International Journal of Knowledge
and Language Processing
Volume 4, Number 3, **2013**

www.ijklp.org
KLP International ©2013 ISSN 2191-2734
pp.27—37

A Rule-Based Mongolian Dependency Parsing Model

S. Loglo and Sarula

College of Mongolian Studies, Inner Mongolia University, Huhhot,
Inner Mongolia Autonomous Region 010021,China
E_mail:sloglo@sina.com

Received May 2013; revised July 2013

Abstract. *In this paper, we describe the design and implementation of a Mongolian dependency parsing model using MDTB as training and evaluation data. The model used the complex features and unification operations. Currently, the model achieves accuracies of 75.21%、69.39% and 75.21% for the tasks of unlabelled annotation, labeled annotation and head word annotation respectively.*

Keywords: Mongolian language, dependency grammar, parsing

1. **Introduction.** In respect of building Mongolian language corpora, the Institute of Mongolian Language in Inner Mongolia University has built 1 million words of modern Mongolian corpus in 8 years (1984-1991), and the corpus has expanded to 10 million words. The 1 million words of modern Mongolian corpus include four kinds of materials: novels, textbooks, newspapers and political articles. The respective portions are: 19.6%, 50.3%, 9.8%, and 22.9%. In respect of corpus processing, the institute has completed a series of basic tasks, such as, the POS tagging and the processing of the set phrases in the 1 million words of modern Mongolian corpus, and research on the tagging of phrases, the automatic segmentation of sentences, and the automatic recognitions of the predicate parts in sentences. At present, the institute is studying the parsing of whole sentences and the tagging of semantic roles with the support of NSF, and has constructed the Mongolian dependency Treebank (MDTB) of about 500,000 words using the method of automatic parsing and manual correction, which provides the Mongolian syntactic parsing with the language information data for training and testing. In this paper, we are going to discuss in detail the rule-based Mongolian dependency parsing model which is used in the process of building the Mongolian dependency Treebank.

2. **Description Systems of the Rules.** Referring to the algorithms of rule-based parsing in English, German and Chinese, and according to being rich in the morphological features in Mongolian language, this paper propose a description system of the rules for the Mongolian dependency parsing based on the complex features and unifications.

Each rule of the rule set is a rule of generating the dependency arc. The condition part of a rule consists of two items of the features of the scanned nodes and the constraint conditions, while its action part includes two items of the generation of dependency relations and the unification of feature sets. The dominance relationship is decided by placing restrictions on the values of the features of the scanned nodes and the scanned range of the nodes. A whole dependency tree for one sentence of n words includes n nodes and n-1 dependency arcs. Hence, the process of automatic parsing needs n-1 production rules to generate n-1 dependency arcs.

The rules of dependency can be expressed as follows: $POS_i POS_j \Rightarrow POS_i \xrightarrow{rel} POS_j$

POS_i and POS_j express the POS information of the two words that will form the relations of dependency, $\Rightarrow$ stands for the generation, $\longrightarrow$ indicates that POS_i depends on POS_j and rel shows the types of dependency relations.

This form of production uses the expression of the single feature, which is POS information. In the practical parsing, the POS information is too rough to place effective restrictions on the generation of actions. For example, the sequence of N V (N is for noun, V is for verb) can generate at least three kinds of dependency relations. BATV (human name, N) and IREBE (came, V) constitute the subject-predicate relation, BVDAG_A (rice, N) and IDEBE (ate, V) constitute the direct object-predicate relation, ORLOGE (morning, N) and YABVBA (went, V) constitute the adverbial-predicate relations.

