International Journal for Housing Science and Its Applications
Publish August 6, 2025. Volume 46, Issue 3 Pages 7277-7288

https://doi.org/10.70517/ijhsa463623

A Study on the Chronological Evolution of Chinese Relative Clauses Based on Computer Grammar Analysis

Jiatian Sun[1,*]
[1] Language and Linguistic Science, University of York, Heslington, YO105DD, UK
Corresponding authors: (e-mail: 756542784@qq.com).

Abstract Chinese grammar is a formal rule describing the grammatical structure of Chinese language, and it is also an important cornerstone of Chinese language related research. In this paper, we take the Chinese relational clauses as the research object, and analyze the basic concepts of fuzzy context-independent grammar in Chinese relational clauses. Based on the characteristics of Chinese grammatical structure, the stack-maximum matching automatic word segmentation model is designed as the automatic word segmentation technique for the grammatical analysis model of Chinese relational clauses by combining the maximum matching method and the stacking technique. For the analysis of grammatical knowledge points in Chinese relational clauses, they are analyzed in modules by decomposing them into lexical and syntactic parts. At the same time, the Chart algorithm is introduced as the grammatical analysis algorithm for Chinese relational clauses, and a grammatical analysis model for Chinese relational clauses is constructed. The model is used to analyze the distribution of accusative constructions in subject relative clauses and object relative clauses, in which there is a significant tendency for accusative prepositional constructions in subject relative clauses ($p<0.001$), while there is no significant tendency for positional order choices in object relative clauses conditions ($p>0.05$).

Index Terms line graph Chart algorithm, syntactic analysis model, Chinese relational clauses, automatic clause modeling

I. Introduction

Relative clauses are a kind of clause clauses, and the central noun modified by the clause also has a syntactic position in the clause [1]. Chinese relational clauses can co-occur with “indicative (+ numeral) + quantifier”, and the co-occurrence involves the issue of the positional order of the two [2]. As a rule, the order of Chinese is “subject + predicate + object” (SVO), which is the same as that of most Indo-European languages such as English and German [3], [4]. However, the difference is that Chinese relational clauses are placed before the center word, which is relatively rare in the typology, and is not consistent with the configurational model of the inflectional order and the position of relational clauses [5], [6].

The usual scholarly concern is with the place order of subject-relative clauses and object-relative clauses. Subject-relative clauses extract the subject, and the central noun refers to the trace left by the subject, while object-relative clauses extract the object of the transitive predicate verb, and the central noun refers to the trace left by the direct object [7]-[9]. Therefore, it is not possible to account for the positional ordering patterns of relational clauses co-occurring with “referential quantities” using speech processing strategies alone [10]. In addition, existing sentence processing theories are mainly based on Indo-European languages that conform to the type of inflectional configurations, and thus Chinese relational clauses can be used as an ideal structure to test sentence processing models that make different predictions about them, in order to narrow down the scope of application of theoretical hypotheses [11]-[13]. Therefore, it is of great significance to study the evolution of Chinese relational clauses, which can provide support for the processing advantages of relational clauses.

In this paper, we firstly describe the arithmetic method and process of the stack-maximum matching automatic word segmentation model established under the combination of the maximum matching method (MM method) and the stack technology as an automatic word segmentation method for Chinese relational clauses. At the same time, the lexical and syntactic analysis steps of Chinese relational clauses are designed with the support of existing natural language processing technologies. Subsequently, the Chart algorithm is chosen as the syntactic analysis algorithm, and the implementation process, advantages and defects of the algorithm are described. The syntactic analysis model of Chinese relational clauses is constructed. Finally, the model is used to analyze and explore relational clauses embedded in the subject of main clauses, the processing advantages of relational clauses, and the distribution of utterance types.

II. A model for syntactic analysis of Chinese relational clauses

II. A.Basic concepts

Definition 1: A fuzzy context-independent grammar is a quaternion $FG=(V_N,V_T,P,S)$ where V_N is a non-empty non-terminal set of exhaustive alphabets. V_T is an exhaustive alphabet set of non-empty terminators. The intersection of V_N and V_T is empty, $S\in N$ is the start symbol, and P is the set of generators shaped like $A\xrightarrow{\mu}\beta$ ($A\in V_N$, $\beta\in(V_N\cup V_T)^*$, and $\mu\in[0,1]$ is the degree of affiliation of the generator).

Definition 2: Fuzzy context-independent language The language produced by a fuzzy context-independent grammar is denoted as Eq. (1):

$$L(FG)=\left\{\left(x,\mu(x)\right)\middle|x\in V_T^*,S\xrightarrow{\mu}x,\mu(x)=\vee\left(\mu_1\wedge\mu_2\wedge\cdots\wedge\mu_k\right)\right\} \tag{1}$$

Definition 3: Fuzzy context-independent grammar derivation rule.

If $A\xrightarrow{\mu}\beta$, then for $\forall\delta,\gamma\in(V_N\cup V_T)^*$ there is a $\gamma A\delta\xrightarrow{\mu}\gamma\beta\delta$, saying that $\gamma\beta\delta$ is a fuzzy derivation of $\gamma A\delta$ and $A_1\xrightarrow{\mu_1}A_2\xrightarrow{\mu_2}A_3\xrightarrow{\mu_3\cdots\xrightarrow{\mu_{n-1}}}A_n$ is the derivation of the fuzzy generators $r_1,r_2,\cdots,r_{n-1}$ from A_1 to A_n.

Definition 4: A fuzzy context-independent grammar is in Chomsky paradigm form if all its rules satisfy one of the following conditions:

(1) $A\xrightarrow{\mu}BC$

Or,

(2) $A\xrightarrow{\mu}a$.

where $A,B,C\in A,B,C\in V_N,a\in V_T$.

Any fuzzy context-independent grammar can be converted to a Chomsky paradigm form.

