some sentences from daily communication can still be classified with higher accuracy. During the process of application, the scale of the training set can be continuously expanded and the accuracy can be improved, which will make it easier for people to interact with computer.

The remainder of this paper is organized as follows. In Section II, we review the related work briefly. Then the proposed method is introduced in Section III. Section IV gives the experimental results and the corresponding analysis. Finally, Section V concludes this paper.

II. RELATED WORK

The earliest research on natural language understanding is machine translation [7]. In 1949, American Weaver first proposed [8] the design of machine translation. There are massive researches on machine translation overseas in the 1960s, which cost huge fees. However, at that time it is obvious that people had underestimated the complexity of natural language. Due to the incomplete language processing theory and technology, little progress is achieved. The main method is to store the words and phrases corresponding to the translation of the two languages, and technically only adjust the same order of language. But the translation of natural language in everyday life is far from that simple.

Great changes have taken place in the field of natural language processing [9]. There are two obvious features: one is the system input. The natural language processing system

978-1-5386-1978-0/17/$31.00 ©2017 IEEE

which is required to be developed can deal with large scale texts, rather than only a few entries and typical sentences. Only in this way has the developed system real practical value. The other is the output of the system, in consideration to the difficulty of total understanding of natural language, we do not require the system to get a deep understanding of natural language texts, but should be able to extract useful information. For example, in the process of lexical analysis, the system should automatically extract indexing words [10], filterings and retrievals, and automatically extract important information, such as expressions, sentences, paragraphs and even the whole programs, etc.

Our recognition method does not consider the language complexity, but adjusts the same order of strings. Such as "abcde", possible segmentation methods are "abc de", "ab cde", etc. And 2^{n-1} methods in total. N is the length of the string, if all the possible segmentation methods are taken into consideration, the amount of computation will be very large. According to the maximum probability word segmentation algorithm, the maximum probability can be calculated to generate a training set to reduce the amount of computation and improve efficiency. Based on the lexical analysis and grammatical analysis of the course of Compilation Principle, by analyzing the process of language processing, through large-scale word segmentation experiments, our method can make students get a deeper understanding of the process of word segmentation.

III. The Maximum Probability

The foundation of natural language processing [11] is the data sets of many kinds of natural languages. During the process of the compiling principle, such as lexical analysis and syntax analysis, it requires students to define different kinds of data such as keywords tables, grammar rules. Then system will perform analysis based on the input parameters. However, a string may correspond to a variety of segmentation results. Thus, based on the pre-defined parameters, our goal is to pick up the segmentation result which holds the biggest probability as the string's analysis result for data processing, which is also the preparation for the next compile.

A. Participles

Segmentation is the process of reclassifying continuous strings into new strings according to certain rules. In the lexical analysis, we scan the source codes from left to right according to the word-formation rules, and generate a sequence of words for parsing [12]. These strings processing involves kind collation, semantic. Meanwhile syntactic analysis's process involves some grammar rules and semantic examination. This paper exploits the Maximum Probability Word Segmentation algorithm [13,14] to find the best segmentation.

The input contains many pending strings, As shown below:

$$C_n = C_1C_2...C_i...C_n(C_i \text{ for the characters }，i \in [1，n]) \qquad (1)$$

By analyzing and processing, the output word string:

$$S_m = S_1S_2...S_j...S_m(S_j \text{ is for the word }，j \in [1,n]) \qquad (2)$$

The process of lexical analysis is clearly presented as above. Herein, the $P(W \mid S)$ is denoted as follows in equation (3):

$$P(W|S) = \frac{P(S|W) * P(W)}{P(S)} \approx P(W) \qquad (3)$$

$$P(W) = P(w_1, w_2,..., w_i) \approx P(w_1) * P(w_2) * ... * P(w_i) \qquad (4)$$

Where W is the word, and S is the pending string. $P(W_i)$ is denoted as follows in equation (5):

$$P(w_i) = \frac{n}{N} \qquad (5)$$

Among them, n is the number of times W_i appears in the training set. N represents the total number of words in the training set. In the process of analysis, there will be left adjacent words. If you scan the string from left to right, you can get $W_1, W_2,..., W_{i-1}, W_i$ and... etc, and if the W_{i-1} 's tail is adjacent to W_i, we call W_{i-1} the left adjacent word for W_i. The leftmost word of the string has no left adjacent word. If a candidate word W_i has a number of left adjacent words W_j, W_k, ... etc, among which the most likely candidate word is called W_i is best left adjacent word.