Then, what kind of rules can basically avoid generating such ambiguity? The form of the most ideal rules are as following: $W_i W_j \Rightarrow W_i \xrightarrow{rel} W_j$

W_i and W_j express the two words in the relations of dependency. In fact, there is not ambiguity in language on the lexical level, except Homonym and polysemous words. Currently, this kind of rule system can only be implemented in the text with a small vocabulary, but there is no way to implement it in the real text because it is impossible to have the relations of dependency between any two words in a real context of the language. The quantity of the rules is infinite if we write out a rule of generation for a pair of words that may include dependency relations. So, it is necessary and possible to create a rule system, which is between the two forms of rule systems discussed above, with properly detailed features and constraints for the practical parsing in a real text. Multiple-Labelled Context Related Node Description Model (MCRNDM) is introduced in this paper for the formal descriptions of the properly detailed features and constraints. It is indicated as the following diagram:

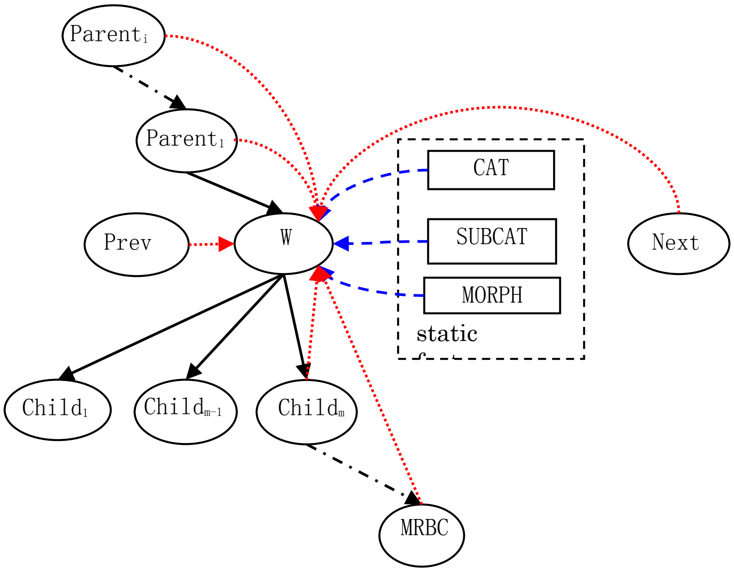


FIGURE 1. MCRNDM MODEL DIAGRAM

In this diagram, parent means father node, parent$_i$ means ancestor nodes, prev and next mean the previous node and the next node in a linear structure respectively, Child$_1$、Child$_{m-1}$、Child$_m$ mean child node respectively, among them, Child$_m$ is the rightmost child node (the child node its node-indexed value is the biggest), MRBC is for the Most Right Bottom Child, CAT、SUBCAT、MORPH mean the W's features of the POS, subcategorizations and the morphological features respectively. The dotted lines pointing at the nodes of ⋯⋯⋯ indicate the static features, while the dotted lines of ------- indicate the dynamic constraint conditions. The features and the constraint conditions constitute the complex feature set.

Static information can be the information of the POS, subcategorization and morphological features of the scanned nodes. The information of the POS and subcategorization is stored in the dictionary from which that information can be obtained by the query. The POS ambiguity are identified using the rule-based method and the context-sensitive rules for recognizing more than two thousand words with POS ambiguity were concluded in this paper. The morphological features can be obtained using the method of FSA-based machine dictionary and the related algorithms. The constraint conditions are from the features of syntactic structures of the related nodes in the results of the partial parsing which including father nodes which can be the multilevel nodes, descendent nodes, brother nodes, and the types of dependency relations between the adjoining nodes in the leaner structures, the numbers of the relations, the leaner distance (the dependence distance), the location of syntactic fragment containing the current node. These lines of information can be obtained using a group of function.

The constraint conditions of nodes are added on the rules according to the needs of constraints. Not each rule has the context-sensitive constraint conditions.

3. **Recognition Rules for the Dependency Relations in Mongolian language.**

3.1. **The segmentation of Sentences.** First of all, the problems of the segmentation of sentences should be resolved in the syntactic parsing of real text. It is possible to parse the sentences only after segmenting the sentences in the text one by one. The punctuation plays an important role in the segmentation of sentences, so we take the full stops, the question marks and the exclamation marks as the boundary markers in this paper. According to the work of [4], the algorithms for the segmentation of sentences are described as following:

(1) To segment the sentences by the full stops, the question marks and the exclamation marks (except those in the quotation marks or the brackets)

(2) To segment the inserted sentences including the boundary markers then. There are two kinds of inserted sentences. One has the boundary marker and the other has not. An inserted sentence is bracketed by the marks of 《》or <>, which is inserted as a whole in the dependency structures in the whole sentence. In this paper, the content of the inserted sentences is also segmented and analyzed with dependency in order to avoid appearing the node bigger than word in the dependency trees. The method of segmenting the inserted sentences is the same as the step of (1).