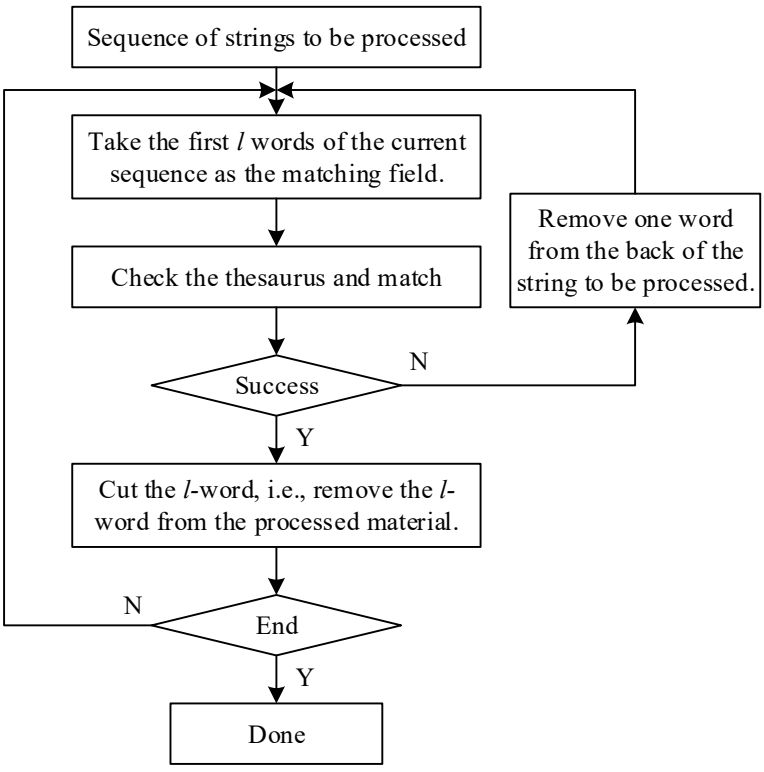


Figure 1: The maximum matching method process

II. B.Stack-Max Match Automatic Segmentation Modeling

Stack-Maximum Matching Automatic Word Segmentation technique mainly combines the Maximum Matching Method (MM method) and the stack technique to automatically slice the words in the article, which is an improvement of the Maximum Matching Method.

The main idea of the Maximum Matching Method is to retrieve the word in the thesaurus that has the longest character length that can be successfully matched with the original sentence as the result of word segmentation. Its process is shown in Fig. 1.

Maximum matching method attaches importance to the length of characters, if encountered in the process of word splitting the string behind the indivisible situation, can automatically pop the stack back, and re-retrieve another successful match of the word as the result of word splitting, it is possible to solve the dilemma of the indivisible string behind the dilemma. The basic design idea is:

First of all, according to the punctuation in the article will be divided into semantic blocks, each semantic block is a string, for each string as a loop. Each time, only one Chinese character is processed, the Chinese character is assumed to be the word head, and the character matches in the thesaurus are retrieved with the Chinese character as the word head, and the character matches after the Chinese character is retrieved. If the string formed by the Chinese character and the characters following it in the semantic block exists in the thesaurus, that is, the string can form the form of a word, then all the words that meet this condition are retrieved. According to the retrieved words as the alternatives of the segmentation result, arranged by length, the word with the longest length, i.e., the maximum match, is firstly taken out, assuming that this word is the segmentation result headed by this Chinese character, and is added to the stack of segmentation result of this semantic block, and then the processing of the next Chinese character after the position of this word is continued.

Following this kind of way to continue to split words in the semantic block, if we encounter a word starting with a certain Chinese character that does not exist in the lexicon, or a word that exists in the lexicon does not match the string starting with that Chinese character in the semantic block, it is assumed that there is an ambiguity in the first half of the splitting, which creates a problem. So the result of the last stack is popped from the result stack of the semantic block. If there are other alternatives for the last participle result, the word with the second longest length among the sub-alternatives is taken as the participle result and stacked. If there is only one alternative in the last result, then continue to pop the stack. If there is a situation that all the participle results are popped up, it means that the Chinese character that cannot be found in the thesaurus is not entered into the thesaurus, so it should be skipped first, and when all the other characters are finished, the user will judge the correct participle result, and then it will be entered into the thesaurus again.

In the process of implementing this method, the author removes the part of the string in the semantic block that has been successfully disambiguated from the semantic block while pressing the stack, that is to say, the semantic block is processed from the beginning of the string for character matching every time. If there is a disambiguation in the result stack that needs to be popped, the popped result will be added to the first part of the original semantic block string. This eliminates the need to calculate the position of the next Chinese character to be processed after each disambiguation result.

The following is the stack-max-match automatic participle model.

Suppose there exists an article model $T=\sum_{i=0}^{n} b_i p_i$, with b_i denoting semantic chunks and p_i denoting intervals of punctuation. $b_i=\sum_{j=0}^{m} a_j$, that is, each semantic block $b_i=a_0a_1\cdots a_m$. a_j denotes a single Chinese character in b_i, and a_0 denotes the first character of this semantic block. There is such a set to represent the set of words, i.e., the lexicon as in equation (2):

$$w_i \in Q\left\{w_i \middle| i=0,1,\cdots\right\} \tag{2}$$

where $w_i=\sum_{i,j=0}^{n} c_j$. During character matching, $\sum_{j=0}^{k} a_j$ is tested for matching with $\sum_{i,j=0}^{n} c_j$, where $0\le k\le m$, and if a match is formed, i.e., $\sum_{j=0}^{k} a_j=\sum_{i,j=0}^{n} c_j$, and $n=k$. Then the result of the match, that is, w_i in the set of words, is added to the set of alternatives E in the result of word splitting as in equation (3):

$$E=\left\{w_i\left|\sum_{j=0}^{k}a_j=\sum_{i,j=0}^{n}c_j\right.\right\} \tag{3}$$

