A Novel Approach to Improve the Mongolian Language Model using Intermediate Characters

Xiaofei Yan, Feilong Bao(✉), Hongxi Wei, and Xiangdong Su

College of Computer Science, Inner Mongolia University, Hohhot 010021, China
Xiaofeiyan_h@sina.com, {csfeilong,cswhx,cssxd}@imu.edu.cn

Abstract. In Mongolian language, there is a phenomenon that many words have the same presentation form but represent different words with different codes. Since typists usually input the words according to their representation forms and cannot distinguish the codes sometimes, there are lots of coding errors occurred in Mongolian corpus. It results in statistic and retrieval very difficult on such a Mongolian corpus. To solve this problem, this paper proposed a method which merges the words with same presentation forms by Intermediate characters, then use the corpus in Intermediate characters form to build Mongolian language model. Experimental result shows that the proposed method can reduce the perplexity and the word error rate for the 3-gram language model by 41% and 30% respectively when comparing model trained on the corpus without processing. The proposed approach significantly improves the performance of Mongolian language model and greatly enhances the accuracy of Mongolian speech recognition.

Keywords: Mongolian language • Intermediate characters • N-gram language model • Speech recognition

1 Introduction

Mongolian language has a wide influence in the world. It is used in China, Mongolia, Russia and other countries where the pronunciations are almost the same but the writing forms are different from each other. The Mongolian language used in China is called "Traditional Mongolian". The corresponding Mongolian language used in Mongolia is called "Cyrillic Mongolian", which letters are borrowed from the alphabets of Russian. In this paper, the Mongolian language particularly refers to the traditional Mongolian language.

In the traditional Mongolian language, its alphabet contains 35 letters and each letter has several different presentation forms. The concrete presentation forms are determined by their positions (initial, medial or final) occurred in words. It leads to a phenomenon that many Mongolian words have the some presentation forms with different codes. In fact, most people only input the words according to their presentation forms without considering their codes. Therefore, the words bearing the same presentation forms with different codes may be incorrect in the Mongolian text. It results in the statistical information from the corresponding Mongolian corpus is inaccurate and weakens the performance of the language model in Mongolian information processing, such as speech recognition [1], information retrieval [2], machine translation [3] and so on.

Table 1. Example of the same presentation forms

No	Presentation form	Position in word	Nominal form	Keyboard mapping	Code
1	ᠠ / ᠠ	medial, final	ᠠ	a	1820
			ᠡ	e	1821
			ᠨ	n	1828
2	ᠠ	final	ᠠ	a	1820
			ᠡ	e	1821
3	ᠣ / ᠣ	medial, final	ᠣ	q	1823
			ᠤ	v	1824
			ᠥ	o	1825
			ᠦ	u	1826
			ᠸ	w	1838
4	ᠣ / ᠣ	initial, medial	ᠣ	q	1823
			ᠤ	v	1824
5	ᠥ / ᠥ / ᠥ / ᠥ	initial, medial, final	ᠥ	o	1825
			ᠦ	u	1826
6	ᠢ / ᠢ	medial, final	ᠢ	i	1822
			ᠶ	y	1836
7	ᠲ / ᠲ	initial, medial	ᠲ	t	1832
			ᠳ	d	1833
8	ᠬ / ᠭ / ᠬ / ᠬ / ᠭ / ᠬ / ᠬ	initial, medial	ᠬ	h	182c
			ᠭ	g	182d
9	ᠵ	medial	ᠵ	j	1835
			ᠢ	i	1832
10	ᠸ	medial	ᠸ	w	1838
			ᠣ	q	1823

To solve the above problem, several correction approaches were proposed in the literature. Chuanjie Su [4] and Jun Zhao [5] adopted language model to correct the coding errors. Sloglo [6] proposed a correction method based on the finite automata. Bule Jiang [7] used a rule-based approach to deal with the correction problem. However, these approaches can only correct part of the coding errors, and the words that do not follows the spelling rules or out of vocabulary (OOV) cannot be corrected.

