A Lexical Dependency Probability Model for Mongolian Based on Integration of Morphological and Syntactic Features

S.Loglo*
College of Mongolian Studies, Inner Mongolia University, Hohhot, China

*Corresponding author email: sloglo@imu.edu.cn

Abstract. Syntactic parsing is an important topic in the field of Mongolian language information processing. Compared with English and Chinese dependency parsing, Mongolian dependency parsing is still at the beginning stage. Mongolian syntactic parsing lack of Treebank resources seriously, under such conditions, a high quality syntactic parser cannot be developed by statistical methods simply. Aiming at the characteristics that Mongolian language has rich morphological features, this paper presented a rule and statistics-based dependency parsing model using Mongolian Dependency Treebank as training and evaluation data. The morphological and syntactic rules are represented using complex features and unification operations. The statistical model is represented using lexical dependent probability. This model has now achieved accuracies of 77.18%, 69.42% and 95.44% for the unlabelled annotation score, the labeled annotation score and the head word annotation score respectively.

1. Introduction
Mongolian information processing has generally been done on word and phrase level and is now en route to sentence processing (Nasunurtu, 1998). Having completed phrase boundary bracketing, phrase structure relationship recognition (Dabhurbayar, 2008), predicate segment automatic identification and other shallow parsing (Wang, 2009), it is now entering the stage of full parsing. The main tasks at this stage include developing a Mongolian syntactic annotation scheme, a sizable Treebank and an effective syntactic parser. Full parsing helps improve the properties of upper-layer application system, including Mongolian machine translation, syntactic checking, information extraction and automatic question answering. After observing the situation of dependency parsing in the past five years (Chen et al., 2015) it is found that the model based on neural network and deep learning has occupied a dominant position in resource rich languages. However, Mongolian belongs to a low resource language, and the method based on statistics and rules has certain advantages in dependency analysis. About related works for Mongolian dependency parsing, Inner Mongolia University obtained support from the National Natural Science Foundation of China from 2008 to 2011, and has constructed the dependency Treebank of middle school Mongolian language textbooks(about 500000words) (S.Loglo, 2016). On this basis, a dependency Treebank of 200000 word news corpus was constructed in 2016 (Qiligee, 2016) and a 140000 word literature corpus dependency Treebank was constructed in 2019 (Chaomurilighe, 2019). At present, the scale of Mongolian Dependency Treebank (MDTB) has reached 840000 words. In terms of dependency parsing, Nanding conducted an in-depth study on Mongolian object-predicate relation based on dependency grammar in his master's dissertation (Nanding, 2016). Qinggeletu Bai carried out a research on automatic recognition of Mongolian indirect object-predicate relation based on MDTB (Qinggeletu Bai, 2018).
2. Dependency Parsing Based on Mongolian Morphological and Syntactic Features

2.1. Description of Mongolian Morphological and Syntactic Features

In reference to rule-based parsing algorithms of English, Germany and Chinese and in view of Mongolian’s rich morphological features, this paper proposes a rule description system for Mongolian dependency parsing based on complex features and unification operations. Each rule of the rule set is a generating rule for dependency arc. Rule conditions comprise characteristic value and constraint conditions of the scanned nodes. Rule actions include dependency relation generation and feature set unification. Dominance and dominated relations are determined by restricting values of the scanned nodes’ features and the scanned range of the nodes. A whole dependency tree for a sentence with n words contains n nodes and n-1 dependency arcs. As such automatic parsing needs n-1 production rules (that can be repeatedly used) to generate n-1 dependency arcs.

Dependency rule can be expressed as: \( \text{POS}_i \text{POS}_j \xrightarrow{\text{rel}} \text{POS}_j \), where \( \text{POS}_i \) and \( \text{POS}_j \) are the part of speech of two words between which a dependency relation is to be built, \( \xrightarrow{\text{rel}} \) denotes generation, \( \to \) represents dependent on \( \text{POS}_j \), and \( \text{rel} \) is dependency relation type.