Also remove $\sum_{j=0}^{k}a_j$ from b_i and regenerate b_i. The set R is the set of partitioning results as in equation (4):

$$R=\left\{w_i\left|\sum_{j=0}^{k}a_j=\sum_{i,j=0}^{n}c_j\right.\right\} \tag{4}$$

If the result of the word separation alternative set $E=\varnothing$, first determine whether a_0 exists in the set as in equation (5):

$$U=\left\{c_i\left|u_i=c_i\sum_{i=0}^{n}b_i,u_i\notin Q\right.\right\} \tag{5}$$

If $a_0\notin U$, then a_0 is added to U, and then the string w_{k-1}, which was last added to the participle result set R, needs to be taken out of the R set and added to the first part of b_i. If $a_0\in U$, then $b_i=\sum_{j=1}^{k}a_j$, omitting a_0.

If there is only one element in the participle result alternative set E, it means that the element is the result of the relevant participle of the Chinese character and is added to the result set R.

If there is more than one result in the participle result alternative set E, the word with the longest length in the set E is selected as the participle result and added to the set R, where $w_i=\sum_{i,j=0}^{n}c_j$, where n is the maximum value in the set E.

In the algorithm description, E is applied as an array. Apply R in the form of a stack.

II. C.Parsing of grammatical points

II. C. 1) Lexical analysis

Lexical analysis is the extraction of grammatical knowledge points for a particular word in a sentence, and the inputs are the six-tuple word information tokenInfo for the word and the sentence token traverser sentObject, where the tokenInfo tuple can be obtained by index number 0-5 for id, word, lemma, pos, and relation, parent value. The specific processing steps for lexical analysis are as follows:

(1) Read in the six-tuple wordInfo $tokenInfo$ and $token$ traverser $sentObject$.

(2) Grouping according to part-of-speech marker categories. The lexical categories in the grammatical knowledge graph correspond to the part-of-speech annotation results, and in addition to the correspondence of the included part-of-speech annotation results, other part-of-speech annotation results correspond to their meanings themselves, such as numeral words corresponding to part-of-speech labeling "CD", prepositions corresponding to "IN", conjunctions corresponding to "CC", etc.

(3) Specific and detailed analysis of different lexical knowledge points.

Among them, the corresponding morphological changes of words are mainly used for the output of grammatical phenomena similar to the rules of plural changes of nouns, comparative and supreme changes of adjectives or adverbs, etc. The parsing result is given by the string comparison between the word itself $word$ and its original form $lemma$.

In the processing of word utility parsing, $tokenInfo[3]$ is the result of lexical tagging of the word pos, and $sentObject[tokenInfo[5]-1].tag\_$ is the result of lexical tagging of the word's dependency $token$, and the combination of both of them can realize the word utility parsing, e.g. $tokenInfo[3]==$“RB”, $sentObject[tokenInfo[5]-1].tag\_\in\{$“$VB$”,“$VBD$”,“$VBN$”$\}$, then we can get that the word is “adverbial function→adverbial modifier verb”.

Recognition of fixed grammatical expressions, such as the sentence “It's ten o'clock.”, contains a grammatical point - the representation of the moment that represents the whole point: the base word $+oclock$, in this paper, we use the grammatical pattern (id) pos: $CD+(id+1)word:o.clock$ recognition, that is, id number of two

consecutive words, the former word of the lexical annotation results for the number of words, the latter word for $o.clock$, on the output of the grammatical phenomenon - indicates the whole point of the moment representation, other fixed grammatical collocation is the same.

Word grouping refinement is used in cases where the knowledge graph contains sub-nodes, but needs to be refined according to its own word grouping. In this paper, we utilize grammar-specific marker words to identify refinement. When the sentence contains a corresponding marker word, similar to fixed expression recognition, the marker word grammar collocation is examined for recognition.

II. C. 2) Syntactic analysis

The syntactic analysis process identifies grammatical phenomena through punctuation, various types of signifiers, dependency relations, and word antecedents, and the process is carried out according to the types of syntactic knowledge points without any sequential relationship. This section introduces the processing of core syntactic points.

(1) Sentence tense recognition. If $\exists i \in \left[0, len\left(sentObject\right)\right)$, $s.t.\ sentObject[i].tag\_ \in \{"MD",\ "VBD",\ "VBP",\ "VBZ"\}$, then the discernment of tense is carried out, followed by the discernment of tense according to the $sentObject[i].lemma\_$ belonging to the word $\{be, have, has, had, will, will, would, would, could, might\}$ for refinement. Such as $sentObject[i].lemma\_ ==$“ be”, $sentObject[i+1].text ==$“ $going$”, $sentObject[i+2].text ==$“ to” and $sentObject[i+3].tag\_ ==$“ VB”, then it is future tense. Further refinement is based on $sentObject[i].text \in \{$“ am”,“ is”,“ are”,“ m”,“ s”,“ re”$\}$ or $sentObject[i].text \in \{$“ was”,“ $were$”$\}$ is determined to be the general future or past future tense. Other tenses are refined in the same way, based on the form or lexical markers of the word and its collocated words.

(2) Judgment of basic sentence types in syntactic knowledge points: Initialize an array 1 with length 9 and all 0 values, and different index values of 0 or l indicate the presence or absence of the subject, tethered verb, predicate, object, direct object, indirect object, object complement, object constituent after the preposition, and the sentence type of “there be”, respectively. Secondly, traverse the data preprocessing module output within the scope of a single sentence word six tuple list $words$, according to the dependency of each word to determine whether it contains the components of l, if so, set the corresponding index bit of the tuple l value of 1. After the end of the traversal, according to the combination of the index bit of the l value of 1, to determine the sentence belongs to the basic sentence type.

