Language Resources for Mongolian

Jaimai Purev
Center for Research on Language Processing,
National University of Mongolia
Ulaanbaatar-210646, Mongolia
purev@num.edu.mn

Chagnaa Altangerel
Center for Research on Language Processing,
National University of Mongolia
Ulaanbaatar-210646, Mongolia
altangerel@num.edu.mn

Abstract

Mongolian language is spoken by about 8 million speakers. This paper summarizes the current status of its resources and tools used for Mongolian language processing.

1 Introduction

As to its origins, the Mongolian language belongs to the Altaic language family, and typologically, it is an agglutinative language. Mongolian is the national language of Mongolia and today it has about 8 million speakers around the world including Mongolia (2.7 mln), Inner Mongolia in China (3.38 mln), Afghanistan (?) and Russia (0.5 mln) (Gordon 2005, Purev 2007). Nowadays, Mongolian is written in two official scripts: the Cyrillic Mongolian Script and the (old, uigur) Mongolian Script. The old Mongolian script was used before introduction of the Cyrillic script in 1940s. But, the Cyrillic Mongolian is predominately used in everyday life and also on the Internet. Thus, it is used also for most of research on local language processing and resource development.

Today, manually produced resources such as dictionary are available in Mongolian, and their use is closed to research purpose. The resource such as text and speech corpora, tagged lexicon and a large amount of dictionary in digital content are needed for further Mongolian language processing.

The paper focuses on the Mongolian language resources and tools developed, which can be used for research in computational linguistics and local language processing.

2 Mongolian Language Resources

Mongolian is a less developed language in the computer environment. There have been very few digital resources and research work for it. Recently, several research works aiming to develop local language resources, which are a 5 million word text corpus, a 3 thousand word speech corpus for Mongolian speech recognition, and a 1,500 sentence speech corpus for text to speech training, have begun, respectively. In last few years, some research projects such as Mongolian Text To Speech (TTS) (InfoCon, 2006) and Mongolian script converter (Choimaa and Lodoisamba, 2005) have implemented. Currently, these projects are inactively used in research and public usage.

In the following chapters we will introduce Mongolian language resources and various tools developed till now.

2.1 Text Corpus for Mongolian

Center for Research on Language Processing (CRLP) at the National University of Mongolia (NUM) in Mongolia has been developing corpus and associated tools for Mongolian. A recent project collected a raw corpus of 5 million words of Mongolian text mostly from domains like daily online or printed newspapers, literature, and laws. This corpus was reduced to 4.8 million words after cleaning and correcting some errors in it. The cleaned corpus comprises 144 texts from laws; 278 stories, 8 novelettes, and 4 novels from literature; 597 news, 505 interviews, 302 reports, 578 essays, 469 stories, and 1,258 editorials from newspaper, respectively. The domain-wise figures are given in Table 1.

Conference on Human Language Technology for Development, Alexandria, Egypt, 2-5 May 2011.

Domains	Cleaned Corpus	
	Total Words	Distinct Words
Literature	1,012,779	78,972
Law	577,708	15,235
publish	2,460,225	118,601
Newspaper “Unen Sonin”	949,558	61,125
Total	**5,000,270**	**192,061**

Table 1. Distribution of Mongolian Corpus

In this corpus 100 thousand words have been manually tagged. As building the corpus for Mongolian, we have developed other resources such as part of speech (POS) tagset, dictionary, lexicon, etc based on the corpus.

2.2 Speech Corpus

Based on the previous 5 million word corpus 1500 phoneme balanced sentences are selected and built speech corpus. This corpus is labeled and used for training Mongolian TTS engine. Beside the corpus, there are also 20 vowel phonemes and 34 consonant phonemes (which make total 54 phonemes) identified.

In mission to develop the Mongolian speech recognition system a general isolated-word recognizer for Mongolian language (native 5 male and 5 female speakers, 2500 isolated words dictionary) has been constructed with using the HTK toolkits (Altangerel and Damdinsuren 2008). Its recognition accuracy is about 95% on isolated word basis.

