What is the Best Solution in Constituency Parsing in Chinese Natural Language Processing?

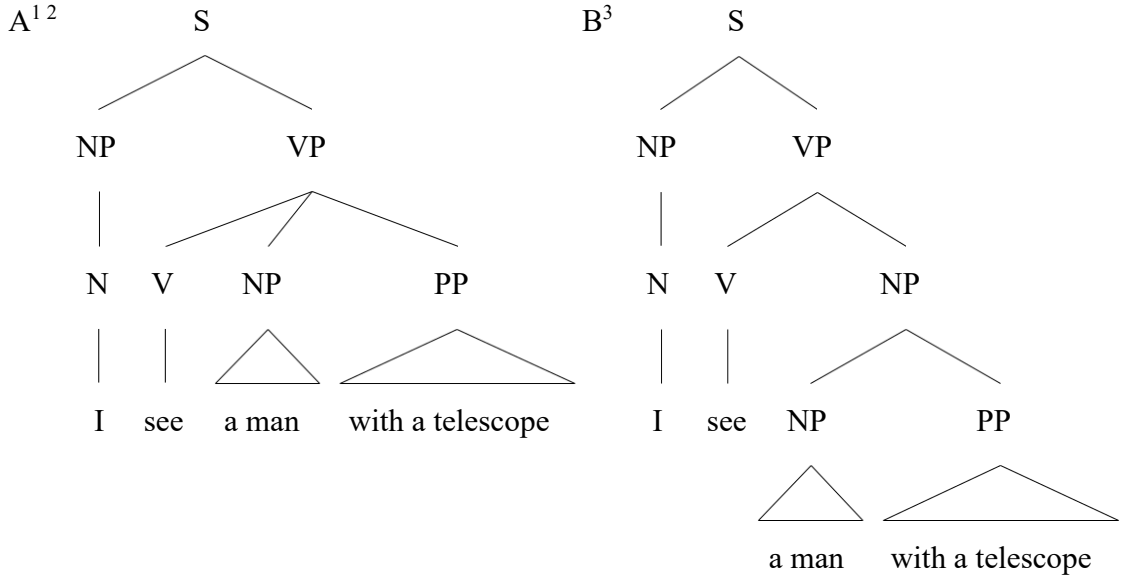
Tongxi(Tom) Liu

University of Washington

Introduction

"I see a man with a telescope."

Does the speaker use the telescope to see the man? Or does the speaker see a man who has a telescope? It turns out that these two interpretations are both possible. If we structure this text into linguistic trees, two valid trees will account for this sentence as shown in tree A and B below. This variation of trees for one text is called **Structural Ambiguity (SA)** in Linguistics. How to build an accurate model to solve the SA problem in computers is a controversial problem.



Before introducing how computers analyze languages, it is essential to consider how humans distinguish SA in daily conversations. When babies grow up, parents do not deliberately teach them grammar as teaching an adult another language. Instead, according to the theories of language acquisition proposed by VanPatten, Keating, and Wulff (2020), children are born with an innate blueprint for languages. This blueprint is

---

[1] S stands for *sentence*; PP stands for *prepositional phrase*; VP stands for *verbal phrase*; N stands for *noun*; V stands for *verb*.
[2] The speaker uses the telescope to see the man.
[3] The speaker sees a man who has a telescope.

also called the **Universal Grammar (UG)**, a "computer program" that all humans are "pre-loaded" with, but that still needs linguistic input to run. UG helps children construct the language model and generate grammar rules in the brain, and with a well-formed model, children can distinguish the SA without teaching. Is it possible for computers to build such a UG program to solve the SA problem?

Some neurologists believe that if we can simulate an infant's brain in a computer, it is possible to develop a UG program to solve language problems (Pulvermüller and Schumann 1994). However, the current neurological exploration of the human brain is only the preliminary stage, so it seems it is still far from solve the SA problem. The need to solve SA problems has been increasing; for example, more accurate search engines and more intelligent voice assistants need to load **Natural Language Processing (NLP)** models that can handle SA problems. Therefore, linguists and computer scientists use computer algorithms to build models for SA problems instead of building a UG program. The way to solve the SA problem in linguistics is called **Constituency Parsing (CP)**. One major step in CP is **Sentence Structure Modeling (SSM)**.

