International Journal of Knowledge and Language Processing
Volume 5, Number 2, **2014**

www.ijklp.org
KLP International ©2014 ISSN 2191-2734
pp.32–42

Construction of a Mongolian Dependency Treebank

S．Loglo and Sarula

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

Selected Paper from Chinese Lexical Semantic Workshop 2014

ABSTRACT. *Mongolian information processing has generally been done on word and phrase level and is now en route to sentence processing. To offer training and evaluation data for subsequent parsing, we have constructed a Mongolian Dependency Treebank (MDTB) using automatic annotation and manual proofreading based on the 1-million-word modern Mongolian corpus developed by Inner Mongolia University. MDTB contains 461,240 words and 31,722 sentences with sentence lengths averaging 14.54 words and dependency distances 2.31. Of the total dependency relations, head-initial dependency relations account for 21.4% and head-final dependency relations 78.6%.*

Keywords: Mongolian, dependency grammar, treebank

1. **Introduction.** Treebank is a parsed text corpus based on part-of-speech (POS) tagging. As a knowledge source to acquire syntactic structure and a tool to evaluate parsing results, treebank is becoming increasingly appreciated in recent years. More and more researches suggest that treebank is a useful research tool in both computational linguistics and theoretical linguistics and lays a solid foundation for syntactic research with the multitude of information on syntactic distribution. Languages used to be annotated and parsed mainly using phrase structure grammar. Yet in recent years, dependency grammar is gaining currency due to its simple form and ease for annotation and application. It has been widely used in parsing languages such as English, Japanese, Chinese, Germany and Czech and improved forthwith. CoNLL (Computational Natural Language Learning) put dependency grammar-based evaluation into its shared tasks [1-4] in four consecutive years from 2006 to 2009, indicating the growing popularity of dependency grammar-based parsing and annotation as a research area down the road.

2. **MDTB Annotation Scheme.** A dependency annotation scheme with a properly sized tag set and a detailed annotation standard is the key to studying Mongolian dependency parsing

and building treebank resources. Content wise, tag set describes dependency relations between words in a sentence while annotation standard covers the methods and regulations on how to determine the dependency relations between words. MDTB annotation scheme is broken down into the following parts, tag set, annotation standard and annotation types.

2.1. **MDTB Tag set.** Dependency treebank tag sets come in different sizes for different languages. By way of context, Czech Treebank PDT2.0 uses 7 tags [5], Germany Treebank TIGER 49 [6], and HIT SCIR Chinese Dependency Treebank 34. Tsinghua Chinese Treebank (TCT) used 106 tags initially before reducing the number to 44[7]. Studies suggest that tag set cannot be too big or too small. If tag set is too small, the accuracy and depth of annotation will suffer. If tag set is too big, a complete and accurate annotation may obtain, but the cost would be too high. On the other hand, parsing a huge text would lead to aggravated data sparseness, a smaller coverage parser and reduced robustness.

Our tag set is built on the classification and naming methods of the relations within phrases in traditional Mongolian such as subject, attribute, object, adverbial, conjunction relations and auxiliary relations [8]. But the dependency relations in this paper are not limited to relations within phrases because dependency relations between words, phrases, sentence elements and subsentences may also exist. In terms of the nature of dependency relations, the same dependency relation across different language units has the same function.

TABLE 1. MONGOLIAN DEPENDENCY RELATION TAG SET

Relation Type	**Dependency Relation**	**Tag and description**
Special Relation	key word in a sentence	HEAD
	independent element	INDE(independent)
Dominance relation	Subject	SUBJ (subject)
	direct object	DOBJ (direct object)
	indirect object	IOBJ (indirect object)
	Attribute	ATT(attribute)
	Adverbial	ADV(adverbial)
Conjunction relation	Coordinate	COO(coordinate)
	Appositive	APP(appositive)
	Summarization	SUM(summarization)
Auxiliary relation	time-local words - auxiliary	TL-AUX(time-local words – auxiliary)
	postposition - auxiliary	PP-AUX (postposition – auxiliary)
	modal particles - auxiliary	MP-AUX(modal particles – auxiliary)
	modals – auxiliary	M-AUX(modals - auxiliary)
	auxiliary verbs - auxiliary	AV-AUX(auxiliary verbs – auxiliary)
	contact verb - auxiliary	CV-AUX(contact verb – auxiliary)
Non-syntactical elements	conjunction - auxiliary	CJ-AUX(conjunction – auxiliary)