(3) To combine the inserted sentences with no markers. The inserted sentences with no markers are the sentences without the boundary markers like the marks of 《》 or <>. The method of segmentation in the step of (1) covers above may result in errors if there are the full stops, the question marks and the exclamation marks in the inserted sentences with no markers. It segments a whole sentence improperly according to the marks of the inserted sentences. The wrong segmentations influence the analyses of the next level if they are corrected immediately. The method of avoiding this kind of errors is to check the results of segmentation in the step of (1) firstly, and combine the current sentence and its previous sentence if there are only the linking verbs of GE or HEME and there are not any content linked by them.

For example, the two following sentences are segmented incorrectly. As a result, those are segmented as two sentences on the point of 《》or <> using the method of the step of (1) according to the marks of the question marks and the exclamation marks.

BI NISHEL DEGER_E-ECE ONGGEYIN ONGGEYIN HARAJV EYIMU SAYIHAN DALAI-YI C0H0M YAMAR T0HITAI UGE-BER JUIRLEHU BVI ? //GEJU D0T0R_A-BAN UGE ERIJU LE YABVL_A .（*I looked out of the window of the aircraft from time to time thinking about what words are used to describe such a vast and beautiful sea.*）

TEGEHU-DU EJEN=HOMON-U HOYI=HAYI ! //GEJU NARIN ONDOR HASHIRHV DAGVN-IYAR H0NID MVHVR HAMAR-IYAN DEGEGSI ERGUN VRVGSI HARAGAD HVDDVG VRVGV DABHILDVBA .（*He is shouting with the host's posture, which makes the sheep raise their wide noses and run forward to the side of the well.*）

If there are more than one inserted sentence with no markers, the method described above can also not recover the whole sentence, either. We adopt the method of manual correcting for resolving this kind of problem.

3.2. **The Recognition of Syntactic Fragments.** The divide-and-conquer strategy is an effective method in controlling the analyses complexity increasing with the gradual increase of sentence length. The divide-and-conquer strategy was used in the work of [1-2], and the concept of chunks or syntactic fragments was introduced in the syntactic parsing of English and Chinese languages. In the work of [3], Some scholars also divided one sentence into three parts: pre-subject, subject, predicate. No matter what approach they are, their target is to reduce the length of the unit of syntactic analyses and enhance the analysis efficiency. As a result, these methods can simplify the analyses of the complex sentences, but most of them depend on the language itself and are not easy to describe other languages.

The length of the sentence also has great influences on the accuracy of syntactic parsing. It can be illustrated as the following figure.

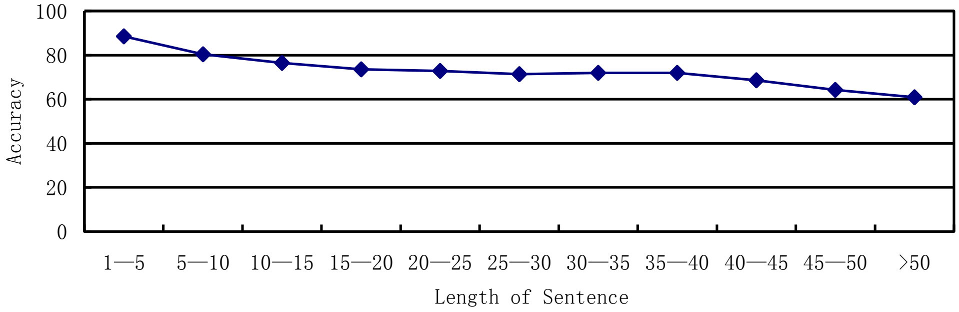


FIGURE 2. RELATIONS BETWEEN THE LENGTH OF SENTENCE AND THE ACCURACY OF SYNTACTIC PARSING

Hence, it is necessary to segment the sentences and reduce the length of the unit of syntactic analyses. In this paper, we define the syntactic fragment as follows: a few of consecutive words with only one head word. The syntactic fragment can be a word, a phrase, a clause or sub-clause. Two fragments with the relations between themselves constitute one bigger fragment.