When the SA problem comes to the language of Chinese, the path to developing a good solution is more complicated. Chinese sentences are glued with words, which are not like English, where the sentences have spaces boundary between words. For the computer to understand the text, in addition to SSM, it needs to implement a **Chinese Word Segmentation (CWS)** model that adds boundary markers between words in Chinese sentences.

Some linguists think traditional methods, such as dictionary-based models and searching algorithms, are good enough to build the CWS model and handle SSM. In

contrast, others argue that advanced algorithms, such as **Conditional Random Field (CRF)**, have more advantages in CWS modeling, and **Context-Free Grammar (CFG)** based algorithm, such as **Cocke–Younger–Kasami (CYK)**, can handle SSM in Chinese language accurately. Through analyzing and comparing traditional methods and new algorithms in this paper, I argue that CRF and CFG based CYK algorithms are the best CP solution in Chinese NLP because (1) CRF can build an accurate CWS mode that account for critical information about the order of words and find new words, (2) CFG based CYK can structure SSM accurately and efficiently, and (3) CYK and its extended algorithms are refreshing the record of CP speed.

CRF Algorithm in CWS

To introduce an example of **Structural Ambiguity (SA)** problem in Chinese, consider the following examples (1) to (3):

(1) Jiehundeheshangweijiehunderen

(2) Jiehunde**/he/shang**weijiehunde/ren

Marry/**and/not**-marry/people

'Married people and single people'

(3) *Jiehunde/**heshang**/weijiehunde/ren

*Marry/monk/non-marry/people

Four words construct noun phrase (NP) (1) *Jiehundeheshangweijiehunderen*. Depending on segmentation, it can generate either grammatical NP (2) or ungrammatical NP (3). A Chinese speaker can use mental grammars to distinguish (2) is the valid interpretation of NP (1) and eliminate the possibility of (3). Building a grammar model for computers to eliminate ungrammatical possibilities of phrases is a significant task in **Constituency Parsing (CP)**. The first step is to segment the words correctly.

The analogy of photos segmentation inspires linguistics to build a **Chinese Word Segmentation (CWS)** model. As introduced in the previous section, the dilemma of CWS is an essential factor hindering the computer from segmenting a sentence into small pieces. To explain how CWS segmentizes a sentence into small pieces, we assume a Chinese sentence is similar to consecutive photos in a day. The job of the CWS model is to group consecutive photos of related activities (or group consecutive word fragments of related semantic meanings). The outcoming program of this grouping process is called a **multivariate classifier**. A simple and intuitive way to make a CWS is to train a multivariate classifier

regardless of the chronological order between these photos. To prepare a multivariate classifier, according to Tsoumakas and Katakis (2007), they use some labeled photos as training data to train a model to classify directly based on the characteristics of the photos. Therefore, a multivariate classifier in CWS is similar: linguists label each word in the sentence as training data and train a model to classify the words directly based on the characteristics.

The multivariate classifier seems like a good solution but has flaws. Since we ignore the critical information about the order of these words, this simple classification is imperfect. Like the above example (3), the classifier conforms *heshang/monk* as a word in this NP because the common words *heshang/monk* are in the dictionary of the training set. However, according to the context, the presence of *heshang/monk* does not fit here. The multivariate classifier fails to eliminate example (3) as an ungrammatical NP.

The **Conditional Random Field (CRF)** algorithm solves the constituency parsing in CWS by part-of-speech (POS) tagging and sequence scoring based on the multivariate classifier. CRF was proposed and tested by Lafferty, McCallum, and Pereira (2001). While training a CWS model, CRF requires POS tagging to each word in a sentence. As shown in (4), there are four words with POS tags [NP PP VP NP]:

(4) Wo mingtian qu xuexiao

I tomorrow go school

NP PP VP NP

'I will go to school tomorrow.'

According to Zhao, Huang, and Li (2006), Huang, and Li (2006), to build a CRF model, they need such POS tags from sufficient number of sentences to build POS sequences.

When an unknown sentence is an input into the CRF model, it first uses POS tags to segment sentences into words. CRF scores each POS sequence and returns the fragments with the highest possibility of multiple segmentation possibilities. In our examples (1) to (3), the CRF model can eliminate NP (3) and select NP (2) as the correct result.