A syntactic annotation scheme shall be built in a way that (1) saves as much storage room as possible, (2) is easily operable and easy for manual annotation, (3) facilitates knowledge acquisition and (4) is easily converted into representation of other language systems and is easy to generate semantic representation.

In reference to the successful dependency treebanks in Japanese [9] and other languages, we have built a preliminary tag set that contains 17 dependency relations under five categories, adapted to the characteristics of Mongolian, as seen in Table 1.

This tag set is proven to cover all the dependency relations in the experiment texts thus annotated without any redundant information. It thus indicates that this tag set is sufficient to annotate the Mongolian dependency relations.

2.2. **MDTB Annotation Standard.** Dependency relations between words are composed of head, dependent, dependency direction and dependency type. Location wise, they can be divided into head-initial dependency relation and head-final dependency relation.

Head-initial dependency relation is formed when head is in front of dependent. Save a few exceptions (like preposed modal particles), auxiliary and summarization in Mongolian belong to this category which is described by a directed arc pointing from dependents to heads.

When dependent is in front of head, head-final dependency relation obtains. But for a few exceptions (such as inverted sentence), subject, object, attribute and adverbial relations all belong to this category which is described by a directed arc pointing from dependents to heads. It should be noted that contrary to the definition of dependency grammar, coordinate words are not in a dominance relation. To comply with the definition of dependency grammar, however, we temporarily take the first word as head and the second dependent when two words are in a coordinate relation. Appositive words are also treated as such. In other words, we take the first word as head and the second dependent when two words are in an appositive relation.

We have made a rather detailed annotation standard for 17 types of dependency relations under 5 categories, on which I will not elaborate for the interest of the size of the standard.

2.3. **MDTB Annotation Type.** Annotation type impacts the readability, utility, editing and proofreading of syntactically annotated corpus to some extent. Based on the annotation types used in PDT, TIGER, TCT and HIT SCIR Chinese Dependency Treebanks and considerations of storage cost and ease for editing and proofreading, we adopt the following two annotation types, viz. bracket annotation and tree annotation.

2.3.1. **Bracket Annotation.** The commonly used bracket annotation is to list head (if any) or the corresponding subscript of the head in the bracket behind each word, and then annotate the dependency relation type and direction behind the head or head subscript. The bracket annotation used in this paper is presented as follows.

W（i→j：rel）

W denotes the i^{th} word of the sentence, i the subscript of the word, j the head's

subscript, rel the type of dependency relation , rel∈annotation set.

Bracket annotation example:

[]C0GMANDVL , (1→18 : SUBJ) []GANGGAM_A-YI (2→3 : DOBJ) TEBERIJU (3→18 : ADV) ABVGAD (4→3 : AV-AUX) , TVLG_A-YIN (5→6 : ATT) GAL-DV (6→8 : IOBJ) VLAYITAL_A-BAN (7→8 : ADV) HALAGSAN (8→9 : ATT) HACAR-IYAR-IYAN (9→18 : IOBJ) []GANGGAM_A-YIN (10→16 : ATT) HEGER_E-YIN (11→12 : ATT) JIBAR-TV (12→13 : IOBJ) DAGARAGSAN (13→16 : ATT) HUITEN (14→16 : ATT) VLAGAN (15→16 : ATT) HACAR-I (16→18 : DOBJ) NI (17→16 : MP-AUX) DVLAGACAGVLVN_A . (18→19 : HEAD) <EOS>.