(3) Sentence type segmentation. According to $sentObject[len(sentObject)-1].text$ to get the sentence punctuation, followed by the type of punctuation according to the coarse division of sentence types, such as “.” and “!” belong to declarative sentences, “?” are interrogative sentences. After the rough division of different sentence refinement process, this paper mainly to special question sentences, imperative sentences as an example to illustrate.

Recognition of special interrogative sentences. A special interrogative sentence is a special interrogative sentence if it is under $sentObject[0].tag- \in \{$“ WDT”,“ WP”,“ WP”$\}$. The second refinement is based on the value $\{what, which, who, whose, when, where, why, hou\}$ taken by the lead $sentObject[0].teat$ of the sentence. Under the recognition of each leading word, the grammatical knowledge graph still needs to be subdivided, further subdivided according to $sentObject[1].teact$, such as $sentObject[0].teat ==$“ $what$”, $sentObject[1].teat ==$“ $else$”, then what else special interrogative sentence, if $sentObject[1].teact ==$“ $color$”, then what color special interrogative sentence, and the same for other special interrogative sentences such as size, time, day, and so on.

Identification of the imperative sentence: statement under $sentObject[0].tag\_ ==$“ VB”, the sentence belongs to the imperative sentence, further refinement of the way need to be based on $sentObject[0].lemma\_$ specific word refinement, such as $sentObject[0].lemma\_ ==$“ let” and $sentObject[1].tag\_ ==$“ PRP”, then it is in the form of the imperative sentence let's. The imperative be+adj. form and the imperative do (sth.) form are recognized in a similar way.

In addition, syntactic analysis also includes other sentence kinds of subdivision, but the refinement rules formulated vary depending on the syntax of the sentence, therefore, here in this paper, we only show some of the syntactic refinement rules.

II. D.Line Chart Algorithm

Line graph grammar analysis is the most widely used, simple to implement and easy to understand analysis algorithm among grammar analysis algorithms. A line graph is an acyclic directed graph that consists of such a set of nodes and edges by taking the intervals between words as nodes and words and phrases as edges connecting the nodes.

Process description of the traditional Chart algorithm:

The string to be analyzed, w, is placed in the input buffer and the schedule (Agenda) is the empty stack. Loop and repeat the following steps until both the input buffer and Agenda are empty.

(1) If Agenda is empty, remove a character from the input buffer and press the character and its start and end positions $(P1, P2)$ into the Agenda stack.

(2) Pop the edge at the top of the stack from the Agenda stack, which starts and ends at $(P1, P2)$, and which is labeled L.

(3) Examine the rules in the rule set, and for all rules of the form $A \to LB$, add a point rule in the set of active edges with start and end positions P1, P2, and $A \to L \bullet B$ on the arc.

(4) Add the edge labeled L that popped out of Agenda between P1, P2 in the line graph.

(5) Examine the set of all active edges, and if there exists an active edge with start and end positions P0, P1 and a point-on-arc rule of $A \to a \bullet LB$, a new active edge is added to the set of active edges with start and end positions P0, P2 and a point-on-arc rule of $A \to aL \bullet B$.

(6) If an active edge (starting and ending positions P0, P2) has an on-arc point rule shaped like $A \to aL \bullet$ (the point number is at the rightmost end of the rule), the edge with starting and ending positions P0, P2 and edge labeled A is pressed into the Agenda stack.

Advantages and disadvantages of Chart's algorithm:

Line graphs can represent unconnected subtrees. In the analysis of natural language, sometimes the local structure is successfully analyzed, but the overall structure is not well analyzed, making it difficult to form a complete tree in the end, the line graph can represent the unconnected sub-trees, it is not necessary to require that a complete tree must be formed in the end, so the local analysis of the correct structure in the form of a sub-tree can be preserved, so as not to waste the analysis of the previous.

The line graph algorithm is intuitive and can represent words that have more than one interpretation. In a line graph, if a word has multiple interpretations, these multiple interpretations can be represented as multiple edges, thus making the ambiguity problem clear. The line graph algorithm is flexible and can be improved more easily by modifying some control strategies in the analysis.

The line graph algorithm also has some shortcomings, the analysis is inefficient and the accuracy of the analysis is reduced due to more redundancy generated by backtracking. Line graph analysis cannot avoid the problem of generating multiple analyzed syntax trees for a single sentence.