One of the two boundaries of a syntactic fragment can be decided according to the punctuation, POS, and some lexical and structural information. The commas, verbs, conjunction words including link verbs and modal particles can be the symbols in segmenting the syntactic fragments in Mongolian language. We summarize the following five rules after doing the statistical analyses in the training set. The algorithm of the segmentation scans the sentences many times and each time splits one sentence or one fragment into two fragments.

Para-Rule1: to divide the sentence into two parts at the location after the comma if there is a comma in a sentence.

Para-Rule2: to divide the sentence into two parts at the location before the conjunction word including the link verb if there is a conjunction word including the link verb.

Para-Rule3: when the sentence-scanning forward finds the structure of “verb + static word”, it should continue to check if the static is an auxiliary constituent. If it is, to divide the sentence into two parts at the location after the auxiliary constituent including the consecutive a number of auxiliary constituents; if it is not, to divide the sentence into two parts at the location after the verb.

Para-Rule4: when the sentence-scanning forward finds the structure of “verb + verb”, it should not be spitted if the second verb is the auxiliary verb or the first verb is a simple adverbial verb; Otherwise, to divide the sentence into two parts between them.

Para-Rule5: to divide the sentence into two parts at the location before the notional word if there is a notional word after the mood words such as the interrogative particle, the positive particle, the post negative particle, the particle indicating the meaning of memory, the particle indicating inference, the particle indicating transmission, the particle indicating impatience, blame, exclamation.

The numbers in the rules show the sequences of preference. First, the rules with the smaller numbers are used in segmenting the syntactic fragments, and then the larger numbers are used in the results of partial parsing. The results of segmentations with the rules may be the clauses, the constituent sentences, the phrases or the words. This kind of method is not introduced for describing the hierarchical structure of sentences. It is the method just for decreasing the difficulty in the process of analyses. As a middle process, it can be adopted as long as it can increase the efficiency of analyses.

The recognition rules of the syntactic fragments are reasonable for processing the compound sentences. It may result in errors of including the parts of a main sentence in processing the middle constituent sentences. For this problem, we adopt the method of post processing. That is, we check the constituents such as the subject, the preset adverbial modifier in the syntactic fragments after forming the complete tree. They can be cut down if the structural features of these constituents fit for some rules in the algorithms for post adjustment.

3.3. **The Rules for Recognizing the Dependency Relations.**

3.3.1. **The Rules for Recognizing the Dependency Relations in the Fragments.** The rule set for recognizing Mongolian dependency relations is made up of the following 7 parts: the rules of recognizing subject-predicate relations, the rules of recognizing attribute-subject relations, the rules of recognizing relations of the direct object and the predicate, the rules of recognizing relations of the indirect object and the predicate, the rules of recognizing relations of the adverbial modifier and the predicate, the rules of recognizing auxiliary relations and the rules of recognizing parallel relations. The rule set is too large to introduce in detail here. Taking the rules of recognizing subject-predicate relations as an example, the formalization of the rules is described as follows:

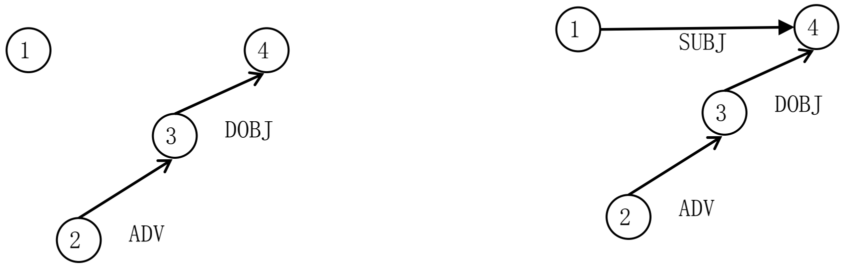
The form of rule : $W_iW_j \Rightarrow W_i \xrightarrow{SUBJ} W_j$, in which, W_i、W_j : words that take part in the dependency parsing, in which,$1 \leq i$, $j \leq n\_S$（show the word number in the sentence） ; $\Rightarrow$: show the generation ; $\longrightarrow$: show the dependency direction ; SUBJ shows the dependency types ;