This production rule adopts single feature representation, viz. part of speech (POS) information. In the actual parsing, POS information is too coarse-grained to constrain generative action effectively. To illustrate, the sequence NV (N denotes Noun and V Verb) can be generated into three different dependency relations at least. For example, BATV (“a person’s name” whose POS is “N”) and IREBE (“come” whose POS is “V”) constitute subject relation; BVDAG_A (“meal” whose POS is “N”) and IDEBE(meaning “have” whose POS is “V”) form direct object relation; ORLOGE (“morning” whose POS is “N”) and YABVBA (“go” whose POS is “V”) constitute adverbial relation. Such ambiguous structure can be best prevented from occurring by the following rule:

\( \text{W}_i \text{W}_j \xrightarrow{\text{rel}} \text{W}_j \),

where \( \text{W}_i \) and \( \text{W}_j \) are the two words participating in a dependency relation respectively. To say that this rule is best at removing ambiguity is because the language nearly has no ambiguity on word level, except for multi-category words and polysemous words. But this rule system can only be realized in empirical text with a small vocabulary, rather than in real text. This is because any two words may generate a dependency relation in real language, and if a generation rule is compiled for each word pair, the number of rules thus compiled may become infinite. Hence it is necessary and feasible to build a rule system that lies between the above two rules and is attached with moderately detailed characteristics and constraint conditions for a pragmatic and real text-based Mongolian automatic parsing. To graphically represent the detailed characteristics and constraint conditions, this paper introduces a Multiple-Labeled Context Related Node Description Model (MCRNDM) as shown in Figure 1.

![Figure 1. Schematic Diagram of MCRNDM Model](image-url)
MRBC (Most Right Bottom Child) is the descendant node at the rightmost bottom. CAT, SUBCAT and MORPH are W’s POS, subcategorization and lexical features respectively. Directional node W’s dotted line—denotes static features, and its dash line—represents dynamic constraint condition. Features and constraint conditions constitute a complex feature set.

Static information can be information of POS, subcategorization (including semantic classification) and morphological changes of the scanned nodes. POS and subcategorization information have been stored in a dictionary from which such information can be obtained by query. This paper puts together context dependent recognition rules for over 2,000 multi-category words based on rule-based processing. Morphological features can be obtained by FSM (finite state machine)-based dictionary and pertinent recognition algorithm. Constraint conditions are syntactic structure features of relevant nodes whose partial parsing has been finished. These conditions include dynamic information such as dependency relation type of parent node (multilevel node included), descendant node, sibling node and the types of dependency relations between adjoining nodes in the linear structure, the number of such relations, linear distance (dependency distance) and the location of the current node’s syntactic fragment. Such information can be obtained using a group of functions.

The constraint conditions of nodes can be added to the rules when necessary. But not every rule has context dependent rule constraints.

2.2. Recognition Rules for Dependency Relations in Mongolian Based on Morphological and Syntactic Features

To realize Mongolian dependency parsing, we have designed and implemented three different parsers based on rules approach, statistical approach and hybrid approach consecutively. Function wise, the hybrid approach-based parser performs the best. In this parser, the rule-based part contributes the most as Mongolian has rich morphological information that is useful for determining some dependency relations. The hybrid approach also solves the sparse data problem in the statistical model and reduces the time complexity of parsing.

The rules in this paper include sentence segmentation, recognition of syntactic fragments and recognition of dependency relations. Sentence segmentation is the first step in real text parsing. It is only after text is segmented into individual sentences for further structure analysis. Punctuation marks play an important role in sentence segmentation. In this paper, sentence is segmented using full stop, question mark and exclamation mark as boundary markers. The algorithm for sentence segmentation is described as follows:

1. Sentences are first segmented using full stops, question marks and exclamation marks (excluding those in the quotation marks and brackets) as boundary markers.

2. Parenthesis sentences with boundary markers are segmented. There are two types of parenthesis sentences, those with boundary markers such as “《》” or “<>” and those without. In this paper, parenthesis sentence as a whole participates in the dependency structure of a sentence. But in case one node becomes bigger than a word in dependency tree, parenthesis sentence is also subject to sentence segmentation and dependency parsing. The segmentation method is the same as that in Step (1).

3. Parenthesis sentences without such boundary markers as “《》” or “<>” are combined. When such a parenthesis sentence has full stop, question mark or exclamation mark, wrong segmentation may ensue if a whole parenthesis is segmented as is done in Step (1). If not corrected, this error may go to the next stage of parsing. The method to avoid this error is to check the results of the sentences segmented in Step (1). If one sentence has such link verbs as “GE” or “HEME” without content linked by them, this sentence is combined with the previous sentence.