(Meaning: Chaogeman holds Ganggama in his arm and press his hot cheek against her red cold face so as to keep her warm). The corpus thus annotated is a plain text which can be opened by any editing software and can be evoked in multiple systems.

2.3.2. **Tree Annotation.** MDTB is developed using automatic annotation and manual proofreading. Given that manual proofreading of bracket parse tree is laborious and time-consuming, we have developed MTEditer, a visual and functionally complete tree structure editing software.

The tree structure used in MTEditer is presented in Figure 1.

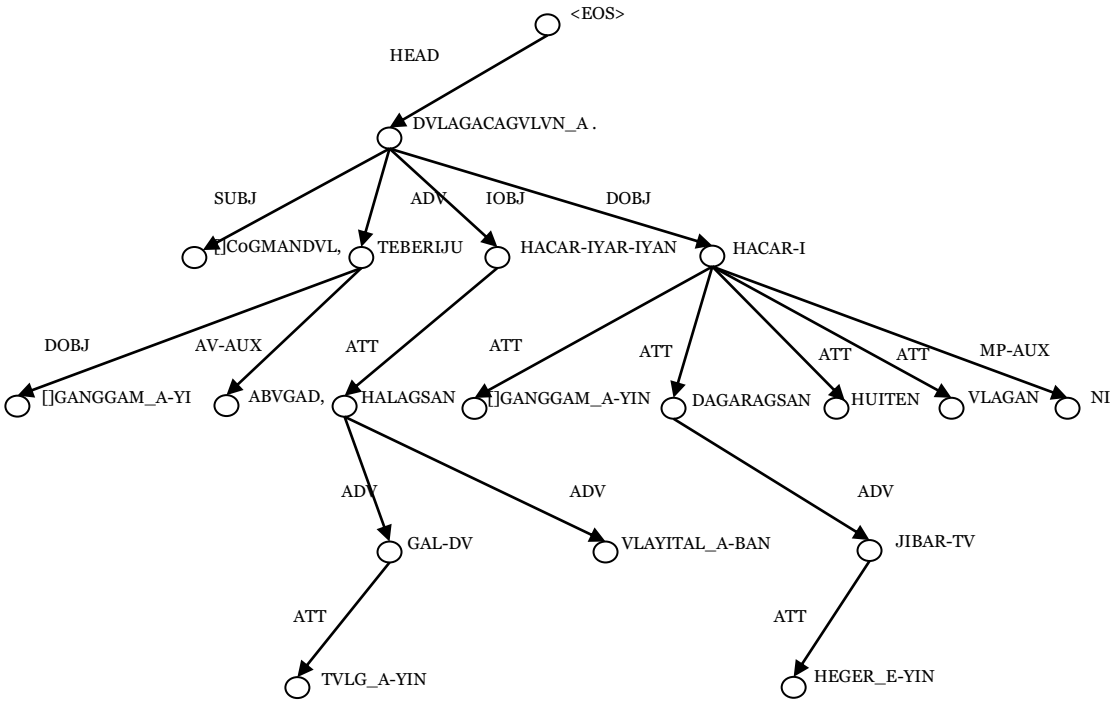

FIGURE 1. TREE ANNOTATION OF MONGOLIAN DEPENDENCY RELATION

3. **Construction of Treebank.** Manual or automatic annotation is often used in the construction of a treebank. MDTB is constructed by automatically annotating and manually proofreading 12 Mongolian text books (6 middle school books and 6 high school books)

extracted from the 1-million-word *Modern Mongolian Corpus* developed by Mongolian College of Inner Mongolia University. MDTB contains 461,240 words and 31,722 sentences. We will introduce the construction method and construction process of MDTB from the perspective of automatic annotation and manual proofreading.