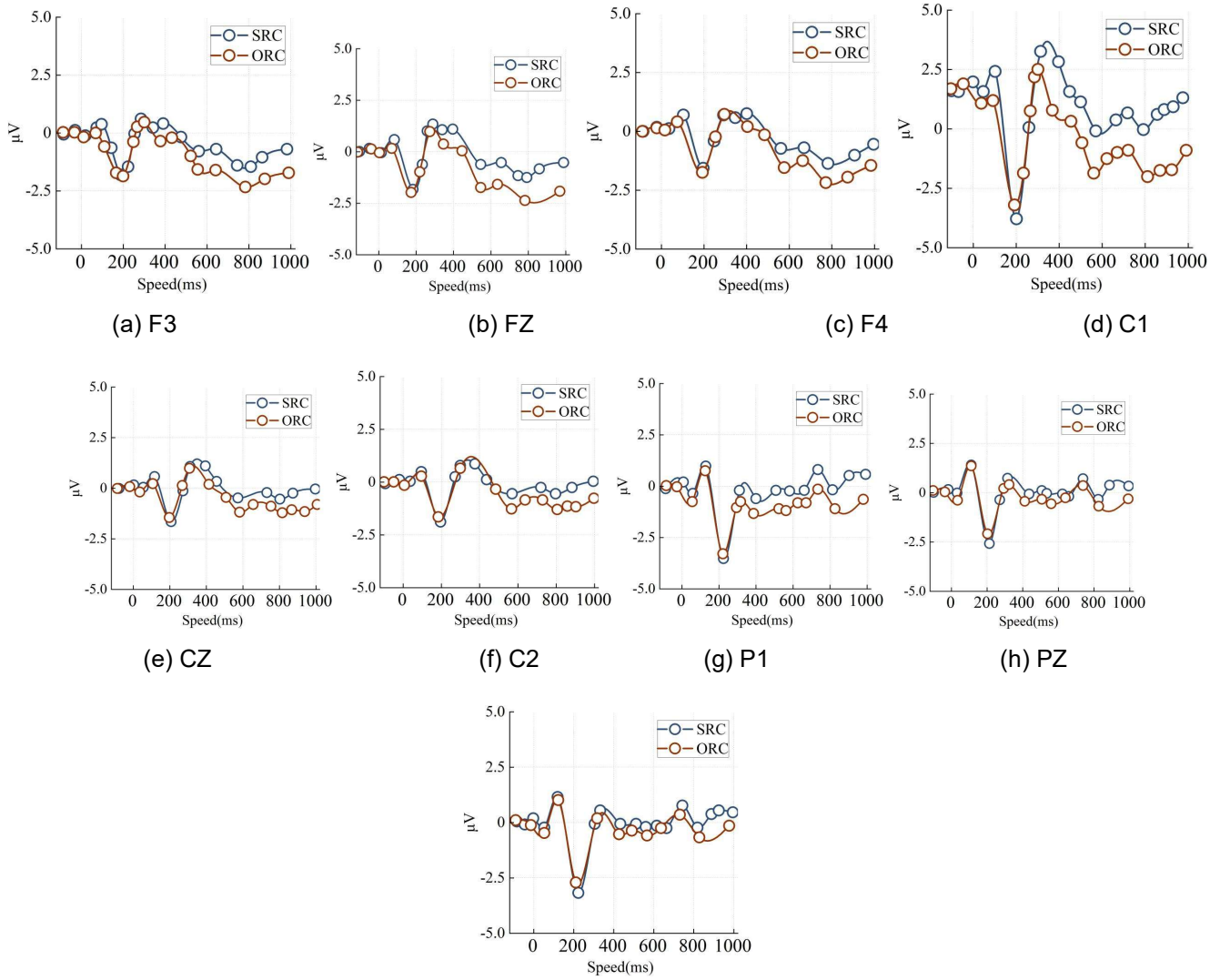
III. Processing Advantages and Distributional Structure of Relational Clauses

III. A. Analysis of relational clauses embedded in the subject of the main clause

In this section, subject-relative clauses (S-SRC) and object-relative clauses (S-ORC) are selected as experimental materials, and the proposed model is used to explore the processing advantages of Chinese subject-object relative clauses with the subject-object relative clauses embedded in the subject of the main clauses as stimulus materials.

III. A. 1) SRC Subjunctive Verbs and ORC Subjunctive Nouns

The first word block of the S-SRC clauses was the subordinate verb, and the first word block of the S-ORC clauses was the subordinate subject.The waveforms of the subordinate verb of the S-SRC versus the subordinate noun of the S-ORC are shown in Fig. 2, and the differences between the two types of stimuli were mainly centered on 350-450 ms (N400).