Conditions of the constraints :

subj-R01：<W_i CAT>=<N>　　　　　RelCount（W_j,SUBJ）= 0
<W_i SUBCAT>=<xN||Nx>　　　　　Parent（W_j）= NULL;
<W_i MORPH>=<Fc0>
<W_j CAT>=<V>
<W_j SUBCAT>=<Ve>

In the above description, subj-R01 is the code for the rule sequence, in which, the number shows the preference in the same category of the rules. “01”in subj-R01 shows that the rate of preference for the subj-R01 is the largest in the recognition rules of subject-predicate relations. The left part in the condition of rule constraint is for the static features, in which CAT,SUBCAT,MORPH show separately the POS, the subcategorization, the morphological features, N、V is for the noun and the verb, xN shows the noun for human, Nx is for the human name, Ve is for the general verb, Fc0 is for the nominative case; the right part in the condition of rule constraint is for the dynamic syntactic features, in which RelCount（W_j,SUBJ）= 0 shows that there is not subject for the control word, Parent（W_j）= NULL shows that there is not the father node for the control word. We made 31 functions in order to extract the dynamic information from the partial syntactic parsing. The introduction for these functions is as follows:

Take an example for the rule invocation. The following is the result at some moment in the analyses of the Mongolian sentence “[]CIMED NEBTERETEL_E N0R0GSAN-IYAN MARTAJAI .（CIMED forgot himself drenched through.）”, as expressed in figure 3. The order number in the figure shows the word location in the sentence. From that figure, we can see that there is the relation of the direct object and the predicate between “N0R0GSAN-IYAN”（drench）and“MARTAJAI .”（forgot）,and the relation of the adverbial and the predicate between “NEBTERETEL_E”（permeate）and “N0R0GSAN-IYAN”. The next parsing will be done between “[]CIMED” and three pairs of words “MARTAJAI .”, “[]CIMED”and“N0R0GSAN-IYAN”, “[]CIMED”and“NEBTERETEL_E”. In light of analyzing the static and dynamic features of three pairs of words, “[]CIMED” and “MARTAJAI .” meet the constraint conditions of the subj-R01. In that, the static features of“[]CIMED” are as follows: noun, human name, nominative; the static features of “MARTAJAI .”are as follows: general verb. The dynamic features are that there is not subject for the verb “MARTAJAI .”and the father node.



(a) before using the rule of subj-R01　(b) after using the rule of subj-R01

FIGURE 3. EXAMPLES WITH THE RULE OF SUBJ-R01

The process of matching features is same in the recognition rules of all the dependency relations.

3.3.2. **The Rules for Recognizing the Dependency Relations between the Fragments.** A number of sub-trees have been made for each sentence through segmenting the syntactic fragments and recognizing the dependency relations in the fragment. The current task is to analyze the relations between the sub-trees and build a complete dependency tree。 There may exist the relations of subject-predicate, attribute-subject, object-predicate, adverbial-predicate or parallel relations. Generally, the syntactic fragments are related to their head words and result in the dependency relations. Only when the head word is an adjective, is it possible that this fragment may modify the first word (its linear distance from the previous fragment is the nearest) in the next fragment or the ancestor nodes of the first word. Hence, in this case, we should use a special processing.

