Realization and optimization of Chinese syntactic analyzer based on PCFG

Dengyun Zhu 1, Bin Zhang 1, Qi Guo 1, Shuang Wang 1, Fucheng Wan 2,* and Ning Ma 2

1 Key Laboratory of China's Ethnic Languages and Information Technology of Ministry of Education Northwest Minzu University Lanzhou, Gansu, China
2 Key Laboratory of China's Ethnic Languages and Intelligent Processing of Gansu Province Northwest Minzu University Lanzhou, Gansu, China

*Corresponding author e-mail: 4235152565@xbmu.edu.cn

Abstract. As a basic research work, syntactic analysis is also a key technology in the field of natural language processing, and is the basis for deep understanding of language. The syntactic analyzer is an important tool for syntactic analysis. The Probabilistic Context Free Grammar (PCFG) model is the basic model of most parsers with excellent performance. Good effect. This article integrates named entity clues into PCFG-based Chinese syntax analysis models, completes joint learning and decoding of named entity recognition, syntactic analysis, and implements a fast and reliable decoding parser.

Keywords: PCFG Parser; Named entity recognition; Decoding performance.

1. Introduction

This work is supported by Natural Science Fund of GanSu Province (NO. 21JR1RA217, 21JR1RA196) and the Fundamental Research Funds for the Central Universities(NO. 31920210020). Natural language is a way of human communication and an important part of human behavior. In the current society, natural language is not only a communication tool between people, but also the basis of human-machine dialogue. Enabling computers to understand natural languages and communicating with computers using natural languages is a long-term goal pursued by human beings, and has important theoretical value and practical significance. Natural language processing is the study of various theories and methods that enable effective communication between humans and computers using natural language. Current researchers generally believe that natural language processing systems include the following knowledge levels: phonological level, morphological level, lexical level, syntactic level, semantic level, and pragmatic level. Moreover, from the syntactic level upwards, it is a deep analysis, which is more closely related to language understanding. Because syntactic analysis occupies the core position of inheritance, it has always been the focus and difficulty of natural language processing research. In addition to providing technical support for word sense disambiguation and semantic analysis, the results of syntactic analysis can be directly used in applications such as machine translation, question answering systems, and information extraction, and improve the performance of applications. It can be said that syntactic analysis is the core technology of natural language processing and the basis for deep understanding of language.
2. Syntax parser introduction
Syntactic analysis is one of the key technologies in natural language processing. Generally speaking, syntax analysis is not the ultimate goal of a natural language processing task, but it is often an important link or even a key link to achieve the final goal. Therefore, various new theories and methods are constantly proposed around this issue.

PCFG is an extension of CFG. Probability-based context-free grammar has both the characteristics of the rule method and the use of probability information. It can be considered as a close combination of the rule method and the statistical method. The research of PCFG-based syntax analysis methods has attracted much attention. At the same time, the PCFG-based syntax analysis model is the basic model of most excellent parsers, but using PCFG modeling alone does not achieve good results. Many researchers have worked hard to come up with solutions to improve performance. To improve and optimize PCFG, we try to break through PCFG's independence assumption by introducing various contextual information. Depending on the information introduced, the PCFG-based improvement methods can be divided into two broad categories: the introduction of lexicalized information and the refinement of non-terminal tags. The introduction of lexicalization information refers to attaching the headword of each phrase in the syntax tree to the phrase node to generate a tree with the headword. Compared with PCFG, the improvement of lexical parsing is quite effective, but due to the introduction of too much lexical dependent information, it will face the problem of data sparseness. In response to this problem, researchers have adopted a non-lexical syntactic analysis method based on markup refinement, and have also achieved significant performance improvements. Among them, UC Berkeley's Petrov et al. Proposed a method of hierarchical tag refinement, which automatically matched and merged the model structure and training data through the automatic splitting and merging of grammatical tags. Because of its language independence and excellent syntactic analysis performance, it has good practicability and has won the favor of researchers.

From lexicalization methods and complex smoothing to category refinement, more syntactic features are obtained by refining grammatical marks. Implicitly labeled probabilistic context-free grammars automatically refine the syntactic and lexical tokens in an unsupervised manner and achieve the best performance currently available in Chinese and English. A method of syntactic analysis training and decoding based on hidden variables proposed by Petrov and others, and based on this, the Berkeley parser parser is completed. In this paper, a parser + named Berkeley parser + that incorporates named entity recognition is constructed.

3. Implementation and optimization of a parser
Syntactic analysis is the core technology of natural language processing and the basis for deep understanding of language. Syntax analyzer is an important tool for analyzing sentences. PCFG-based syntactic analyzers are the basic models of most excellent parsers, but using PCFG modeling alone does not achieve good results. Based on the research of PCFG-based Berkeley parser, this paper incorporates named entity clues to optimize the PCFG-based syntactic analysis model.