2.3 Machine readable dictionaries and thesaurus

An English-Mongolian bilingual dictionary which is based on Oxford-Monsudar printed dictionary has about 43K headwords which have about 80K senses. Each head word has its POS tag and each sense has its appropriate key words for collocation and/or senses. Those keywords are used for disambiguating word senses in English-Mongolian machine translation system.

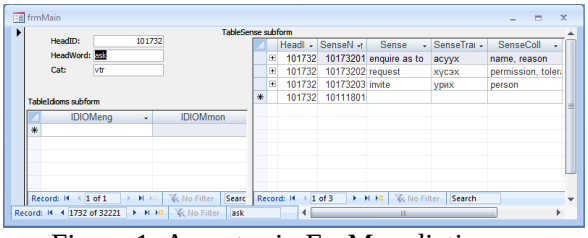

Figure 1. An entry in En-Mon dictionary

Secondly, there is a corpus of digitized Mongolian monolingual dictionary of 35K entries, with their corresponding examples. From these entries about 10K nouns were selected and created a hierarchy of hyponymy via manual and semiautomatic method (the tool is introduced in the following subsection).

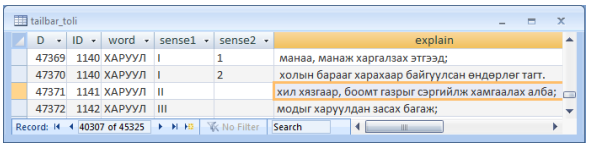

Figure 2. Entries in the monolingual dictionary

Additionally, there is an English-Mongolian idiom dictionary created from frequently used idioms with about 2K entries. In addition to English idiom and corresponding Mongolian translation, each entry has a regular expression to identify it in a sentence.

to toe the party line	\b(toe|toes|toeing|toed)\b the party line	намын дэг жаягийг яштал баримтлах; намын ёс журмыг яштал баримтлах
to touch a raw nerve	\b(touch|touchs|touching|touched)\b a raw nerve	эмзэг газрыг нь олж хатгах
to touch nerve	\b(touch|touchs|touching|touched)\b nerve	эмзэг газрыг нь хөндөх
to treat sb with kid gloves	\b(treat|treats|treating|treated)\b (.*) with kid gloves	хүний эвийг олох
to turn in one's grave	\b(turn|turns|turning|turned)\b in (.*'s|his|my|her|our|their) grave	ас нь өнгөлөх

Figure 3. Some entries in an idiom dictionary

For example, to recognize the all possible forms of the idiom *to toe the party line* we have set the regular expression as *\b(toe|toes|toeing|toed)\b the party line*, and since it is a verbal phrase, we have selected the verb *toe* as a main constituent. In further applications such as machine translation, this idiom is seen as a verb and its tense is reflected in its main constituent *toe* as toes, toed, toeing etc. In more detail, the sentence *He toed the party line after he joined the club* is simplified to *He toed* after he joined the club*. Analyses such as POS tagging and/or syntactic parsing will be simpler afterward, since the whole idiom is replaced with only one constituent (a verb in this example) without any change of the sentence constituents. After analyzing the simplified sentence the translation of the idiom is put into the verb *toed**, which is in the past tense.

Beside those dictionaries a smaller size dictionary of proper noun and abbreviation is also compiled and is being enriched.

2.4 Tools

There are also additional tools available through CRLP for text normalization, dictionary based spell checking, POS tagging and word, sentence segmentation, word syllabication etc. These will be introduced in the following parts.

Spell Checker

We have developed a dictionary-based spell-checker for cleaning a Mongolian corpus. The overall system is comprised of a user GUI, a word speller, a word corrector, a lexicon and corpora as shown in the following figure.

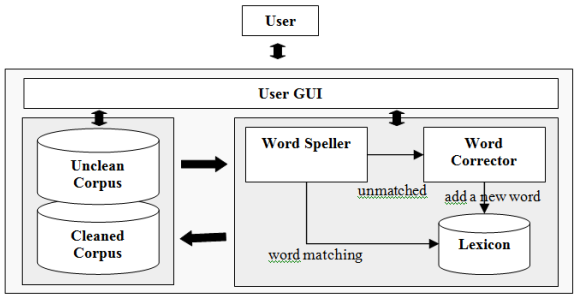

Figure 4. Overall architecture of Mongolian Spell-checker

Word speller is an important part of the system. It will perform an operation of checking an input word that is given by a user. For doing that, the input word is compared to the words in a dictionary the speller contains. The dictionary is loaded into memory from a lexicon, currently containing around 100 thousand words. Its data structure is a tree shown in the following figure.