Some researchers argue that the CRF algorithm is complicated, and the CWS problem can be solved based on the dictionary word segmentation algorithm. According to Qiu, Xie, Wu, and Li (2018), the dictionary-based word segmentation algorithm matches the string with a word in a well-established and sufficiently extensive dictionary. If the dictionary contains an entry, the match is successful, and the word is recognized. It makes sense that Qiu's dictionary-based model can achieve a fast solution to the CWS problem considering their paper focuses on the subject domain of geoscience. However, when the scope of application is enormous, new words not in the dictionary appear. For example, in the application of movie review sentiment, users often discuss the phrases of characters and unique nouns in movies. These new words never occur in the dictionary; thus, the dictionary-based model fails to segment them correctly.

CRF algorithm can handle emerging new words in the Chinese language. According to Tian, Song, Xia, Zhang, and Wang (2020), if a new word that never appears in the training dataset, the CRF model can predict what kind of POS it is through inference of context POS sequence. Therefore, a well-trained CRF model is better than dictionary-based word segmentation algorithm since CRF can handle CWS with new words and adapt to new language variations.

Context-Free Grammar in Sentence Structure Modeling

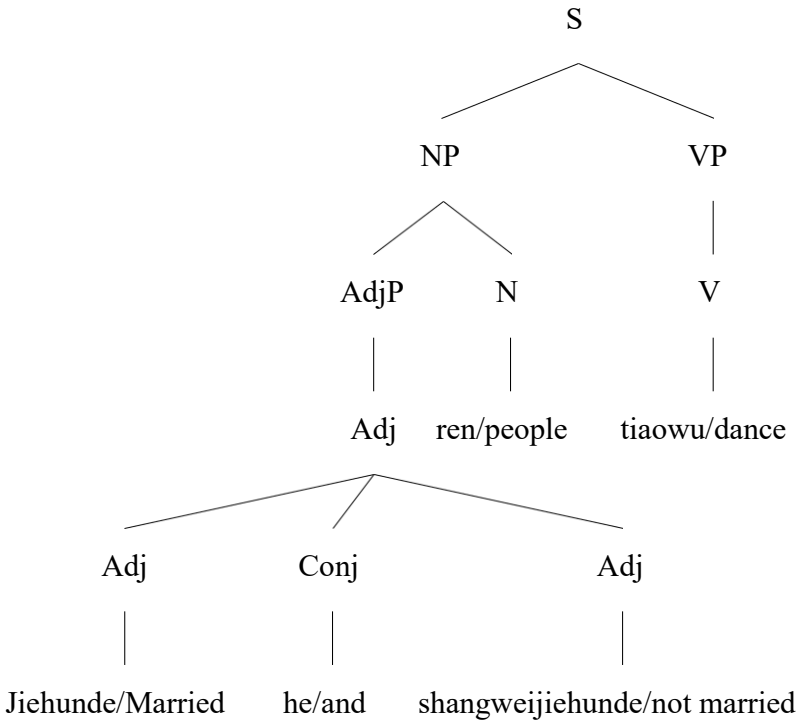
Context-Free Grammar (CFG) is an excellent method to model sentence structure in Chinese because it can use simple rules and lexicon to account for most Chinese sentences. Designed by Chomsky (1956), as each word in a sentence has a POS tag, CFG links these tags together to generate a set of simplified grammar rules. To demonstrate how CFG models sentences, consider an example of CFG shown in (5) and (6):

(5) [4]

Rules:	Lexicon:
S→NP VP	N: ren/people
NP→N	Adj: jiehunde/married, shangweijiehunde/not married
AdjP→Adj	V: tiaowu/dance
VP→V	Conj: he/and
X→ X Conj X	

[4] AdjP stands for *adjective phrase*; Adj stands for *adjective*

(6)



A CFG comprises two parts to help model sentence structures, the first is a set of rules, and the second is a lexicon containing a list of words. To derive a set of rules as in (5), according to Sag, Wasow, Bender, and Sag (1999), they decompose a sentence into a top-down tree structure as in (6). The node above a bottom node is called the mother of the bottom daughter node. For example, node **S** is the mother of daughter nodes **NP** and node **VP**. By observing the tree, linguists summarize what node(s) can be specific mother node's daughter, then writes rules ***Mother*→ *Daughter*$_1$ *Daughter*$_2$ *... Daughter*$_n$**. Although the set of rules in (5) can only generate sentence (6), by analyzing more sentences in this way, the set of rules grows and can cover every grammar in Chinese. Example (7) is a demonstration.