Named entities are those nouns or noun phrases that can explicitly refer to an object in the outside world. The quotation of the concept of entity in text can take three forms: named referential, nominal referential, and pronoun referential. Related to this task, named entity recognition is the process of identifying and identifying types of text fragments such as person names, place names, and agency names in a document. It is one of the key technologies for natural language processing research such as information extraction, question answering system, machine translation, document summary, and cross-language retrieval. For a long time, named entity recognition and syntactic analysis have been modeled and analyzed at different levels. This article attempts to jointly study named entity recognition and
syntactic analysis, integrate named entity information into syntactic analysis, and perform integrated learning and decoding.

Mark the named entity information on the tree structure, comprehensively consider the lexical layer and syntactic layer information to avoid cascading errors. One obstacle to using the joint model to produce consistent output is that currently there is no suitable large-scale corpus to label all this information. The most commonly used corpus for part-of-speech tagging tasks is the People's Daily Annotated Corpus, but this corpus does not contain a syntactic structure. In the Chinese syntax analysis task, Penn's Chinese treebank is most widely used, but the treebank does not have enough accurate named entity information. In this paper, a more informative corpus is constructed by refining the named entity structure tags in the Chinese Penn Treebank (CTB).

Penn Chinese Tree Bank is the most widely used tree bank in Chinese syntax analysis. The corpus of CTB word segmentation, part-of-speech tagging and syntactic tagging currently has 500,000 words (more than 800,000 Chinese characters). There are a total of 890 files in CTB5.0. Chinese syntactic analysis is usually based on CTB sparse part-of-speech and syntactic tags. In CTB, named entity phrases are simply labeled as noun phrases (NP) without distinguishing between their multiple types. Similarly, named entity words are simply labeled as unique words (NR), basic numbers (CD), ordinal numbers (NT). They correspond to words in the syntax tree without marking their internal word structure.

How to define the type of a named entity is an important issue. OntoNotes is a corpus labeled with named entities and syntactic information. OntoNotes Release 4.0 is a manually annotated corpus that contains a variety of text types and tags. It is also a corpus containing Chinese named entity tags. It contains a 403 file also in CTB5.0, which contains a test set and a development set with standard syntax evaluation settings. Named entities in OntoNotes 4.0 are labeled as 18 types including people (PERSON), organization (ORGANIZATION), geopolitical economic entity (GPE), place (LOC), product (PRODUCT), etc. Some entity types do not appear often and are not always useful in practice, such as art, product, and law. Therefore we ignore these types in the study of text. In addition, we have marked codes, percentages, and phone numbers. We use another idea to explicitly represent named entities where the syntactic structure is nested. Nested named entities are marked in the syntax tree, each corresponding to a node in the syntax tree. Next, we will detail the labeling process. We label named entities at the word level and the word level, respectively. In the end, we enrich the syntax tree with refined named entity tags to get joint tags.

The process of labeling named entities is as follows: First, the sentences in OntoNotes will be selected to generate a small tree library with labeled entities. The parser is then used to mark the remaining sentences. The sentence after the parsing will be corrected manually. Two people gave correct labeling to each named entity. Manual labeling accuracy is necessary to avoid the danger of low recall. Two people should negotiate to find a reasonable mark when there is a mark difference.

At the word level, we refine the labels of named entity components. The structure of all words has not changed, but a thin label has been added to replace the previous rough mark. At the word level, we then label the internal structure of a word to represent a nested named entity. They are three named entities: GPE, PERSON, and time expressions. We design a series of labeling rules for these three types so that they can more accurately express named entity information and date information. We will deal with them in the following ways. For GPE, we split the GPE name and geographic unit in a tree structure. This marking method makes full use of the combined structure of the usual GPE. For example, "Shenzhen" and "Shenzhen" are GPE. The Chinese character " city " will get a special mark. In this way, we get a production: GPE→ GPE GPEend. An example is shown in Figure 1.

![Figure 1. Tree structure representation of affix](image-url)
We also distinguish foreign names from Chinese names by named entity tags NR→PERSONF (foreign name) and NR-PERSONC (Chinese name). Obviously, a name containing "" is a foreigner's name. Using this clue, identifying foreigners' names is a simple matter. An example is shown in Figure 2.

![Figure 2. Tree structure representation method of European and American names](image)

For time expressions, the nested structure is divided into numeric expressions and time units. For example, the word "fifteenth day" will be divided into fifteen-NUM and day-day. An example is shown in Figure 3.

![Figure 3. Method for representing date tree structure](image)

Finally, if a word-level syntax tree is used, the word-level syntax tree is transformed into a word-level syntax tree according to the rules. A simple conversion rule is described as follows: all word-level part-of-speech tags are consistent with the word-level syntax tree. For each word, we create a new node for it and assign a new label to the new node. The new tag contains the part-of-speech tag and its place in this class. (B indicates the beginning of the word, E indicates the end of the word, and M indicates the position in the word). For example, the Chinese character "jiao" will be marked as "NNb" in "NN-Education Bureau". Figure 4 shows a specific word-based syntax structure.