This paper proposes a novel approach by using a kind of Intermediate characters to express the words being the same presentation forms with the different codes. In detail of our approach, the words can be converted into same forms by the Intermediate characters. Then, the language models (without processing, in Intermediate characters) were constructed. And these language models are compared by perplexity and accuracy of Mongolian speech recognition. Experimental results show that the proposed approach not only greatly reduced the perplexity of the N-gram language model, but also greatly reduced the word error rate (WER) of Mongolian speech recognition.

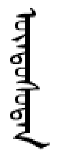
The rest of the paper is organized as follows. Section 2 presents the characteristics of Mongolian encoding. Section 3 describes the Mongolian Intermediate characters. Section 4 gives the Mongolian language model based on the Intermediate characters. Section 5 briefly introduces the process of speech recognition. Section 6 shows the experimental results. Section 7 draws the conclusion.

2 Characteristics of Mongolian Encoding

Mongolian characters contain two character types: nominal characters and presentation characters. According to Universal Coded Character Set (UCS) ISO/IEC 10646 and PRC National Standard GB 13000-2010, Mongolian character set only includes the nominal characters, and the units larger than one letter or less than one letter are not encoded. Generally, Mongolian letter set refers to the nominal characters (also known as nominal form). It is suitable for Mongolian writing, transmission, processing, storage, displaying. A few coding standards that created by some commercial companies use the presentation characters to encode Mongolian words [8].

Mongolian letter set contains 35 nominal characters and 58 presentation forms. Each nominal characters has several presentation forms according to its positions in words [8]. Table 1 shows Mongolian nominal characters and its corresponding presentation forms. From Table 1, we can see that some characters have different nominal forms but same presentation forms.

There is a phenomenon that many words have correct presentation forms but with the incorrect Mongolian code in Mongolian corpus. The reasons are twofold: first, the pronunciations of some letters are often confused in Mongolian dialects, such as the vowels "u" and "v", the vowels "o" and "u", the consonants "t" and "d", and thus Mongolian people living this regions often make many typo errors in text; second, some typists only care about whether the presentation forms of the words are correct or not, rather than the codes of these words, and freely replace the correct letter with another one with same presentation forms. The typo errors in Mongolian corpus makes it difficult for us to count, retrieval of the text, as well as training Mongolian language model. We use an example to illustrate this.



correct spelling :
undusuten

undusuden(36187)	**undusuten(24708)**	undvsvden(7902)	undvsvdan(5141)
ondosoden(2403)	undusudan(1989)	undusutan(1895)	undqsqden(1828)
undvsvten(1181)	ondvsvdan(976)	untusuten(915)	undusudee(869)
ondqsqden(860)	undvsvdaa(840)	ondvsvden(788)	untusutee(723)
uedvsvden(706)	undqsqdan(661)	untvsvtan(658)	ondosoten(650)
undvsuden(622)	undusvden(510)	uedvsvdee(474)	undusudaa(450)
uadvsvdan(406)	uadqsqden(363)	undvsudan(281)	undvsvtan(259)
uedqsqdan(256)	ondqsqdan(245)	uadusudee(240)	uedvsvdan(235)
uedusuden(230)	uadusudan(217)	oedvsvdan(199)	oetvsvten(193)
uadqsqdan(189)	untvsvten(187)	undusvdan(177)	ondosodan(160)
undqsuden(158)	undqsvden(136)	ondvsoden(132)	ondvsvten(128)
ondosotan(128)	undvsqden(128)	uetqsqtea(123)	undusoden(118)
oetvsvtan(115)	undvsuten(114)	undqsqtan(113)	undqsqten(106)
uadvsvden(106)	uadvsvdaa(100)		

Fig. 1. Different spelling and frequency about the same Mongolian word "ᠦᠨᠳᠦᠰᠦᠲᠡᠨ"

For the Mongolian word "ᠦᠨᠳᠦᠰᠦᠲᠡᠨ" (meaning: minority), its keyboard mapping is "undusuten". According to the analysis on a Mongolian corpus including 76 million Mongolian words, this word appears 102532 times, and only 24708 times of its codes are correctly. The other 78124 ones are typed as other words with the same presentation forms. Actually, there are 291 words that have the same presentation forms as the word "ᠦᠨᠳᠦᠰᠦᠲᠡᠨ" (meaning: minority). Figure 1 shows the Mongolian word "ᠦᠨᠳᠦᠰᠦᠲᠡᠨ" (meaning: minority) and its typos whose frequency is greater than 100 in the corpus.