(a) F3 (b) FZ (c) F4 (d) C1

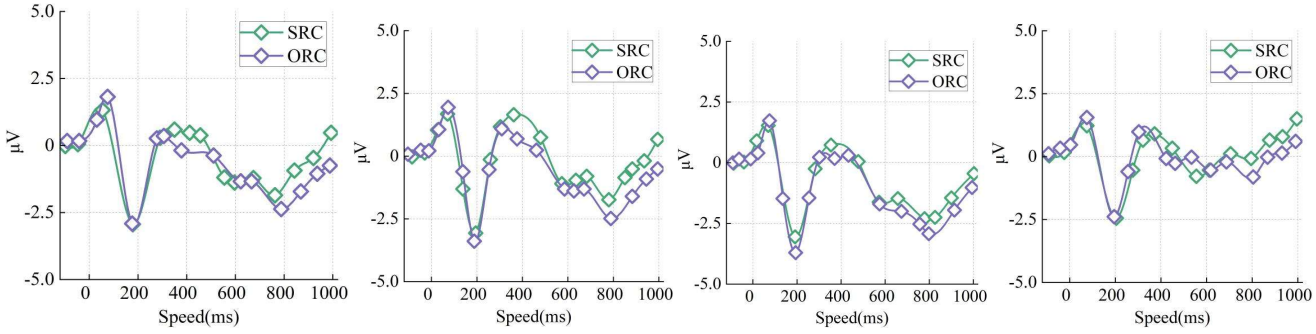
(e) CZ (f) C2 (g) P1 (h) PZ

(i) P2

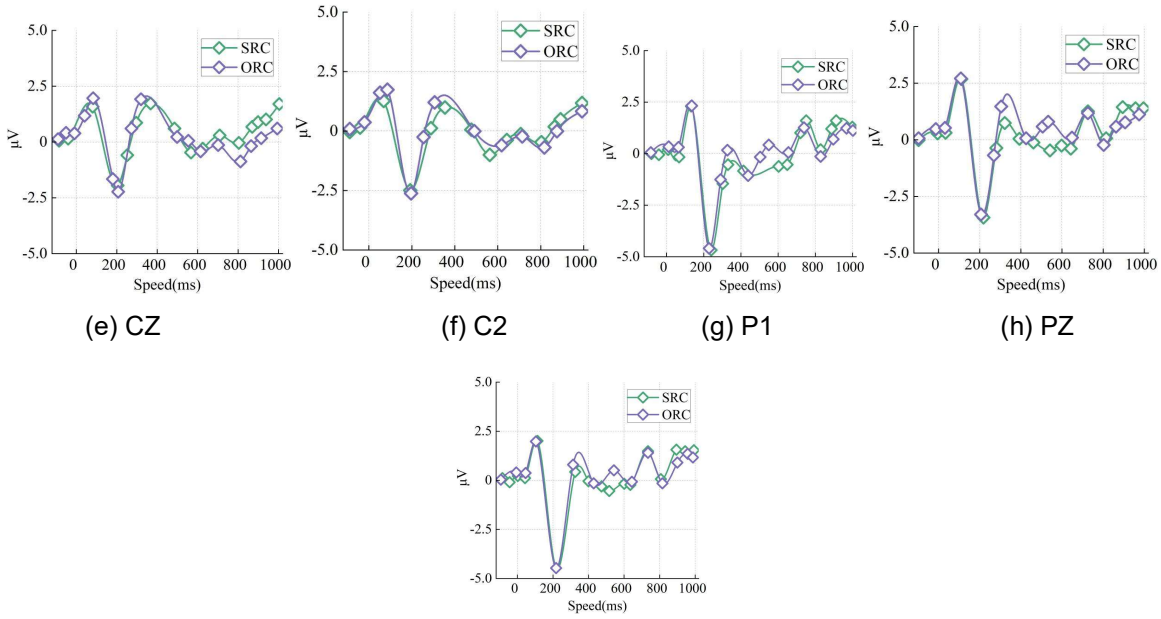
Figure 2: The verb in the S-SRC clause and the noun waveform in the S-ORC clause

III. A. 2) SRC Subjunctive Nouns and ORC Subjunctive Verbs

The second word block of S-SRC clauses is the object of the clause, and the second word block of S-ORC clauses is the verb of the clause.The SRC clause noun and ORC clause verb waveforms are shown in Fig. 3, and the difference between the two types of stimuli is mainly centered on 350-450 ms (N400).



(a) F3 (b) FZ (c) F4 (d) C1



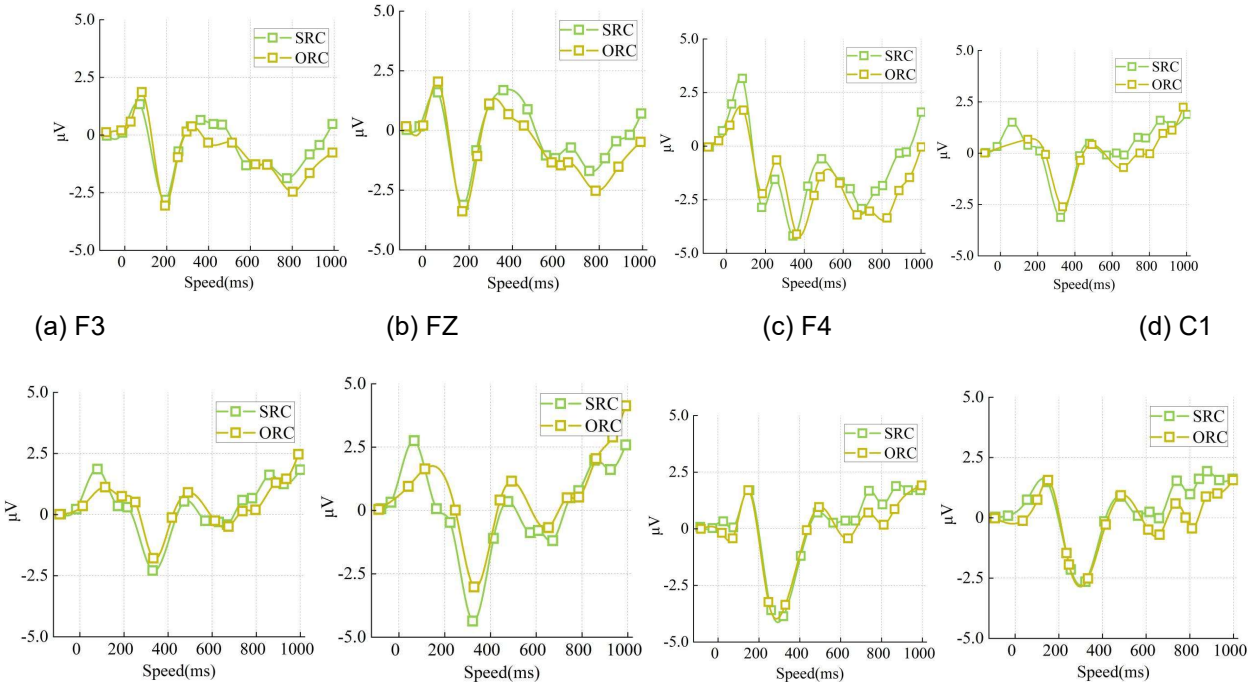
(e) CZ (f) C2 (g) P1 (h) PZ

(i) P2

Figure 3: The noun in the S-SRC clause and the verb waveform in the S-ORC clause

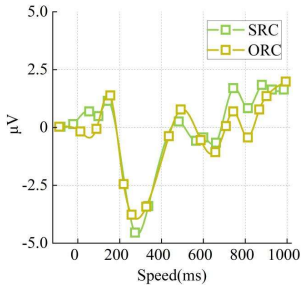
III. A. 3) Marking of relational clauses: "of"

The third lexical block of S-SRC and S-ORC is the relational clause marker.The S-SRC clause marker and S-ORC clause marker waveforms are shown in Fig. 4, and the difference between the two types of stimuli is mainly centered on 600-800 ms (P600).



(a) F3 (b) FZ (c) F4 (d) C1

(e) CZ (f) C2 (g) P1 (h) PZ



(i) P2

Figure 4: The clause marking of S-SRC and the clause marking waveform of S-ORC

It can be seen from the above analysis that when the relational clauses are embedded in the subject position, the processing difficulty of the Chinese subject relational clauses is lower than that of the object relational clauses. Therefore, clause position should be included as one of the factors to be considered in the analysis and research of Chinese relational clauses.