3.1. **Automatic Annotation.** The rich morphological changes in Mongolian make it easy for syntactic rules to be summarized and extracted. Initially we developed a rule-based parser MParser-1 which includes sentence segmentation, syntactic fragment identification, dependency analysis within fragments and sentence dependency analysis. Dependency analysis is further divided into framework analysis and dependency annotation.

Syntactic fragment identification is a key step in dependency analysis for long sentences. The internal structure of fragments is analyzed before dependency relation between fragments is established. Doing so would reduce the difficulty in parsing [10]. In order to analyze raw corpus, we integrated into MParser-1 compound words automatic identification, POS tagging and parsing, among other functions.

The complete parsing process in MParser-1 is roughly divided into the following nine steps.

The first step is pretreatment. Raw corpus text is edited to meet the requirement of parser.

The second step is compound words annotation. Compound words in the text are annotated using compound words dictionary and relevant identification rules.

The third step is sentence segmentation. Figure 2 shows that the text is segmented into n-independent sentences based on segmentation rules, and n sentence nodes are built for the treebank.

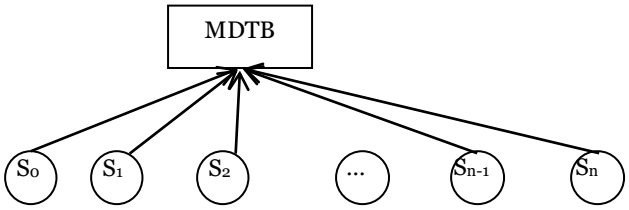

FIGURE 2. MDTB'S FIRST-LAYER STRUCTURE CHART

In this graph, MDTB denotes Mongolian Dependency Treebank and S_0, S1, S2, S_{n-1} and S_n represent the n sentence nodes.

In the fourth step, sentences are segmented into word nodes. As presented in Figure 3, a word node is generated for each word under each sentence node (one node for one compound word). Each word node remains dependent on the virtual <EOS> node in the sentence before the sentence is parsed.

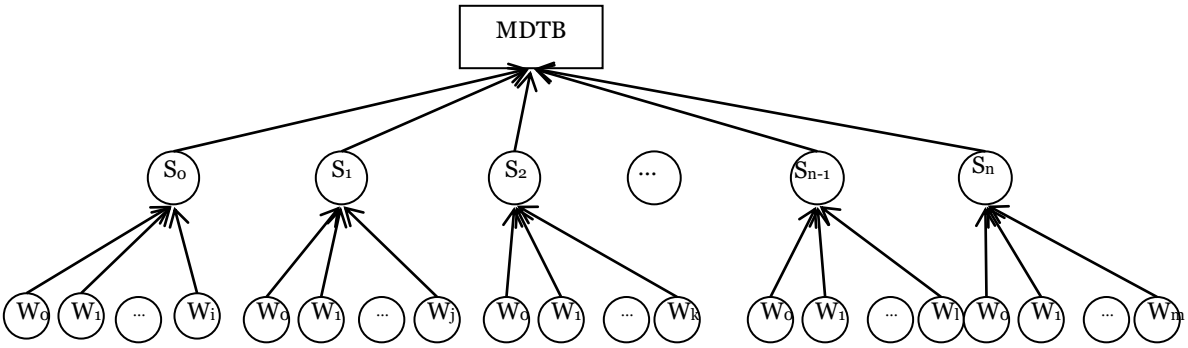

FIGURE 3. MDTB'S SECOND-LAYER STRUCTURE CHART

In this graph, W_0,W_1，．．．W_i denote word nodes in each sentence.

In the fifth step, each word node is analyzed to retrieve lexical information such as POS, subclass and morphological changes. The retrieved information is then stored in the property fields corresponding to the current node so as to prepare for segment identification and parsing in the following step.

The sixth step is syntactic segment identification. The lexical information and punctuation retrieved in step 5 are used to divide the sentence with n word nodes into m syntactic fragments ($1 \leq m \leq n$).