(7)(henduo/dade/piaoliangde/)qiqiu(/zaitianshang)/fei

(many/big/pretty/)balloon(/in the sky)/fly

'(Many big pretty) balloons fly (in the sky).'

Eliminating many big pretty and in the sky, (7) is still a grammatical sentence (i.e., *Balloons fly*). Noticeably, *many big pretty* comprises three optional AdjP, and *in the sky* is also an optional PP. Therefore, based on the set of rules in (5), we can modify the NP rule as **NP→(AdjP)* N** and VP rule as **VP→(PP) V**. The parentheses between a node means that the daughter is optional, and the star means that the daughter can appear more than once. CFG allows the recursive rules (i.e., a rule contains the mother node on the right side of the rule), such as **NP→(NP) N**. Therefore, the flexibility and recursiveness allow CFG to model most grammar structures in Chinese.

When there is a set with broad enough rules that can cover most grammars in Chinese, building the lexicon enables CFG to generate infinite sentences. Like the lexicon dictionary in (5), a lexicon contains keys of POS and values of words of key's POS. In the

traditional CFG model building process, according to Chomsky (1956), lexicon building is a complement step with CFG rules set building. However, the previous section introduced how CRF algorithm handles CWS by POS tagging: CFG lexicon can borrow the POS dictionary from trained CWS model. Once well-formed lexicon is built, linguists can apply the words in the lexicon into the rules. Eventually, linguists can compare the corpus to generated sentences and automatically model the valid sentence structures.

Some researchers argue searching algorithms can solve SA problem already. With a well-formed CFG model and a big enough lexicon, it seems whether a CFG accepts the sentence is a mathematical fact; thus, the SA problem can be solved already. It is an approach to solve the SA problem in Chinese, as experimented by Loritz (1992). Loritz builds a GPARS system (Generalized Transition Network System) to solve the SA problem as a search problem searching. The search algorithms contain depth-first search (DFS) and breadth-first search (BFS) algorithms. Although it can solve SA problem, Loritz does not mention the vital problem of the worst-case performance of search algorithms in the paper. According to Even (2011), the worst-case performance for DFS and BFS is exponential. In other words, with the increasing length of a sentence linearly, the time consumption of parsing increases exponentially. The application of parsing, such as real-time translation, requires a relatively small-time complexity, so this method to solve the SA problem in Chinese is not an efficient solution.

CYK in CP

Extended on the CFG sentence structure model, the **Cocke–Younger–Kasami (CYK)** algorithm uses dynamic programming to generate possibilities for ambiguous sentences with low time complexity. Designed by Cocke (1969), Kasami (1966), and

Younger (1967), CYK is a bottom-up syntax analysis algorithm that utilizes dynamic programming: tabulating and storing substring parses based on CFG structures to avoid doing repeated work.

CYK first formalizes a CFG model into a **Probabilistic Context-Free Grammar (PCFG)** to save the time of searching. PCFG adds two steps after building a CFG model: computing (a) the possibilities of each rule occurring in the set of rules with the same mother node and (b) the possibilities of each word with the same POS tag. Example (8) is a transformed PCFG of example (5) with numbers in the brackets showing the possibility of occurrence.

(8)

Rules:	Lexicon:
S→NP VP [1.0]	N: ren/people [1.0]
NP→N [0.7], NP→ADJP N [0.3]	Adj: jiehunde/married [0.5], shangweijiehunde/not married [0.5]
AdjP→Adj [1.0]	V: tiaowu/dance [1.0]
VP→V [0.7], VP→V PP [1.0]	Conj: he/and [1.0]
X→ X Conj X [1.0]	

By pre-building the possibilities of each rule and word, PCFG serves as a pre-built dictionary for further CYK processing and saves the time to search possible combinations in parsing.

CYK transforms the sentences into the matrix and utilizes dynamic programming to extract the possibilities of the syntactic structure of the sentences. According to Cocke (1969) Kasami (1966), and Younger (1967), by building an $(n + 1) * (n + 1)$ matrix, where n is the number of words in the sentence, CYK incrementally builds a parse that spans the whole input string column by column from left to right and bottom to top. In this way, the matrix will generate possible parsing situations with their possibilities computing from the PCFG

model. So far, the different parsing situations represent different meanings for the ambiguous sentence. The parsing situation with a higher possibility is more likely to be the preferred meaning of the ambiguous sentence. Thus, the CYK algorithm is an excellent solution to the SA problem.