III. B. Processing characteristics of relational clauses

In this section, a reading eye-tracking experiment paradigm was used to further examine the real-time processing characteristics of Chinese subject and object clauses based on this paper's model by adding a subsequent clause after the relational clauses and introducing a personal pronoun to refer back to the middle word of the relational clauses.

The analysis results of this paper's model showed that there was no significant difference in accuracy between the two subordinate clause conditions (b = -0.02, SE = 0.21, Z = -0.08). Table 1 presents the results of the analysis of this paper's model under the eye movement metrics for each region of interest, where the eye movement metrics are (EM1) first reading time, (EM2) retrospective path time, (EM3) second reading time, and (EM4) total reading time, and the selected regions of interest are (A1) the center word, (A2) the region after the center word, (A3) the first region after the pronoun, (A4) the pronoun after the second region, (A5) pronoun merged with the latter region.

Table 1: Linear mixed-effects model

Regions of interest	Eye movement index	b value	Standard error	t value
A1	EM1	0.04	0.04	-0.96
	EM2	-0.06	0.06	-1.05
	EM3	<0.05	<0.07	0.64
	EM4	<0.000	<0.05	0
A2	EM1	-0.02	0.05	-0.66
	EM2	-0.003	0.06	-0.19
	EM3	-0.26	0.06	-4.51
	EM4	-0.11	0.07	-2.92
A3	EM1	<0.0006	<0.05	-0.06
	EM2	<-0.005	<0.05	-0.14
	EM3	-0.03	0.07	-1.07
	EM4	-0.05	0.05	-1.43
A4	EM1	-0.06	0.03	-1.6
	EM2	-0.08	0.07	-1.87
	EM3	-0.00	0.07	-0.27
	EM4	-0.07	0.05	-1.38
A5	EM1	-0.007	0.05	-0.25
	EM2	0.005	0.06	0.05
	EM3	-0.09	0.06	-1.43
	EM4	-0.03	0.03	-2.22

Observation of Table 1 reveals that:

(1) In the region of (A1) center word, there is no significant effect of all indicators.

(2) In (A2) the first region after the center word, the second reading time and the total reading time of subject-relative clauses are shorter compared to object-relative clauses, indicating that subject-relative clauses have processing advantages.

(3) In the first region after (A3) pronouns, there was no significant effect for all indicators.

(4) In the second region after the (A4) pronoun, a trend toward a processing advantage for subject-relative clauses in retrospective path time was shown, but did not reach statistical significance.

(5) In the combined region of (A5) pronouns and the latter, the total reading time of subject-relative clauses was shorter compared to object-relative clauses, indicating a processing advantage for subject-relative clauses.

III. C. Distribution of statement types

The purpose of this section is to explore the distribution of different types of relational clauses when co-occurring when referring to quantifiers and the object position of the main clause. In the subject relational clause condition, all sentences were answered, generating a total of 420 sentences. While there was one case of unanswered sentences in the object relational clause condition, a total of 399 sentences were collected. Table 2 is plotted according to the coding of the actual output sentences, and the following are the statistical results of each data. In the case of subject-relative clauses, “S1” is used to refer to them, and object-relative clauses, “S2” is used to refer to them, and the types of sentences generated are categorized as: completely correct, legal but not in accordance with the requirements, and incorrect. The type of “legal but non-compliant” is further subdivided into: (LN1) relational clause positional change, (LN2) relational clause constituent deletion, (LN3) quantifier neutralization, (LN4) quantifier deletion, (LN5) relational clause deletion, (LN6) subject-subject constituent substitution, and (LN7) morpheme shift. The types of “errors” are further subdivided into (ER1) incomplete sentences, (ER2) unqualified grammar.

Table 2: The types and distribution ratios of the generated sentences

		S1		S2	
The type of generated sentence		N	Proportion(%)	N	Proportion(%)
Absolutely right		415	98.81	386	96.74
Legal but not in compliance with the requirements	LN1	0	0.00	4	1.00
	LN2	0	0.00	0	0.00
	LN3	2	0.48	0	0.00
	LN4	1	0.24	3	0.75
	LN5	0	0.00	2	0.50
	LN6	1	0.24	0	0.00
	LN7	0	0.00	1	0.25
Error	ER1	0	0.00	1	0.25
	ER2	1	0.24	2	0.50
Total		420	100	399	100

Analyzing the overall percentage of correctness, the percentage of correctness of (S1) subject-relative clauses (98.81%) was slightly higher than that of (S2) object-relative clauses (96.74%). Among the “legal but not conforming” sentence types, the proportion of (S1) subject-relative clauses (0.96%) was lower than that of (S2) object-relative clauses (2.5%). Among the “wrong” sentence types, the proportion of (S1) subject-relative clauses (0.25%) was slightly lower than that of (S2) object-relative clauses (0.50%). Overall, the proportion of non-fully correct sentences in the (S1) subject-relative clause condition (1.24%) was slightly lower than in the (S2) object-relative clause (3.25%).

III. C. 1) Distributional positional order of quantifiers in subject-object relative clauses

After the non-compliant or incorrect sentences were screened out, statistics on the distribution of referents in a total of 801 completely correct sentences were carried out, and the distribution dynamics of different types of relational clauses in referent prepositional and post-positional structures are shown in Fig. 5.

Among the 420 cases (S1) of subject-relative clauses, there were 331 cases of preterite clauses referring to quantifiers, with a share of 79.76%. There are 84 cases of postpositions, and the share is 20.24%. Among the 399 cases of (S2) object-relative clauses, there are 206 cases of preterite clauses referring to quantifiers, with a share of 53.37%. There are 180 cases of postpositions, with a share of 46.63%. It is not difficult to find that there is a significant tendency for referring to quantifiers to be prepositioned in subject-relative clauses ($p<0.001$). However,

there was no significant tendency (p>0.05) for the positional order choice of referring quantifiers in the object-relative clause condition.