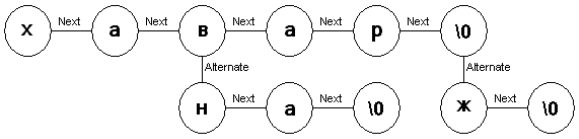

Figure 5. Tree Structure of Dictionary

Here is an experiment that shows the effective memory usage and data structure. For loading 24,385 headwords whose characters are 184,777 in total into the memory, the number of tree nodes allocated into memory is 89,250, or two times less than that of the actual words characters. Thus, this kind of tree structure is suitable to storing a large amount of words and retrieving a word from them in a short time.

Word corrector is a part to correct an input word if it is not found in the dictionary, or could be misspelled. This module of the system finds possible correct forms of the input word by matching it with the dictionary used in the word speller. A correcting algorithm considers four kinds of misspelling errors. First, one of the letters in a word is mistyped, for example, a word *father* can be mistyped as *tather*. Second, a letter is mistakenly inserted into a word, for example, *father* can be mistyped as *fatherr*. The third error type is that one letter of a word is missed, for example, father can be mistyped as *fathr*. Last, two letters of a word can be exchanged, for example, *father* can be mistyped as *fahter*. The algorithm is potential to correct a wrong word whose up to three letters are misspelled.

The spell checker needs more development in the future. Since it uses only a dictionary, that is a simple list of words, it automatically corrects words without accounting about the context in which the word occurs. Sometimes it leads to a problem when working on a optically character recognized (OCRed) texts. OCRed texts are not mistyped words but they are misrecognized words and in this case spell checker needs to be more intelligent.

Unicoded PC-KIMMO and two level rules

History of using a finite state tool (FST) in Mongolian morphology begins earlier [Purev etc, 2007]. Currently, for processing Mongolian words, generator and recognizer part of famous morphological parser PC-KIMMO was updated to support Unicode. Besides this, Mongolian morphological rules were compiled in 84 two level rules. These rules account the vowel harmony of the Mongolian language, which was not done in previous attempts. The system generates and recognizes with high level of accuracy. It achieves about 98 percent on 1,000 words covering all the types (29 classes for nouns, 18 classes for verbs) in Mongolian morphological dictionary. In the future system it needs to include word grammar files.

Text normalization tool

Since the collected texts are stored in digital form, they have two problems which are mixed character encoding and miss-file encoding. Cyrillic characters are encoded in either American Standard Code for Information Interchange (ASCII, Windows 1251 encoding) or Unicode. Thus some files contain texts written in both encodings. Also some UTF files are saved in normal .txt files even though they contain Unicode texts. That is why we developed a tool, named "Text file Converter", for changing the files encoded with Unicode to UTF8, and for changing ASCII characters into Unicode ones. This tool first checks file's encoding and converts to UTF. Afterwards it checks mixed character encodings and fixes them. We have developed these tools from scratch instead of using existing converter, because we needed to do fixing of mixed encoding within a file, but existing converter tools mainly converts between homogenously encoded files.

Mongolian raw corpus has some problems needed to be cleaned, such as: ASCII and Unicode mixture, Interchanged characters: Characters with similar shapes are used interchangeably, Latin and Cyrillic characters used in a same word, Letters in number, such as ‘o’ is used as ‘0’ in numbers: “35oo”, Characters located closely on the keyboards are mistyped, longer words separated by hyphen(s), Quotations used asymmetrically, Character order is changed, Words combined without delimiters, Misspelled words etc.