Since our goal is the maximum probability, there is a data sparse problem. If a n-gram does not exist in the training set, the probability of n-gram must be 0. The solution is to extend the training set. However, no matter how big the training set you extend, it is impossible to ensure that all words exist in the training set. Due to the lack of samples, the estimated distribution is not reliable, which is called data sparsity problem. In the field of NLP, data sparse problems are always present, and there is not likely to be a large enough training set. There are some solutions, for example: Laplace smoothing technology, Good - Turing estimates [15]. The basic idea is to cluster the parameters according to the occurrence times(assuming $\#(x_j) = \#(x_j)$, then $\theta[j] = \theta[j']$), and then use the occurrence times+1 to estimate.

$$r^* = (r+1)\frac{n_{r+1}}{n_r} \qquad (6)$$

In the formula above, in the test set V , n_r stands for a total of n_r elements that have occurred n_r times in training set T. This method ensures that the total number of the test concentration elements in the training set is constant. That is:

$$N1 = \sum_{r=0}^{\infty} rn_r = 0 \times n_0 + 1 \times n_1 + 2 \times n_2 + ... \qquad (7)$$

$$\begin{aligned} N2 &= \sum_{r=0}^{\infty} r^* n_r = 1 \times \frac{n_1}{n_0} \times n_0 + 2 \times \frac{n_2}{n_1} \times n_1 + ... \\ &= 1 \times n_1 + 2 \times n_2 + ... \end{aligned} \qquad (8)$$

Obviously, $N1 = N2$. We make use of the training set to calculate the probability of all the possible string segmentation, looking for the most probable string of words as the best segmentation.

B. The Maximum Probability Segmentation Algorithm

The Maximum Probability Segmentation algorithm can be divided into the following steps:

- For a string of the words to be segmented, take out all the candidates words from left to right $W_1, W_2,..., W_{i-1}, W_i,..., W_n$;
- To look out the probability value of each candidate words in the training set. When the candidate word does not appear, the probability of the candidate word is $1/(\text{The training set}+1)$, and the entire left adjacent word of each candidate word is recorded.
- Calculate the cumulative probability of each candidate word according to formula (1), and compare the best left adjacent word of each candidate.
- If the current W_n is the tail of string S, and the cumulative probability $P'(W_n)$ is the largest, W_n is the terminal word of S.
- Starting from W_n , output the best left adjacent word output of each word from right to left, which is namely the segmentation result of S.

In the lexical analysis, input the pre-defined keyword lists and the sentences, the system will scan the input statements verbatim, utilize algorithm above to analyze, and finally output the results.

IV. Experiments And Results Analysis

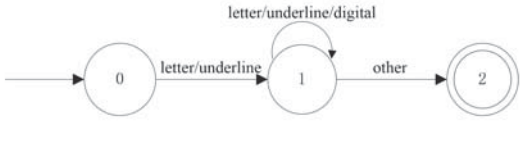
A. Datasets

In the process of lexical analysis, special characters have their own code tables, which outputs the corresponding number according to the corresponding relation when segmenting the strings. The corresponding relationship is shown in Table I:

TABLE I. CODE TABLE

Word Symbol	Code	Word Symbol	Code
begin	1	<=	21
if	2	,	22
then	3	>	23
while	4	>=	24
do	5	=	25
end	6	==	26
letter(letter|digit)*	10	(	27
digit digit*	11	)	28
+	13	;	29
-	14	[	30
*	15	]	31
/	16	{	32
:	17	}	33
:=	18	!=	34
<	20	#	0

When reading strings, it is always in an ordered sequence, one character by one character. In the state transition diagram, 0 is the initial state, and 2 is the final state. The process of identifying the transition diagram is to start with the initial state 0, and if the input character is a letter or underscore under state 0, then read it and shift into state 1.In state 1, if the next input character is alphabetic or numeric or underlined, read it and return to state 1. Keeping repeating the process until the input character is no longer a letter or number or underlined in state 1.Then it enters state 2. State 2 is the final state, which means that this string has been identified.



The identified string continues to be performed on the maximum probability analysis to get an accurate string segmentation.

B. Evaluation And Analysis

When applying the Maximum Probability Analysis algorithm, such as reading a string:

S : study/ idlewhilecollege

W_1 : study/ / / idle/ while/ college

W_2 : study/ / / idlewhile/ college

$$P(W_1) = P(\text{study}) \times P(/) \times P(\text{idle}) \times P(\text{while}) \times P(\text{college}) = 1.8 \times 10^{-9}$$

$$P(W_2) = P(\text{study}) \times P(/) \times P(\text{idlewhile}) \times P(\text{college}) = 1 \times 10^{-11}$$

$P(W_1) > P(W_2)$, So choose W_1 .

Therefore, the string is segmented using the Maximum Probability Analysis method, and the results are segmented into W_1 , W_2 .Then identify the while statement, and look up the identifier code by the corresponding relationship in Table I.