When a sentence is segmented into fragments, shortening the parsing units is necessary. In this paper, we define syntactic fragment as a string of consecutive words with only one head word. It can be a word, a phrase, a clause or sub-clause. Two fragments can constitute a bigger fragment through the dependency relation between them.

Left and/or right boundary of a syntactic fragment can be determined by punctuation, POS or specific lexical and structural information. We use coma, verb, conjunction words (including link verbs) and modal particles as main tokens to segment Mongolian syntactic fragments. After statistically parsing
the training corpus, we put together five segmentation rules. The segmentation algorithm scans a sentence multiple times, and each time splits a sentence or a fragment into two fragments.

Para-Rule1: If a sentence contains a comma, it is divided into two fragments at the location after the comma;

Para-Rule2: If a sentence contains a conjunctive word (including link verb), it is divided into two fragments at the location before the conjunctive word;

Para-Rule3: When forward scan finds a sentence with “verb+static word”, check whether the static word is an auxiliary constituent. If yes, this sentence is divided into two fragments after the location at the “auxiliary constituent” (including several consecutive auxiliary constituents). If no, this sentence is divided into two fragments after the location at the “verb”.

Para-Rule4: When forward scan finds a sentence with “verb+verb”, this sentence cannot be split if the second verb is an auxiliary verb or if the first verb is a pure conjunctive verb. Otherwise, this sentence is divided into two fragments between the two verbs.

Para-Rule5: If a sentence has notional words behind interrogative modal particles, positive modal particles, postposition negative modal particles, and modal particles used to express memory, narration, impatience, blame and exclamation and shouting, this sentence is divided into two fragments at the location before the notional word.

The labeling numbers in the code of rules denote sequence of preference. Smaller numbers are used to segment sentence into fragments first before partial fragments are segmented using larger numbers. The result of rule-based segmentation may turn out to be a clause, a constituent sentence, a phrase or a word. This segmentation method is not designed to recognize hierarchical structures of sentences, but rather to reduce difficulty in parsing. As an intermediate process, however, it can be adopted so long as it improves parsing accuracy.

We have put together nearly two hundred recognition rules for dependency relations based on static features like part-of-speech, subcategorization and morphological changes of word nodes and dynamic syntactic features obtained from partial analysis. The rule set is too big to be presented in detail here. For the purpose of illustration, we present one example below:

\[ w_i, w_j \Rightarrow w_i \xrightarrow{(\\text{subj-R01})} w_j \]

subj-R01: \(<w_i, \text{CAT}>=<N>\) \quad \text{RelCount (W_j, SUBJ)} = 0

\(<w_i, \text{SUBCAT}>=<xN||Nx>\) \quad \text{Parent (W_j)} = \text{NULL};

\(<w_i, \text{MORPH}>=<Fc0>\)

\(<w_j, \text{CAT}>=<V>\)

\(<w_j, \text{SUBCAT}>=<Ve>\)

where \(w_i\) and \(w_j\) denote dependent word and head word respectively. CAT, SUBCAT and MORPH on the left of constraint conditions are static information as part-of-speech, subcategorization and morphological changes of word nodes. The rightward constraint conditions are dynamic syntactic features. To obtain partial dynamic information, we have developed 31 functions which will not be elaborated here. The labeling numbers indicate the sequence of preference in the same category of rules. For example, “01” in <subj-R01> shows that subj-R01 has the highest preference in recognition rules for subjects.