In the seventh step, parsing is done within fragments.

In the eighth step, dependency relation between fragments is built.

In the ninth step, preposed adverbial and subject in some syntactic fragments (normally the first syntactic fragment) may be adverbial or subject of the whole sentence, which are adjusted based on corresponding rules.

3.2. **Manual Proofreading.** In the automatically annotated MDTB, dependency relations within fragments account for 90% of the total dependency relations in the whole sentence and are annotated highly accurately. The small-numbered dependency relations between fragments, however, are not so accurately annotated because many complex dependency relations such as coordinate, subject, object and adverbial exist between them and the dependency distances are quite long. As such manual proofreading must focus on (1) building or correcting dependency relations between fragments, and (2) finding and correcting wrong collocations and dependency relations.

Syntactic proofreading example: during automatic annotation, the following sentence, *JVN-V HALAGVN EDUR-UD-TU MVHVLAG=TERGE HOTOLOGSEN EMUN_E GAJAR-VN AYANCID GAL GAL-IYAR CVBVJV ARV JUG OGEDELEN_E* (In a hot summer, many southern travelers file in a van bound for the north), is segmented into three fragments as presented in Figure 4, but no dependency relations are built between these fragments. Manual proofreading is thus tasked to build dependency relations between three fragments and correct wrong ones, HOTOLOGSEN→EDUR-UD-TU: ADV, EMUN_E→CVBVJV: ADV and JUG→OGEDELEN_E:SUBJ. Figure 5 shows the manually proofread dependency tree.