Of course, in the practical process, if the string is too long , the number of word segmentation will be very large. And the amount of calculation will increase exponentially. So we need to adopt a certain algorithm to reduce the computational complexity, we can see that the probability is calculated by multiplication. But in the lexical analysis and syntactic analysis of compiling principle, the ambiguity can be reduced, and the knowledge is collected and sorted out. Finding the right form and putting them into a computer system can help students understand and try to achieve practical experimental process.

To validate the accuracy of the method proposed in this paper, the newspaper China Daily was choosed for experiments. In China Daily of November 2010, there are about 3 million Chinese words in the newspapers. Based on those words, a total of 52380 entries were created, which consists of 4433 sentences in total.

The experimental results are shown in Table II:

TABLE II. WORD SEGMENTATION ACCURACY RATE

	Ordinary word segmentation	Maximum probability analysis
Total correct number of sentences	1929	2220
The total number of segmentation	115018	115108
Correct number	107198	110015
Accuracy rate	93.3675%	95.5755%
Combinational ambiguity	945	859
Intersecting ambiguity	1132	338

As can be seen from Table II, after the Maximum Probability Word Segmentation Algorithm is applied, the complete correct number of sentences, the correct number of segmentation and accuracy are all obviously increased. Meanwhile the number of ambiguous sentences is reduced. As a result, our method make the interaction between people and computers more convenient.

V. CONCLUSIONS AND DISCUSSIONS

Nowadays with the boom in computer education, the study of compiling principle play a quite important role in computer education. However, compiling principle experiment system is quite complex, which can be divided into different modules, each module can be divided into smaller parts. It is fairly difficult for students. Our proposed method provides a new thought on the string processing, which can better help students to master professional knowledge, smoothly finish courses. But from the theoretical perspective, our method have troubles in the collection, sorting, expressing and effective application of large scale knowledge, which is worthy of further research.

ACKNOWLEDGMENT

This work was supported in part by National Science Foundation of China under Grants No. 61402304, the Beijing Educational Committee Science and Technology Development Planned under Grants No. KM201610028015, Beijing Science and technology innovation platform.

REFERENCES

[1] Niklaus Wirth. A basic course on compiler principles[J]. BIT,1969,94.

[2] Garrido, Antonio,Onaindia, Eva.Assembling Learning Objects for Personalized Learning: An AI Planning Perspective[J].IEEE intelligent systems,2013,28(2):64-73.

[3] Michele Carenini,Angus Whyte,Lorenzo Bertorello et al.Improving Communication in E-democracy Using Natural Language Processing[J].IEEE intelligent systems,2007,22(1):20-27.

[4] Kevin Warwick,Huma Shah. The importance of a human viewpoint on computer natural language capabilities: a Turing test perspective[J]. AI &amp; SOCIETY,2016,312.

[5] Bogdan Patrut,Ioana Boghian. A Delphi Application for the Syntactic and Lexical Analysis of a Phrase Using Cocke, Kasami and Younger Algorithm[J]. Brain. Broad Research in Artificial Intelligence and Neuroscience,2010,12.

[6] Wu Yang. On the look-ahead problem in lexical analysis[J]. Acta Informatica,1995,325.

[7] Álvaro Peris,Miguel Domingo,Francisco Casacuberta. Interactive neural machine translation[J]. Computer Speech &amp; Language,2016.

[8] Xu Chen,Judith Gelernter,Han Zhang,Jin Liu. Multi-lingual geoparsing based on machine translation[J]. Future Generation Computer Systems,2017.

[9] Michael Tanana,Kevin A. Hallgren,Zac E. Imel,David C. Atkins,Vivek Srikumar. A Comparison of Natural Language Processing Methods for Automated Coding of Motivational Interviewing[J]. Journal of Substance Abuse Treatment,2016.

[10] Andreas Maletti. Survey: Finite-state technology in natural language processing[J]. Theoretical Computer Science,2016.

[11] I. De Falco,A. Della Cioppa,A. Iazzetta,E. Tarantino. An evolutionary approach for automatically extracting intelligible classification rules[J]. Knowledge and Information Systems,2005,72.

[12] Randall Art. The cost of data processing. 1985.[J]. Health Management Technology, 2010, 31(4).

[13] Jie Ding. Research on the Chinese word segmentation method based on maximum probability segmentation algorithm [J]. Science and technology information, 2010,(21):587. [2017-10-08].

[14] Hualin Zeng , Tangqiu Li, Xiaodong Shi . A segmentation algorithm based on extraction context information [J]. Computer application. 2005(09).

[15] Da Wang, Rui Cui . Data smoothing technology review [J]. Computer knowledge and technology, 2009,5(17):4507-4509. [2017-10-08].