The recognition rules of dependency relations such as subject-predicate, attribute-subject, object-predicate and adverbial-predicate can also be apply to the processing of dependency relations between the fragments. The following focuses on the recognition of parallel relations. From the statistics in MDB, we can see that the dependency distance of the parallel relations is 6.06 words on the average. This value is the first in the value-ranking of all the dependency relations. The recognition rate for the parallel relations is the smallest one from the perspective of automatic parsing. For the rule-based parser, it reaches 40.09%. From the above statistics, at the current, using rules to process the parallel relations is currently a goog choice. We develop the algorithm for recognizing the parallel relations according to the features of the related nodes in two sub-trees including the POS feature, subcategorization feature, the punctuations and the morphological features. The description of the algorithm is as follows:

(1) if the head word in the left sub-tree is with the coordinating conjunction, the parallel relation is built directly between the head words of the two sub-trees;

(2) if it cannot meet the condition of (1), the similarities between the two sub-trees compared. The value of the similarity is decided by computing the similarities between the head words and the rightmost child nods in the two sub-trees. If the similarity is bigger than the value of the preset threshold, the parallel relation can be built between the head words of the two sub-trees.

(3) if it cannot meet the condition of (2), the rules of recognizing the relations such as subject-predicate, attribute-subject, adverbial-predicate, object-predicate and auxiliary relations are employed.

4. **Algorithms for Searching.** Along with inducing and refining the search algorithm development is also an important step in the rule-based syntactic parsing. In the work of [5], a search algorithm with local optimization based on the divide-and-conquer strategy was introduced. A Mongolian sentence with n words changed as follows after segmenting the words and the fragments:

W_1 W_2 … W_i || W_{i+1} W_{i+2} … W_k || … || W_m W_{m+1} … W_n， in that， W_i is for word,|| is for the fragments segmentations.

In the syntactic parsing, the internal dependency relations of each fragment are analyzed firstly, and then, the sub-trees of each fragment are combined. As for the parsing method, there is not any difference between parsing in a fragment and parsing between fragments. The fragment segmentations play an important role in decreasing the length of analysis, limiting the error propagation and increasing the accuracy of parsing.

The parsing starts from two rightmost nodes. A sentence changed as following after parsing many times:

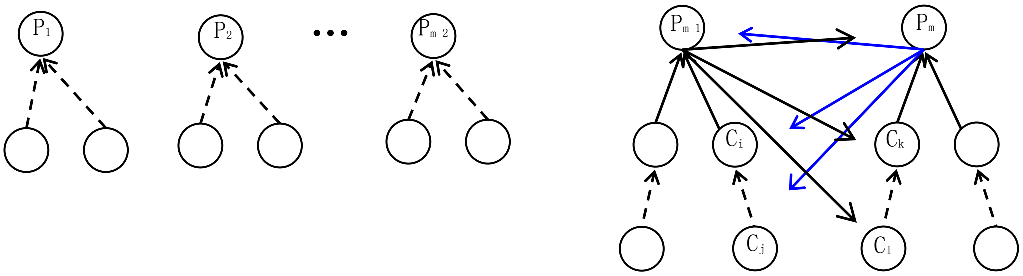


FIGURE 4. PARSING ALGORITHM DIAGRAMS

The dotted line shows that there are n nodes between the nodes at the edges, P_1，P_2，…，P_{m-2}，P_{m-1}，P_m show separately the root node in each sub-tree, C_i，…，C_j，C_k，…，C_l show the descendent nodes related to the current analysis in the two sub-trees. These nodes are located on the edges that are merged, that is, in the sub-tree with the root node of P_{m-1}. These descendent nodes are the rightmost nodes in their father node. In the sub-tree with the root node of P_m, these descendent nodes are the leftmost nodes in their father node.

The next analyses go on between P_{m-1},C_i,… ,C_j and P_m,C_k,… ,C_l, similar to the description by the arrows in the figure. The groups of nodes with possible dependency relations are as follows : $P_{m-1}\rightarrow C_l$; $P_{m-1}\rightarrow C_k$; $P_{m-1}\rightarrow P_m$; $P_m\rightarrow C_i$; $P_m\rightarrow C_j$; $P_m\rightarrow P_{m-1}$; at last, which two of them can have the dependency relations? It depends on the binding abilities between the two nodes.

If there is only one group of nodes finding the corresponding rules, the dependency relations are built and this parsing ends.

If there is more than one group of nodes finding the corresponding rules, they are sorted according to the preferences, and the dependency relations are built in the group of rules with the highest score and this parsing ends.

After the above analyses, P_{m-1} and P_m are combined into one tree. The tree after the combination and Pm-2 are combined once again. Analogously, the parsing of all the sub-trees is completed.