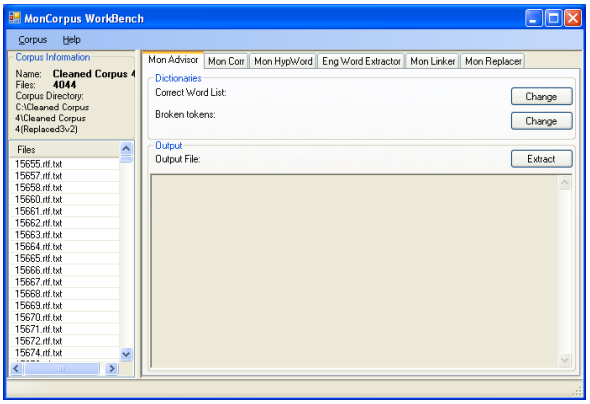

Figure 6. A Mongolian corpus normalizing tool, “MonCorpus Workbench”

To remove those we have developed Mongolian Corpus Workbench, a set of semi-automatic cleaning and normalizing tools.

POS Tagger

Incorporated knowledge of the language in corpus is used as the main data for the development and application of the corpus-based systems. Specially, part of speech tag is a key to the language processing.

Initially, manual POS tagger, followed by unigram, bigram and trigram taggers are developed.

The manual tagger, named MTagger, is used for tagging the text in the XML format. In a corpus building, there is a need to analyze raw text and tagged text except tagging, and we also needed the tool for such purpose. Therefore, we developed some text analyzing tools such as searching, filtering and computing statistical information, and plugged in MTagger.

POS Tagset editing: The user can edit the POS tagset freely during the tagging and can create own POS tagset. The tagset is designed to be in a common text file. The file format of POS tagset is very easy to understand and each line of a tagset file contains individual tag information that consists of tag and its description. Also some tags are grouped into one group as a ">,separator" line that must be in below position of the last tag of the group tags.

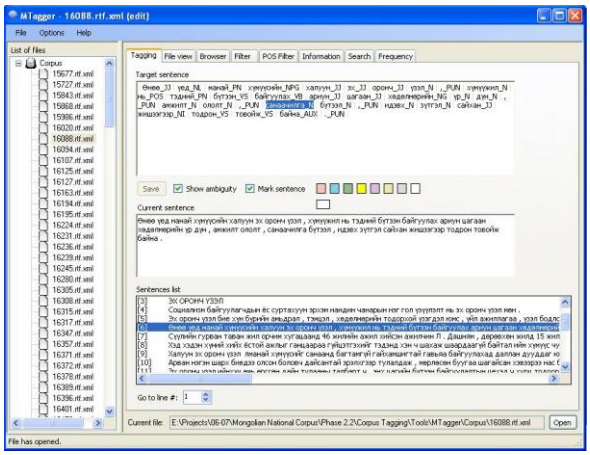

Figure 7. Manual tagger Mtagger.

POS tag suggestion: POS tag suggestion helps users to assign a corresponding POS tag to a word and lists appropriate POS tags based on corpus words and tags frequency. If you have created list of word frequency from the corpus, MTagger will suggest appropriate POS tags when you are assigning a tag.

Ambiguity tag bookmarks: Users may encounter ambiguity case because of one or more tags may seem to be assigned a given word. If users are not sure to assign POS tag for selected word or they cannot choose one of believable POS tags, they can use bookmarks for that. In fact, users should use ambiguity POS tag "?" in such situation, then attaches ambiguity POS tags for selected word. After that, users can search bookmarked words.

Auto-completion tool: The auto-completion tool lets you choose POS tags to auto-complete names for you while documents are being tagged. This functionality requires accessibility to each of the POS tags you want to use. To enable auto-completion, select each of a word that you want to use auto completion in the auto-completion control. This ability allows users to tag documents without mouse.

Searching word in documents: Users can search word only from a document that is being tagged. Mostly, the searching function is needed when user is editing a previously tagged file and the searching function highlights the words to be searched in the file. Then, user can select a sentence with searching word and view on Tagging field. There are two searching options; match case and match whole word.

Word comparison by POS tag: MTagger compares words by POS tags from a working file or whole corpus. The limitation of MTagger is users can select at most 5 POS tags for comparing words. Shown in a table, result of comparing can be printed. The result of comparing shows the words and their frequencies. The first column of the result is words, second is their frequencies. A total number of words and a total number of tags are shown in the bottom of each result.