3 Mongolian Intermediate Characters

This paper puts forward a novel method to represent the Mongolian words according to the characteristics of Mongolian presentation forms. This method uses Latin characters to represent the nominal letters, and the nominal characters with same presentation forms are represented by the same Latin characters. These Latin characters are called Intermediate characters. In some cases, a nominal character is converted into multiple Intermediate characters depending on its positions in words. That is, the words in same presentation forms are mapping to the same Intermediate characters string. It is worth pointing out that a string of Latin characters can only correspond to a string of Intermediate characters and a string of Intermediate characters can correspond to one or multiple strings of Latin characters with some presentation forms.

In experiment, we compare the language models built on the Mongolian corpus without processing and the corresponding Mongolian corpus represented in Intermediate characters.

Table 2. Normalize Rules on different nominal form with same presentation characters

<table>
<tr><th>No</th><th>Match regular expressions (RE)</th><th>Replace RE</th><th>Priority</th><th>Interpretation</th></tr>
<tr><td>1</td><td>(g|h)([eiouIU])</td><td>G$2</td><td>4</td><td>When "g" and "h" are in initial of the word and the next letter is "e","i", "o", "u", it uses "G" to replace "g" and "h"</td></tr>
<tr><td>2</td><td>^[_"'`&*\^]{0,}(u|o)([[:ANY:]])</td><td>U$2</td><td>4</td><td>When "u" and "o" are in initial of the word, they will be replaced with "U"</td></tr>
<tr><td>3</td><td>^[_"'`&*\^]{0,}(w|v)([[:ANY:]])</td><td>V$2</td><td>4</td><td>When "w" and "v" are in initial of the word, they will be replaced with "V"</td></tr>
<tr><td>4</td><td>([^ghG])["'`]{0,}(a|e)["'`]{0,}$</td><td>$1A</td><td>5</td><td>When the previous character is "non-g" or "non-h", "a" and "e" in the final of word will be replaced with "A"</td></tr>
<tr><td>5</td><td>([^ghG])["'`]{0,}(a|e)([[:ANY:]])</td><td>$1A$3</td><td>4</td><td>When the previous character is "non-g" or "non-h", "a" and "e" will be replaced with "A"</td></tr>
<tr><td>6</td><td>([[:VOW:]])["'`]{0,}(yi|ii)([[:ANY:]])</td><td>$1I$3</td><td>5</td><td>When the previous character is a vowel, "yi" and "ii" will be replaced with "I"</td></tr>
<tr><td>7</td><td>([[:VOW:]])["'`]{0,}y["'`]{0,}([[:CSNT-W:]])</td><td>$1I$2</td><td>4</td><td>When the previous character is a vowel and the next character is consonant of "non-W", "y" is replaced with "I"</td></tr>
<tr><td>8</td><td>([[:VOW:]])["'`]{0,}h["'`]{0,}([[:CSNT:]])</td><td>$1g$2</td><td>4</td><td>When the previous character is a vowel and the next character is consonant, "h" is replaced with "g"</td></tr>
<tr><td>9</td><td>([nmlNrsd])["'`]{0,}aaa([[:ANY:]])</td><td>$1ha$2</td><td>6</td><td>When the previous character is "n", "m", "l", "N","r", "s", "d"; "aaa" will be replaced with "ha"</td></tr>
<tr><td>10</td><td>([[:ANY:]])ng["'`]{0,}([[:CSNT:]])</td><td>$1N$2</td><td>4</td><td>When the next character is a consonant, "ng" will be replaced with "N"</td></tr>
</table>