As for the performance, CYK and its extended algorithm are much faster than search algorithms. According to Cocke (1969), Kasami (1966), and Younger (1967), the worst-case running time of the original CYK algorithm is only $O(n_3)$, where n is the length of the sentence. In other word, the time consumption of parsing is only cubed of the length of a sentence, which is far less than exponential relationship of search algorithms discussed in previous section. There is still room for improvement for CYK by adding the step of matrix multiplication. Lee (2002) proves CYK can reach even a lower time complexity of $O(n^{3-\text{epsilon}/3})$, where time consumption of parsing is less than the cubed of the length of a sentence. Thus, with linguists and computer scientists' work, the CYK based algorithms are continually refreshing the records of SSM in Chinese NLP (Tian, Song, Xia, Zhang, and Wang 2020).

Conclusion

In conclusion, CRF and CFG based CYK algorithms are an excellent solution to CP in Chinese since CRF builds an accurate CWS model and CFG based CYK structures SSM efficiently. Beyond these advantages, CRF and CFG based CYK also outrank other methods. In detail, the multivariate classifier method in CWS modeling is outranked by CRF because it lacks critical information about the order of words. The dictionary-based method in CWS has a poor ability to handle emerging new words in the language. The searching algorithms in SSM cannot meet a small-time complexity requirement in real-world applications compared to CFG model. Therefore, CRF and CFG based CYK become the best solution that can contribute to the realm of Chinese NLP applications.

Although CRF and CYK algorithms achieved the fastest result, there is no perfect model to solve the SA problem in NLP because our languages evolve every day. The potentials, helping computer scientists and linguists faster adapt to new applications and solve practical problems, make CRF and CYK the best solution in CP in Chinese NLP.

References

Chomsky, N. (1956). Three models for the description of language. *IRE Transactions on information theory*, *2* (3), 113–124.

Cocke, J. (1969). *Programming languages and their compilers: Preliminary notes*. New York University.

Even, S. (2011). *Graph algorithms*. Cambridge University Press.

Kasami, T. (1966). An efficient recognition and syntax-analysis algorithm for context-free languages. *Coordinated Science Laboratory Report no. R-257* .

Lafferty, J., McCallum, A., & Pereira, F. C. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

Lee, L. (2002). Fast context-free grammar parsing requires fast boolean matrix multiplication. *Journal of the ACM (JACM)*, *49* (1), 1–15.

Loritz, D. (1992). Generalized transition network parsing for language study: The gpars system for english, russian, japanese and chinese. *Calico Journal*, 5–22.

Pulvermüller, F., & Schumann, J. H. (1994). Neurobiological mechanisms of language acquisition. *Language learning*, *44* (4), 681–734.

Qiu, Q., Xie, Z., Wu, L., & Li, W. (2018). Dgeosegmenter: A dictionary-based chinese word segmenter for the geoscience domain. *Computers & Geosciences*, *121* , 1–11.

Sag, I. A., Wasow, T., Bender, E. M., & Sag, I. A. (1999). *Syntactic theory: A formal introduction* (Vol. 92). Center for the Study of Language and Information Stanford, CA.

Tian, Y., Song, Y., Xia, F., Zhang, T., & Wang, Y. (2020). Improving chinese word segmentation with wordhood memory networks. In *Proceedings of the 58th annual meeting of the association for computational linguistics* (pp. 8274–8285).

Tsoumakas, G., & Katakis, I. (2007). Multi-label classification: An overview. *International Journal of Data Warehousing and Mining (IJDWM)*, *3* (3), 1–13.

VanPatten, B., Keating, G. D., & Wulff, S. (2020). *Theories in second language acquisition: An introduction*. Routledge.

Younger, D. H. (1967). Recognition and parsing of context-free languages in time n3. *Information and control*, *10* (2), 189–208.

Zhao, H., Huang, C., & Li, M. (2006). An improved chinese word segmentation system with conditional random field. In *Proceedings of the fifth sighan workshop on chinese language processing* (pp. 162–165).