Searching word using XPath: It is possible to search using XPath expression only from a selected file. By entering a POS tag in POS field and a suffix tag in Suffix field, MTagger creates automatically XPath expression. Or, user who has enough knowledge of XPath expression can write easily in XPath field. Selecting words with options locate in file view or tagged view from the result, a selected word is located either editing file field or tagging field.

Word frequency: As counting word frequency, it creates corpus statistic information and list of word frequency file which is created from all of corpus files, with stat extension. The name of stat file is same as corpus name that it contains information about how many parts-of-speech tags and how often it is tagged for each word. After counting word frequency, POS suggestion function of MTagger will be activated.

Statistical information: MTagger shows neither selected file nor Corpus statistic information. For file information, paragraphs, sentences and tagged words are included and for corpus information, number of files, number of tagged files, tagged words (tokens) and distinct tagged words (word types) are included.

Because of the statistical issue occurring in insufficient training data, in a trigram model we have also taken unigram and bigrams into account. This method was first used in TnT tagger [Brants, 2000] and known as one of the widely used methods in the field. The trigram tagger was trained on 450 sentences which includes 60K words and tested on the 10K word texts of various contents. The following table shows the accuracy of the trigram tagger.

Text	Text 1	Text 2	Text 3	Average
#Words	10,390	11,858	11,000	11,083
OOV, %	3.2	14.6	19.6	12.5
GuessEnd, %	40.3	45.6	26.4	37.4
TotalAcc, %	**95.8**	**90.3**	**83.3**	**89.8**

Table 2. Accuracy of trigram tagger

Out of vocabulary (OOV, not trained) words in the test set was about 13 percent (OOV column) and the tagger guessed tags (GuessEnd column) based on the word endings. It is shown that as the number of OOV word increases the accuracy of the tagger decreases. It mainly depends on the length of the endings used in the guessing algorithm.

Manual and Semiautomatic WordNet Creating Tool for Mongolian

Manual editor for Mongolian lexical semantic network (LSN) is developed in VS #C [Altangerel, 2010]. User interface shows Network in a tree structure, Vocabulary or word sense repository, Detail fields, editing section. Also it has functions of filtering entries, Adding new nodes by drag and drop from entry list to hierarchy/tree via mouse, Editing LSN /by right mouse click: Adding, Removing, Editing a node.

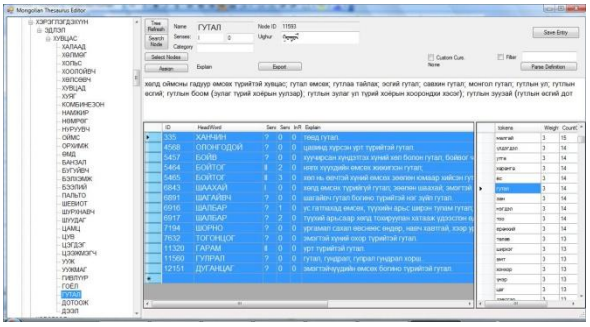

Figure 8. Semi-automatic tool for Mongolian lexical semantic network

A semiautomatic tool has additional modules for clustering, definition parsing etc. It uses some language specific heuristics to select the feature of the word from its defintition in monolingual dictionary [Altangerel and Cheol-Young, 2010]. User can select a paritcular cluster and remove some entries from it and assign it to the network.

Syllabication and phoneme tools

In the framework of Mongolian TTS development, syllabication tool and phoneme frequency counter are built.

Transfer rule editing tools

For building rule based machine translation system, transfer and generation rule editing tools have been developed in Java.

The generation tool retrieves some patterns from the parsed corpus based on Tregex (tree regex) [Roger and Galen, 2006] patterns and does some operations (remove, insert, move, rename etc) on the tree.

Additional to the pattern search user can also use dependency information in the query. With this tool we have created about 300 generation

rules for English Mongolian machine translation system.