In this paper, we use the regular expression to convert Mongolian words into Intermediate characters form. This takes the advantages of regular expression that the rules can be easily expressed by regular expressions. At the same time, we integrate some rules to correct the spelling errors. In this paper, we summarize 116 transformation and correction rules of Intermediate characters. We do the related statistics that these rules can cover most of the Mongolian word. Table 2 shows part of these rules. " _"'&*^ " represents the Mongolian control character; ":ANY:" represents of any Mongolian letter; ":VOW: " represents all vowels; ":CSNT:" means all consonants; "G", "U", "V", "A", "I" and other characters represent Intermediate forms which are defined. In Table 2, No 1-7 is the Intermediate characters conversion rule and No 8-10 is the correct rule. For example, the 291 Mongolian words having the same presentation forms as Mongolian words "ᠦᠨᠳᠦᠰᠦᠲᠡᠨ" (meaning: minority) will be converted into the same Intermediate characters string "UnTOsOTAn".

4 Language Model Establishment

Language model is a mathematical model to describe the inherent laws of natural language. It is the core of computational linguistics. In theory, the structure of language model is to induce, discover, and obtain the inherent laws of natural language in statistical and structural aspects. Language model are crucial components in many Natural Language Processing (NLP) applications, such as speech recognition, handwriting recognition, machine translation, information retrieval and so on.

N-gram language model [9] has been widely used in statistical language model. The probability of a Mongolian word sequence $w = w_1 w_2 \dots \dots w_m$ can be written in the form of conditional probability:

$$p(w) = p(w_1 w_2 \dots \dots w_m) = \prod_{i=1}^{m} p(w_i | w_1^{i-1}) \approx \prod_{i=1}^{m} p\left(w_i \middle| w_{i-n+1}^{i-1}\right) \quad (1)$$

The probability of the m-th words w_m depends on all the words $w_1 w_2 \dots \dots w_{m-1}$. We can now use this model to estimate the probability of seeing sentences in the corpus by providing a simple independence assumption based on the Markov assumption [10]. Corresponding to the language model, the current word is only related to the previous n-1 words. From the equation (1), we can see that the target of language model is how to estimate the conditional probability of the next word in the list using $p(w_i | w_{i-n+1}^{i-1})$. The most commonly probability estimation method we used is the maximum likelihood estimation (MLE).

$$p\left(w_i \middle| w_{i-n+1}^{i-1}\right) = \frac{c(w_{i-n+1}^{i})}{c(w_{i-n+1}^{i-1})} \quad (2)$$

$c(*)$ means the total count of the N-gram in the corpus. However, a drawback of the MLE is that the N-tuple corpus which does not appear in the training set will be given zero-Probability. This is not allowed in the NLP. Smoothing algorithm can be used to solve this kind of zero-Probabilities problem. In this paper, we us the Kneser-Ney smoothing algorithm [11].

Based on the Intermediate characters to build language model, the Mongolian word sequence $w = w_1 w_2 \dots \dots w_m$ are converted into its corresponding Intermediate characters $w' = w_1' w_2' \dots \dots w_m'$. We can use N-gram probabilities to approximate this as:

$$p(w') = p(w'_1 w'_2 \dots \dots w'_m) \approx \prod_{i=1}^{m} p\left(w'_i \middle| w'^{i-1}_{i-n+1}\right) \tag{3}$$

5 Mongolian Speech Recognition

Speech recognition allows the machine to turn the speech signal into text or commands through the process of identification and understanding [12]. The process of speech recognition mainly includes pre-processing, feature extraction, model training, decoding, post-processing. Its basic structure is shown in Figure 2. Pro-processing consists of pre-filtering, sampling, quantization, adding window, endpoint detection, and pre-emphasis towards the speech signal. Feature extraction is to effectively extract the features form the speech signal. Speech decoding is to look for the maximum probability of the output word sequences toward the speech signal, which greatly relies on the acoustic model, language model and pronunciation dictionary and is carried out by Viterbi algorithm [15]. For the speech feature $x = x_1 x_2 \dots \dots$and its corresponding word sequence $w = w_1, w_2 \dots \dots$, the formula of speech recognition based on the maximum a post-probability (MAP) is shown as follows:

$$\widehat{w} = \underset{w}{argmax}\{p(w|x)\} \approx \underset{w}{argmax}\{p(x|w)p(w)\} \tag{4}$$

Form the equation (4), we just need to calculate the maximum product of $p(x|w)$ and $p(w)$. $p(x|w)$ is the probability of speech feature vector sequence x under the condition of word sequence w, which is determined by the acoustic model. $p(w)$ is the probability of a word sequence w, which is determined by the language model.