3. Mongolian Dependency Probability Model Based on Integration of Morphological and Syntactic Features

The philosophy of parsing is to produce parse trees \(t\) for a sentence \(s\) based on certain grammar \(G\) (Eisner, 1996) In most cases where there are more than one parse trees for one sentence, we use \(T\) to represent all possible parse trees. Modeling in statistical parsing is to find an evaluation function, sort the analytical results using probability values and output the most probable result. The probabilities of parse trees \(t\) are such that:

\[ P(t | s, G), \]  
\[ \sum_{t \in T} P(t | s, G) = 1 \]  

(1)
Statistical parsing model turns disambiguation into an optimization process. In other words, the probability of each parsing result is calculated to find a highest-probability parse tree $t^*$. This process can be expressed as:

$$ t^* = \arg \max_{t \in T} P(t \mid s) = \arg \max_{t \in T} P(t \mid s, G) \quad (2) $$

With the rapid development of machine learning methods and growing enrichment of data resources, statistical dependency parsing is also evolving. The most common modeling methods are generative dependency model and discriminative dependency model. Each model has different processing techniques. Given the sophistication of integration and the size of the available corpus, this paper adopts a lexical dependency probability model, one type of generative dependency probability model. Parsing is a bona-fide grammatical disambiguation process supported by multiple types of information. Word is most capable of separating because ambiguity seldom arises on word level (Ma Jinshan, 2007). But constrained by the sizes of corpora, lexicalized statistical parsing model has only begun to be applied in recent years.

A sentence that is formed by a finite number of words can be expressed as:

$$ S = \{<w_1, f_1, t_1>, \ldots, <w_i, f_i, t_i>, \ldots, <w_n, f_n, t_n>\}, $$

where $w_i$ is the $i^{th}$ word of the sentence, $f_i$ is the $w_i$’s morphological feature, and $t_n$ is the $w_i$’s annotation information such as part-of-speech and subcategorization. Function (2) can be restated as:

$$ t^* = \arg \max_{t \in T} P(t \mid <w_1, f_1, t_1>, \ldots, <w_i, f_i, t_i>, \ldots, <w_n, f_n, t_n>, G) \quad (3) $$

To calculate $P(t \mid <w_1, f_1, t_1>, \ldots, <w_i, f_i, t_i>, \ldots, <w_n, f_n, t_n>)$, we make an independent hypothesis for dependency arc. In other words, a dependency arc is only related to EOS (end of sentence) words, independent of other nodes and arcs. The dependency tree for a sentence with $n$ words comprises $n-1$ dependency arcs. As per the above independent hypothesis, the probabilities of the dependency tree are a product of the $n-1$ dependency arcs’ probabilities such that:

$$ P(t \mid s) = \prod P(A_{ij} \mid w_i, w_j) \quad (4) $$

where $A_{ij}$ denotes the dependency arc between the two words $w_i$ and $w_j$ $(1 \leq i, j \leq n; i < j)$. Its direction is determined by the locations of the dependent word and the head word. If $w_i$ depended on $w_j$, “1” would be used to denote its direction. If $w_j$ depended on $w_i$, “0” would be used to denote its direction.

3.1. Searching Algorithm

**Step1:** A sentence is split into $m$ fragments ($m=1$, $m\leq n$ (number of words)) using sentence segmentation rule.

$$ W_1 \ W_2 \ldots \ W_i \ || \ W_{i+1} \ W_{i+2} \ldots \ W_k \ || \ldots \ || \ W_m \ W_{m+1} \ldots \ W_n, $$

where $W_i$ denotes word and $\|$ represents fragment segmentation.

**Step2:** Fragment-internal auxiliary dependency relations are individually recognized using recognition rules for auxiliary relations. This is because dependency distance between auxiliary relations is relatively short, and their structural rule is also relatively simple. In Mongolian, only auxiliary relations are head-initial dependency relations.

**Step3:** Head-final dependency relations are recognized. When a head-final dependency relation is built between two subtrees, such relation only arises out of the root node of leftward subtree and the leftward node of rightward subtree (including root nodes and leftward descendant nodes).

If a syntactic fragment has $m$ subtrees, combination starts from the two rightmost trees and goes on until one subtree is left.

Specifically, matching rules are searched for each group of nodes that could be combined; If the matching rules are only found for one group of nodes, a dependency relation will be generated and this round of parsing is concluded.;
If matching rules are found for more than one group of nodes, the system will sort the group of nodes according to the sequence of preference and generate a dependency relation is built for the highest-scoring group. This round of parsing is concluded;

If no matching rules are found, the statistical model will be initiated to calculate the probability for each group of nodes that could be combined, sort probability values and then generate a dependency relation for the highest-scoring group. This round of parsing is over.