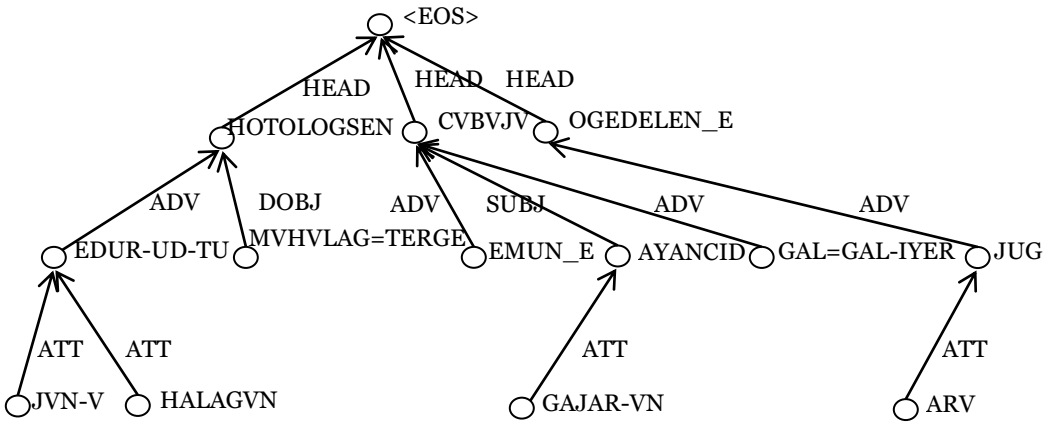

FIGURE 4. AUTOMATICALLY ANNOTATED DEPENDENCY TREE

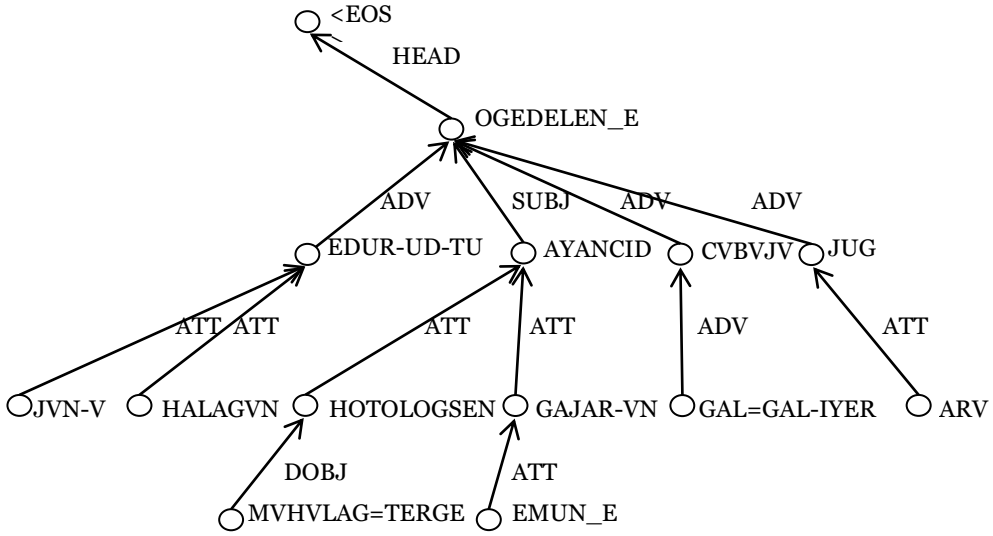

FIGURE 5. MANUALLY PROOFREAD DEPENDENCY TREE

When we built MDTB more than one year ago, both annotation and manual proofreading fully observed the annotation scheme we had laid down. But inconsistencies occurred due to the difference in understandings proofreaders had over some ambiguous structures. To solve these unavoidable inconsistencies, we annotated special structures using human-computer-interaction. For example, "NOHOR BATV" (Mr Batu) had been annotated as appositive by some but as attribute by others before we settled with appositive eventually. Is such treatment right? We believe that in dependency treebank developed for information processing, determining collocation relation is more important than determining dependency type because the former represents dependency relation between words but the latter is only a man-made classification. As such, for any particular dependency relation, changing the annotation type will not impact the dependency framework of the entire sentence.

To ensure consistency for annotations done by many proofreaders, we also laid down detailed rules on some special structures that are still controversial in traditional linguistic

research or whose dependency relations remain hard to be determined. Whenever these special structures were annotated in line with the relevant rules, the annotations were deemed correct.

Eventually we extracted 1,000 sentences from the treebank for consistency check. Statistically, the number of inconsistent and wrong dependency relations is less than 5% of the total, indicating that the treebank is reliable for training and testing parser.

3. **Statistics of dependency distance, dependency types and sentence length.** Dependency distance is the linear distance between head and dependent [11], i.e. the difference between locations of two words between which a dependency relationship holds. There are now two methods to calculate dependency distance. Many foreign scholars argue that the dependency distance between two neighboring words in a dependency relation is 0. But Chinese researchers posit that this figure should be 1 when it comes to calculation of Chinese dependency distance. We used the second method to perform statistical analysis on MDTB which returned the distribution of Mongolian dependency distances (as presented in Figure 6) and the mean value of dependency distances (as shown in Table 2).