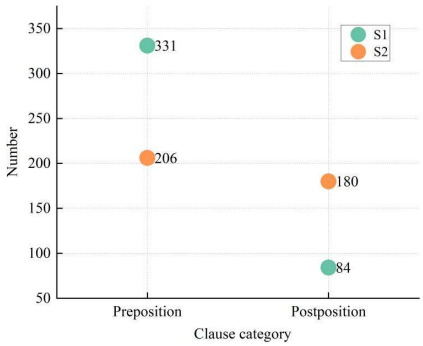


Figure 5: The distribution of different types of relative clauses

III. C. 2) Distribution of different types of relational clauses in referential constructions

The distributional dynamics of the number of different types of relational clauses in the prenominal and postnominal constructions of the referent is shown in Figure 6.

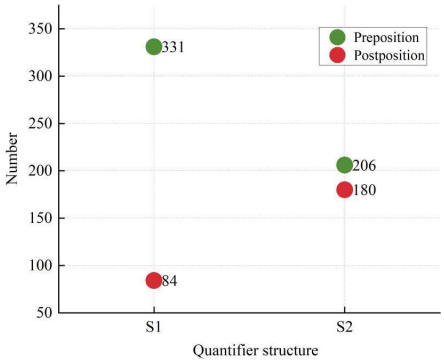


Figure 6: The distribution of the number of different types of relative clauses

By comparing the number of accusative prepositions and postpositions, it can be found that among the 537 cases of accusative prepositions, there are 331 cases of (S1) subject-relative clauses, with a weight of 61.64%, and 206 cases of (S2) object-relative clauses, with a weight of 39.36%, which indicates that the prepositional effect of accusative prepositions is more significant when they co-occur with (S1) subject-relative clauses (p<0.001). As for the 264 cases of referent-measure postpositions, there were 84 cases of (S1) subject-relative clauses, with a proportion of 31.82%, and 180 cases of (S1) object-relative clauses, with a proportion of 68.18%, indicating that referent-measure postpositions are more inclined to be generated in object-relative clauses than in subject-relative clauses (p<0.001).

IV. Conclusion

In this paper, a grammatical analysis model of Chinese relational clauses is established by integrating the stack-maximum matching automatic analysis model and the line graph Chart algorithm, and designing the steps of parsing grammatical knowledge points based on natural language processing technology.

Using the grammatical analysis model of Chinese relational clauses, the processing advantages of Chinese subject and object relational clauses as well as the structural distribution of accusative case are successively unfolded. In the relative clauses embedded in the main clause advantage, the differences between the stimuli of the subject relative clause verb and the object relative clause noun, as well as the stimuli of the subject relative clause noun and the object relative clause verb, were mainly concentrated in the range of 350-450 ms, and the second reading time and the total reading time of the subject relative clause were shorter compared with those of the object relative clause, which indicated that the subject relative clauses had a greater processing advantage than the object relative clauses. The advantage of subject clauses over object clauses in processing. In the

structural distributional order, there was a significant tendency for quantifiers to be anterior in subject-relative clauses ($p<0.001$), but posterior structures were more likely to be produced in object-relative clauses ($p<0.001$).

V. References

[1] Hesamoddin, S., Farzaneh, S., & Ahmad, A. (2018). An examination of relative clauses in argumentative essays written by EFL learners. Journal of Language and Education, 4(4 (16)), 77-87.

[2] Wu, F., Kaiser, E., & Vasishth, S. (2018). Effects of early cues on the processing of Chinese relative clauses: Evidence for experience-based theories. Cognitive science, 42, 1101-1133.

[3] Mansbridge, M. P., Tamaoka, K., Xiong, K., & Verdonschot, R. G. (2017). Ambiguity in the processing of Mandarin Chinese relative clauses: One factor cannot explain it all. PloS one, 12(6), e0178369.

[4] Bulut, T., Cheng, S. K., Xu, K. Y., Hung, D. L., & Wu, D. H. (2018). Is there a processing preference for object relative clauses in Chinese? Evidence from ERPs. Frontiers in Psychology, 9, 995.

[5] Chang, L. P., & Center, M. T. (2017). The acquisition of relative clauses in L2 Chinese: A corpus-based study. Journal of Chinese Language Teaching, 14(1), 47-80.

[6] Kwon, N., Ong, D., Chen, H., & Zhang, A. (2019). The role of animacy and structural information in relative clause attachment: evidence from Chinese. Frontiers in psychology, 10, 1576.

[7] Lau, E., & Tanaka, N. (2021). The subject advantage in relative clauses: A review. Glossa: a journal of general linguistics, 6(1).

[8] Sichel, I. (2018). Anatomy of a counterexample: Extraction from relative clauses. Linguistic Inquiry, 49(2), 335-378.

[9] Cunnings, I., & Fujita, H. (2023). Similarity-based interference and relative clauses in second language processing. Second Language Research, 39(2), 539-563.

[10] Xia, V. Y., White, L., & Guzzo, N. B. (2022). Intervention in relative clauses: Effects of relativized minimality on L2 representation and processing. Second Language Research, 38(2), 347-372.

[11] Graf, T., Monette, J., & Zhang, C. (2017). Relative clauses as a benchmark for Minimalist parsing. Journal of Language Modelling, 5(1), 57-106.

[12] De Vries, M. (2018). Relative clauses in syntax. In Oxford Research Encyclopedia of Linguistics.

[13] Kou, X. (2019, June). The restrictions on the genitive relative clauses triggered by relational nouns. In Workshop on Chinese Lexical Semantics (pp. 746-752). Cham: Springer International Publishing.