The above parsing will allow the two rightmost subtrees to be combined into one subtree. The combined subtree is then combined with the third rightmost tree. It goes on until all subtrees are parsed.

**Step 4:** Syntactic fragments are combined. In terms of parsing method, there is no difference between intra-fragment parsing and inter-fragment parsing. But fragment segmentation shortens parsing units and restricts the spread of errors, thus enhancing the accuracy of parsing.

The algorithm is described as follows:

**Algorithm 1. A Rule and Statistics-Based Dependency Parsing Algorithm**

3.2 Algorithm Example

Below is a concrete example to illustrate the decoding process of this algorithm. Let’s assume that the input sentence is:

"To facilitate computer processing, this sentence is converted using Latin Transliteration into the following: TNGRI-YIN AGAR DVLAGABTVR NEMEYIJU 0GTARGVI-YIN ONGGE HOGENGGITUN VNIYARTVGAD, HOMOS-TU HABVR-VN IREHU-YI SANAGDAGVLVJ BAYIL_A. (The warm and cloudy weather remind people that Spring is coming. This sentence is extracted from **Spring Sun Rises in Beijing, a book by Nasaiyinchaoketu**)."

The first step is to split this sentence into three fragments: “TNGRI-YIN AGAR DVLAGABTVR NEMEYIJU”, “0GTARGVI-YIN ONGGE HOGENGGITUN VNIYARTVGAD” and “HOMOS-TU HABVR-VN IREHU-YI SANAGDAGVLVJ BAYIL_A.”. The second step is to annotate head-initial dependency relations fragment by fragment. Result of front-end parsing shows that the third fragment has an auxiliary verb “BAYIL_A.”. A head-initial dependency relation is generated for this verb. The third step is to build subtrees corresponding to the three fragments using algorithm rules, as shown in Figure 2.
The fourth step is to combine each subtree. Figure 3 shows the combined subtree.

![Image of combined subtree](image_url)

**Figure 3.** Subtree after fragment combination

4. Experiment and Analyses

We test the parser using Mongolian Dependency Treebank (MDTB) established by College of Mongolian Studies, Inner Mongolia University in 2011. This treebank contains 461,240 words and 31,722 sentences (mean length=14.54). The training corpus (Middle School Mongolian Textbook 2 and Textbook 11) contains 26,737 sentences and 402,432 words (mean length=15.04). The testing corpus (Middle School Mongolian Textbook 1 and Textbook 12) contains 4,985 sentences and 58,808 words (mean length=11.80). Unlabeled Annotation Score (UAS), Labeled Annotation Score (LAS) and Head-Word Annotation Score (HAS) are used as indicators to evaluate three parsers. The evaluation results are given in Table 1. MPParser-1 denotes rule-based dependency parser, MPParser-2 statistics-based dependency parser, and MPParser-3 hybrid dependency parser integrating rules and statistics.

|               | UAS   | LAS   |
|---------------|-------|-------|
| MPParser-1    | 75.21%| 69.39%|
| MPParser-2    | 71.17%| 61.37%|
| MPParser-3    | 77.18%| 69.90%|

**Table 1.** Three Parsers Compared

Mongolian dependency relations are divided into head-initial dependency relations and head-final dependency relations. Most auxiliary relations belong to the former. Experiment shows that if one sentence has only head-initial dependency relation or head-final dependency relation, parsing difficulty will be greatly reduced. Such sentences are few and far between in real text. But dependency relation between subtrees can be turned into either head-initial or head-final dependency relations at a certain stage in parsing. To this end, we first use rules to annotate auxiliary relations in parsing...
algorithm integrating rules and statistics. This is because dependency distance of auxiliary relations is relatively short, and their structural rules are relatively simple. Dependency distance is a major factor influencing parsing. We test UAS and LAS of MParser-1, MParser-2 and MParser-3 in the testing corpus. As dependency relation affects both UAS and LAS in pretty much the same way, Figure 4 only presents UAS’ distribution.

![Figure 4. Parsing effect on dependency relations over different distances (UAS)](image)

In this figure, UAS decreases as dependency distance increase, with few exceptions. This trend is more pronounced in MParser-1 or MParser-3. MParser-3 is rather hierarchical with a distinct hierarchy observed respectively for dependency distance 1, dependency distance 2-4 and dependency distance greater than 4. This hierarchy is related to the division of dependency ranges. MParser-1 and MParser-2 are internally flawed, but are complementary with each other on account of their respective advantages. As such an effective integration of these two parsers is the best parsing strategy.