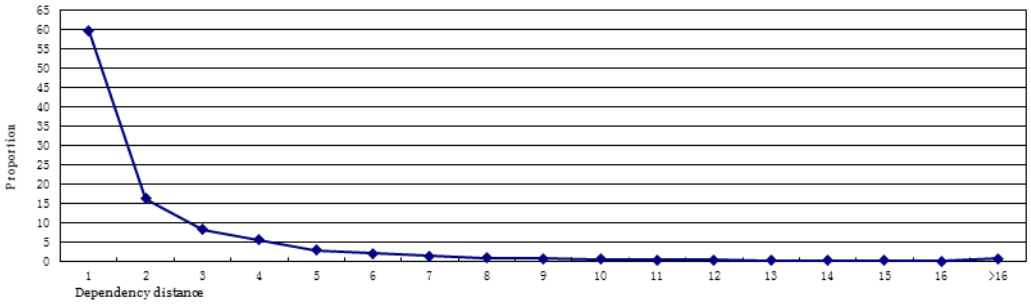

FIGURE 6. DISTRIBUTION OF DEPENDENCY DISTANCE IN MDTB

TABLE 2. MEAN OF DEPENDENCY DISTANCES IN DIFFERENT DEPENDENCY RELATIONS

Dependency Relation	Mean of Dependency distances	Dependency Relation	Mean value of Dependency distances
HEAD	1.82	TL-AUX	1.01
SUBJ	3.61	MP-AUX	1.13
ATT	1.39	PP-AUX	1.02
ADV	3.01	M-AUX	1.41
DOBJ	1.71	AV-AUX	1.02
IOBJ	2.06	CV-AUX	1.15
SUM	1.08	CJ-AUX	3.96
APP	1.40	INDE	5.04
COO	6.06		**Average 2.31**

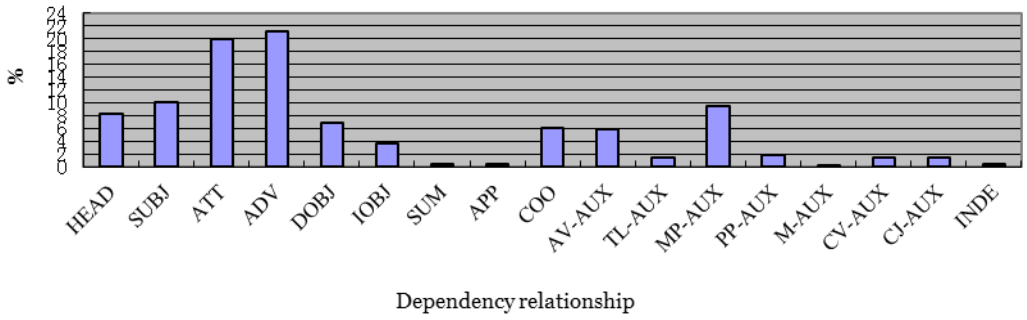

FIGURE 7. PROPORTION OF DEPENDENCY RELATIONS IN MDTB

Figure 7 presents the proportion of each dependency relation in the Treebank.Statistically, ATT, COO and APP are only in head-final dependency relation and SUM, PP-AUX, M-AUX, TL-AUX and CV-AUX are only in head-initial dependency relation, whereas SUBJ, ADV, DOBJ, IOBJ, INDE, AV-AUX, MP-AUX and CJ-AUX are in both head-initial dependency relation and head-final dependency relation. The details are presented in Table 3.

TABLE 3. STATISTICS OF HEAD-DEPENDENT AND TAIL-DEPENDENT RELATIONS

Dependency Relation	Head-initial relation%	Head-final relation%	Dependency Relation	Head-initial relation%	Head-final relation%
SUBJ	0.50%	99.50%	INDE	23.49%	76.51%
ADV	0.43%	99.57%	AV-AUX	99.25%	0.75%
DOBJ	0.27%	97.73%	MP-AUX	92.49%	7.51%
IOBJ	0.14%	99.86%	CJ-AUX	48.60%	51.40%

Sentence length is another major factor influencing the accuracy of syntactic analysis. Figure 8 shows the distribution of sentence lengths. It can be seen from this graph that sentences with 5 to 10 words are the most frequent and the number of sentences continuously and regularly decreases as the number of words increase in sentences. Sentence lengths in MDTB average 14.54 words.