The description of the algorithm is as follows:

```
BOOL RuleBasedParsing (NodeStack &S)
{
     CwordNode N,P,T;// N refers to the right subtree, P refers to the left subtree,
                    //T refers to a temporary variable
     Carray NodeArray; // A array to store the node set, they set up a dependency
                      //relations temporarily.
          While NodeStack is not empty
                    POP(S,N);
          POP(S,P);
                    If((N!=NULL)&&(P!=NULL)){
          T=GetMostLeftConner(N);//Get the most left-bottom node of the right subtree
                    While(T!=NULL){
                              If(IsHaveMatchedRules(T,P)) NodeArray.Add(T+P);
                              T=T->Parent;}
          T=GetMostRightConner(P);//Get the most right-bottom node of the left
                                  //subtree
                    While(T!=NULL){
                              If(IsHaveMatchedRules(T,N)) NodeArray.Add(T+N);
                              T=T->Parent;}
                    If(!NodeArray.IsEmpty()) {
               Get Max-Score node set;establish relation;} //Select the node
          //combination, they are most likely to establish dependency relation.

}
```

ALGORITHM 1. RULE-BASED DEPENDENCY PARSING ALGORITHM

5. **Experiment and Analyses.** We tested the parser in the set of test with the first 1332 sentences (the textbook of the Mongolian language in middle school) in MDTB and the last 3653 sentences (the first textbook of the Mongolian language in high school, 6 volumes). The tested aspects include the sentence segmentation, the syntactic fragment segmentation and the dependency relation tagging. The results show that the accuracy of segmenting sentences is reaches 98.6%. The error took place in the inserted sentences without any mark. The segmentation of syntactic fragments increases the whole ability of the parser by about 2.56%. In tagging the dependency relations, the accuracy without any tag, the accuracy with tags and the accuracy of finding head words exactly reaches respectively 75.21%、69.39% and 75.21%.

6. **Conclusions.** Morphological features are the static information with the ability of disambiguation in the rule-based Mongolian dependency parsing. Making full use of the case and the morphological features of verbs, this paper completed the automatic parsing with the method of sub categorization. From the perspective of parsing process, syntactic parsing is the process of combining the sub-trees. Each sub-tree owns only one node at first. With the combination of many sub trees, one complete dependency tree can be built at last. In the process of combining the sub-trees, the root node of the new tree owns more and more structural information through the unifications. We use the dynamic structural information including the dependency distance, the number and the property of slave nodes, and the related features of the ancestor nodes and the rightmost child nodes in the

recognition rules and algorithm.

Overall, the accuracy of this dependency parser reaches the target and meets the expected results. However, there also exist so many problems in recognizing some special relations. For example, the recognition rates of the parallel relations are very low, because there are the parallel relations on many levels including single words, phrases, sentences constituents, and clauses, and the constituents in the parallel relations have not any distinguishing features in the morphological aspect and the POS aspect.

Acknowledgment. This work is partially supported by National Natural Science Foundation of China (the project no. is 60763003 and 61262046) and National Social Science Foundation of China (the project no. is 10CYY022).

REFERENCES

[1] ZHOU Qiang, ZHAN Weidong and REN Haibo, Build a large scale Chinese Functional Chunk Bank, *Proceedings of JSCL2001*, Taiyuan, pp.102-107, 2001.

[2] MA Jinshan, Research on Chinese Dependency Parsing Based on Statistical Methods, *Doctoral Dissertation of Harbin Institute of Technology,* Harbin, pp.52-69, 2007.

[3] Lyon C. and B. Dickerson, Reducing the Complexity of Parsing by a Method of Decomposition, *Proceedings of Fifth International Workshop on Parsing Technology*, Boston, pp.215-222, 1997.

[4] Qinggeltei, *Mongolian Grammar*, Inner Mongolia people's Publishing House, Hohhot, pp.215-222, 1997.

[5] S. Loglo, HUA Shabao and Sarula, Mongolian Dependency Parsing Based on Statistical Methods, *Journal of Chinese Information Processing,* vol.26, no.3, pp.27-32, 2012.