5. Conclusions and Future Work

In the case of limited Treebank resources, it has good processing ability to incorporate recognition rules into statistical parsing model. Statistical model has strong robustness in syntactic parsing, while grammatical rules have the characteristics of accuracy and less computation. After the fusion of the two, higher recognition rate and efficiency are obtained in the actual work. In Mongolian syntactic parsing some short-distance dependency relations can be determined by morphological forms and part of speech features, such as auxiliary relationship, direct object-predicate relationship and noun modification relationship. After these features are added into the statistical model in the form of rules, this study has achieved good results. The problem is that the statistical model uses a simple lexical dependency probability model, so the parsing ability is relatively moderate. In the follow-up research, we continue to expand the scale and precision of dependency Treebank, add semantic classification information, and then use more complex statistical model or neural network model to improve the accuracy of analysis.

Acknowledgements

This work is partially supported by National Natural Science Foundation Project (the project No. is 61662050,) and National Social Science Project (the project No. is 10CYY022).

References

[1] Nasunurtu (1998). Mongolian Information Processing, Wulaanhad: Inner Mongolia Science and Technology Press.
[2] Dabhurbayar (2008). Several Questions on Basic Mongolian Verb Phrases, Journal of Inner Mongolia University (Mongolian Version), 37(5), 30-39.
[3] Wang Serguleng (2009). Rule-based Mongolian Sentence Automatic Segmentation Procedure, Journal of Inner Mongolia University (Mongolian Version), 38(3), 51-55.
[4] Chen X, Zhou Y, Zhu C, et al. (2015). Transition-based Dependency Parsing Using Two Heterogeneous Gated Recursive Neural Networks. Proceedings of the 2015 Conference on
Empirical Methods in Natural Language Processing. Lisbon, Portugal: Association for Computational Linguistics, 2015:1879-1889.

[5] Dyer C, et al. (2015). Transition-based dependency parsing with stack long short-term memory. Computer Science, 37(2), 321-332.

[6] Tai K S, Socher R, Manning C D. (2015). Improved semantic representation from tree-structured long short-term memory networks. Computer science, 5(1), 36.

[7] Zheng X, Peng H, Chen Y, et al. (2015). Character-based parsing with convolutional neural network. Proceedings of the 24th International Conference on Artificial Intelligence. Austin, Texas, USA: AAAI Press, 1054-1060.

[8] Kiperwasser E, Goldberg Y. (2016). Simple and accurate dependency parsing using bidirectional LSTM feature representations. Transaction of the Association of Computational Linguistics, 4(1), 313-327.

[9] Kurita S, Kawahara D, Kurohashi S. (2017). Neural joint model for transition-based Chinese syntactic analysis. Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 1:1204-1214.

[10] S.Loglo (2016). Research on Automatic Dependency Parsing of Mongolian Language, Hohhot: Inner Mongolia University Press.

[11] Qiligeer (2017). A quantitative study of some problems in written and spoken Mongolian, Master's degree thesis of Inner Mongolia University.

[12] Chaomurilige (2018). Dependency parsing of "Tungalag Tamir", Master's degree thesis of Inner Mongolia University.

[13] Nanding (2016). Description and recognition of Mongolian Object-Predicate Relation based on dependency grammar, Master's degree thesis of Inner Mongolia University.

[14] Qinggeletu Bai and S.Loglo (2019). Mongolian indirect object-predicate relation recognition based on dependency grammar, Journal of Inner Mongolia University for Nationalities (Social Sciences), 44(3), 30-34.

[15] Eisner, J.M. (1996). Three new probabilistic models for dependency parsing: An exploration. Proceedings of ACL, P340-345.

[16] Ma Jinshan (2007). Statistical Method-based Chinese Dependency Parsing, Doctoral dissertation of Haerbin Institute of Technology.

[17] S.Loglo, HUA Shabao and Sarula (2012). A statistical method-based Mongolian dependency parsing model, Journal of Chinese Information Processing, 26(3), 27-32.