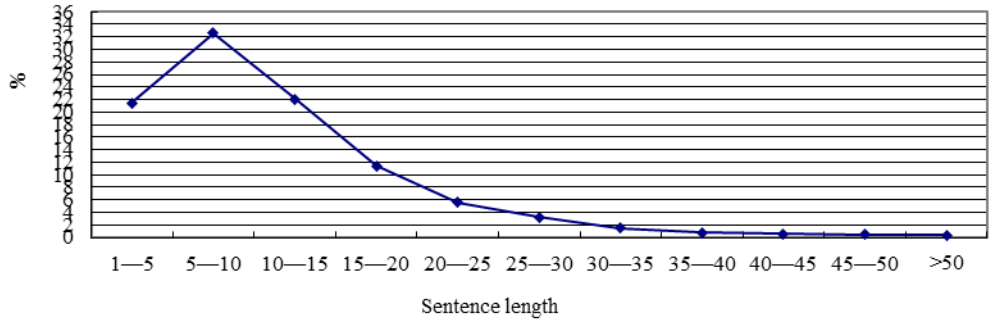

FIGURE 8. STATISTICS OF SENTENCE LENGTH IN MDTB

5. **Summary.** Automatic annotation and manual proofreading wise, the annotation set has a reasonable size and contains a reasonable number of dependency types. It covers all

dependency phenomena in Mongolian. And the treebank thus annotated contains sufficient information on syntactic structure. But the annotation standard is not detailed enough. Many inconsistencies were spotted in the process of manual proofreading. This increased the workload of proofreaders and reduced the reliability of treebank corpus. Moving forward, we will improve the existing annotation standard of the treebank, making it better adapted to the Mongolian characteristics and more appropriate for manual annotation and automatic analysis.

Block annotation and tree annotation have their respective advantages. The former is better in terms of storage room and universality because it uses text file. The latter runs faster in treebank loading, program processing and computability because of the structured document used. Tree annotation also has better readability, which is helpful for manual proofreading.

In sum, our annotation scheme is relatively mature but the details of annotation standard need to be further improved.

Acknowledgment. This work is partially supported by National Natural Science Foundation of China (the project No. is 61262046).

REFERENCES

[1] Sabine Buchholz, Erwin Marsi, CoNLL-X Shared Task: Multi-lingual Dependency Parsing, In *Proceedings of the Tenth Conference on Computational Natural Language Learning*, pp.149–164, 2006.

[2] Joakim Nivre, Johan Hall, Sandra Kübler et al, The CoNLL-2007 Shared Task on Dependency Parsing, In *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNL*, pp.915–932, 2007.

[3] Mihai Surdeanu, Richard Johansson, Adam Meyers et al, The CoNLL-2008 Shared Task on Joint Parsing of Syntactic and Semantic Dependencies, In *Proceedings of the 12th Conference on Computational Natural Language Learning*, pp.159–177, 2008.

[4] Jan Hajič, Massimiliano Ciaramita, Richard Johansson et al, The CoNLL-2009 Shared Task on Syntactic and Semantic Dependencies in Multiple Languages, In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, pp.1–18, 2009.

[5] Max Jakob, Mapping the Prague Dependency Treebank Annotation Scheme onto Robust Minimal Recursion Semantics, *Master's Thesis*, 2009.

[6] Ines Rehbein, Josef van Genabith, Treebank Annotation Schemes and Parser Evaluation for German, *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, Prague, pp.630-639, 2007.

[7] MA Yongjun, The Current Status and Prospect of Natural Language Processing Based on Dependency Grammar, *Academic Exchange*, Serial Number 163, Number 10, pp.137–140, 2007.

[8] Qinggeltei, *Mongolian Grammar*, Inner Mongolia People's Publishing House, Huhhot, pp.215–222, 1997.

[9] Shinsuke Mori, Hideki Ogura and Tetsuro Sasada, A Japanese Word Dependency Corpus, in *Proceedings of the Nineth International Conference on Language Resources and Evaluation*,

pp.753–758, 2014.

[10] S.Loglo and Sarula, A Rule-Based Mongolian Dependency Parsing Model, *International Journal Of Knowledge And Language Processing*, Volume 4, Number 3, pp.27–37, 2013.

[11] LIU Haitao, *Dependency Grammar（From Theory to Practice）*, Science Press, BeiJing, 2009.