![Figure 4. Word-based syntax tree tagging results](image)

4. Experimental setup and analysis
In this paper's experiments, we will verify the different effects of PCFG-based syntactic analyzers in Chinese syntactic analysis with or without fused named entity cues. First, a new corpus is used to train a PCFG-based syntax analysis model. At the same time, the syntax analysis model also outputs named entity recognition results.

4.1. Corpus selection
This experiment uses Penn Chinese tree library. Add tags for named entity information in Penn's Chinese
tree library for related experiments. This article refers to the use of a general popular syntax analysis
and evaluation division method.

4.2. Experimental setup
Experiment 1:
The word-based parser does not include named entity information as the baseline of the results of the
parsing and is labeled Berkeley parser.

Experiment 2:
Word-based parser, adding named entity information, labeled Berkeley parser +.
Note that the Berkeley parser here needs to be modified for word-based input. The regular matching
rules and the thresholds of unregistered words must be modified accordingly.

4.3. Syntactic analysis performance
The experiment includes two items, one is to test the decoding accuracy and decoding speed with a
simple word-based parser (Berkeley parser), and the other is to test the decoding accuracy with a word-
sequence-based parser including named entity information And decoding speed. The following table 1
shows the comparison of decoding accuracy of Chinese syntax analysis, and Table 2 shows the
comparison of decoding speed of Chinese syntax analysis.

| Table 1. Comparison of Chinese Syntactic Analysis Precision |
|------------------------------------------------------------|
| System          | P   | R   | F   |
| Berkeley parser | 79.94 | 78.97 | 79.43 |
| Berkeley parser+ | 81.96 | 80.84 | 81.28 |

| Table 2. Comparison of Chinese Syntactic Analysis Speed |
|--------------------------------------------------------|
| System          | Time(second) | Speed(sentences/s) | Speed(character/second) |
| Berkeley parser | 570.79       | 0.54               | 20.91                 |
| Berkeley parser+ | 874.37       | 0.40               | 15.78                 |

The above experiments show that adding named entity information can effectively improve the
performance of syntactic analysis. The word formation and context of named entities and common
words are significantly different. After adding the category information of named entities, it can increase
the discrimination and improve the overall performance. When using nested named entity information
in the training corpus, the experimental results achieved the best results, and the F1 value was about 2%
higher than the baseline system. In terms of speed, Berkeley parser + will decrease. The reason here is
that more marker information is introduced, the decoding space becomes larger, and the speed decreases.

5. Conclusion
This paper presents a PCFG-based parser that incorporates named entity information. After investigating
the current corpus, we refined the named entities and syntactic structure tags in the treebank. Then, an
integrated syntactic analyzer based on hidden variables is implemented, and the validity of the tag is
verified through experiments. This joint marking method improves the accuracy of syntactic analysis.
At the same time, the named entity recognition result is better than the CRF-based named entity
recognition system. However, after the introduction of named entities, the decoding space can continue
to be optimized, and the clues of historical memory information are not better used during the decoding
process. At the same time, we need to think deeply about the influence of the Beam size of the column
search in the learning and decoding process.
Acknowledgments
This work is supported by Natural Science Fund of GanSu Province (NO. 21JR1RA217, 21JR1RA196) and the Fundamental Research Funds for the Central Universities (NO. 31920210020).

References
[1] Sample Selection for Statistical Parsing[J]. Rebecca Hwa. Computational Linguistics. 2004 (3).
[2] A sequential algorithm for training text classifiers[J]. David D. Lewis. ACM SIGIR Forum. 1995 (2).
[3] Thumbs Up or Thumbs Down Semantic Orientation Applied to Unsupervised Classification of Reviews. Peter D. Turney. Proceedings of the 40thACL. 2002.
[4] A Non-negative Matrix Tri-factorization Approach to Sentiment Classification with Lexical Prior Knowledge. Tao Li, Ti Zhang, Vikas Sindhwani. Proceeding of ACL. 2009.
[5] Automatic Seed Word Selection for Unsupervised Sentiment Classification of Chinese Text. Taras Zagibalov, John Carroll. Coling**. 2008.
[6] Tackling the poor assumptions of naive Bayes text classifiers. RENNIE J D M. Proc. Twentieth International Conference on Machine Learning. 2017.
[7] An Empirical Study of the naive Bayes Classifier. Rish I. Journal of Universal Computer Science. 2001.
[8] Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. Cho K, Mrienboer B V, Gulcehre C, et al. Computer Science. 2014.
[9] A Sentiment Classification Model Based on Multiple Classifiers. CATAL C. NANGIR M. Applied Soft Computing. 2017.
[10] On the effects of using word2vec representations in neural networks for dialogue act recognition. Cerisara C, Kral P, Lenc L. Computer Speech & Language. 2017.
[11] Syntactic Processing Using the Generalized Perceptron and Beam Search[J]. Yue Zhang Stephen Clark. Computational Linguistics 2011 (1).