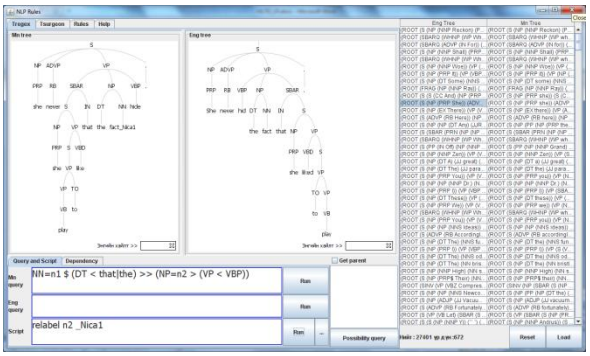

Figure 9. Generation rule editor

Transfer rule editing tool searches a phrase structure grammar from sentences in the Penn Treebank corpus and allows editor to set the constituent order into target language. We have set the transfer order (from English to Mongolian) for the most frequent 5K rules from Penn Treebank.

English-Mongolian Machine Translation Engine

Finally, rule based English-Mongolian machine translation engine should be introduced here. The engine uses most of the resources mentioned above and translates English sentence into Mongolian.

It is mainly developed in Java, based on open source tools and put in public use via web interface. More information about this engine is given in a separate paper.

3 Conclusion

This paper lists some core linguistic resources of Mongolian and related tools, available through CRLP and other sources. For dissemination it needs to be addressed.

The resources and tools in Mongolian language are being used in systems such as Mongolian text to speech engine, English to Mongolian machine translation system etc.

We have created the tools mainly from scratch, instead of using existing tools, because of some peculiarities in Mongolian language, and also of some requirements of the data origin.

As our experience in relating to develop Mongolian language processing for years, we have faced some situations challenging us. They are that how to improve local people interests in Mongolian language processing, lack of researchers and staffs who have experience and knowledge on language processing and government issues to support local language processing.

4 Acknowledgement

Some work described in this paper has been supported by PAN Localization Project (PANL10n), ITU-AMD and ARC.

TTS for Mongolian project is undergoing and it is supported by ITU-AMD with cooperation with NECTEC, Thailand.

Reference

Altangerel, A. and Damdinsuren, B. 2008. *A Large Vocabulary Speech Recognition System for Mongolian Language*. the Proceedings of Oriental COCOSDA 2008 Workshop, Koyoto, Japan

Altangerel Chagnaa. 2010. *Lexical semantic network for Mongolian*, Khurel Togoot national conference proceeding, pp 207-210, Ulaanbaatar, Mongolia.

Altangerel Chagnaa and Cheol-Young Ock. 2010. Toward Automatic Construction of Lexical Semantic Networks, 6th International Conference on Networked Computing (INC), 2010, ISBN: 978-1-4244-6986-4

Brants T. 2000. *TnT – a Statistical Part-of-Speech Tagger*, in Proc. sixth conference on applied natural language processing (ANLP-2000), Seattle, WA, 2000.

Choimaa Sch. and Lodoisamba S. 2005. *Mongol helnii tailbar toli* (Descripitve dictionary of Mongolian).

Christopher D. Manning and Hinrich Schutze. 1999. *Foundations of Statistical NLP*, MIT Press.

Daniel Jurafsky, James H. Martin. 2000. *Speech and Language Processing*, Singapore.

Gordon, Raymond G., Jr. (ed.). (2005). *Ethnologue: Languages of the World*, Fifteenth edition. Dallas, Tex.: SIL International.

InfoCon Co., Ltd, TTS for Mongolian: http://www.infocon.mn/tts/

Purev Jaimai, Tsolmon Zundui, Altangerel Chagnaa, and Cheol-Young Ock. 2007. *PC-KIMMO-based Description of Mongolian Morphology*. International Journal of Information Processing Systems, Vol. 1 (1), pp. 41-48.

Purev Jaimai. 2007. *Linguistic Issues in Mongolian Language Technology*. In the Proceedings of 1st Korea-Mongolia International Workshop on Electronics and Information. Chungbuk national University, Cheongju, Korea.

Roger Levy and Galen Andrew. 2006. *Tregex and Tsurgeon: tools for querying and manipulating tree data structures*. 5th International Conference on Language Resources and Evaluation (LREC 2006)