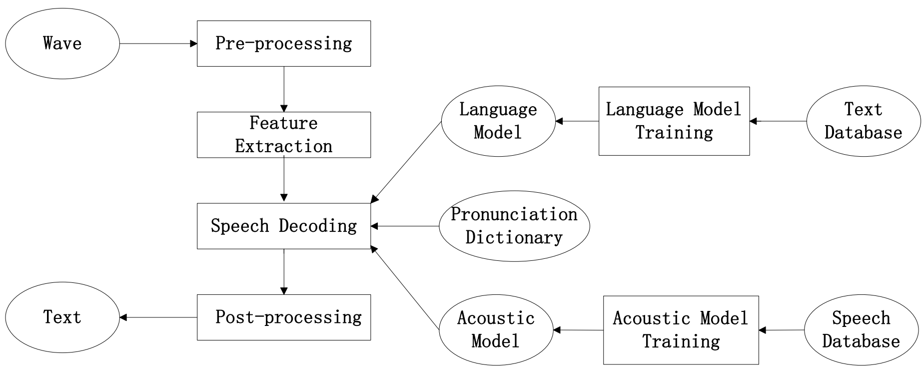


Fig. 2. The structure of automatic speech recognition

Currently, mainstream technology is that use the N-gram and recurrent neural network (RNN) [13,14] to build language model, combining with deep neural network (DNN) and hidden Markov model (HMM) for acoustic model. This paper uses the language model of 3-gram and the acoustic model of DNN or LSTM in Mongolian speech recognition. The language model (without processing, in Intermediate characters) was respectively created, compared in perplexity experiments and speech recognition experiments.

6 Experiment

In this paper, the performance of the language model was verified on the Mongolian corpus through a series of experiments, including the perplexity experiments and speech recognition experiments. The evaluation metrics are perplexity (PPL) of language model and WER of speech recognition.

6.1 Perplexity

6.1.1 Data

Table 3. Statistics of Mongolian corpus

Dataset	Sentences	Tokens	Vocabulary	OOV
Train_Mon	3033075	70886970	1778571	
Train_IC	3033075	70886970	**1033450**	
Test_Mon	337008	7886050	444289	123504
Test_IC	337008	7886050	**263301**	**70439**

The datasets is constructed by the page content coming from *mgyxw.net*, *mongol.people.cn*, *holvoo.net* and more than 20 other websites. Part of those website uses the Mongolian menksoft encoding and the other uses the Mongolian standard encoding. In order to unify coding, the text of Mongolian menksoft encoding was converted into Mongolian standard encoding. The correct rate reached over 99% by using the Mongolian conversion toolkit (http://mtg.mglip.com) which was developed by Inner Mongolia University. The Mongolian corpus used for constructing the language model is about 1.2G in data size. In this paper, 10-fold cross-validation method was used to evaluate the experimental performance. Table 3 lists the details of the training set and testing set (without processing, in Intermediate characters) in average. We use Train_Mon and Test_Mon to represent the Training set without processing and Testing set without processing respectively, and use Train_IC and Test_IC to represent the Training set in Intermediate characters form and Testing set in Intermediate characters form respectively. Tokens refer to the number of Mongolian words in the dataset; vocabulary means the number of distinct words; OOV represents the word which is not included in the training set but in the testing set. It is worth noting that, the case suffixes in the corpus are treated as individual tokens in Mongolian language model training. This is a widely used technique in Mongolian language processing.

In Table 3, the number of vocabulary relatively reduced by 41.89% in training set and 40.74% in testing set. The OOV of testing set decreased 42.97%. The reduction of vocabulary was due to Mongolian spaces which were incorrectly used. For example, case suffix "ᠪᠠᠷ" (correspond to the keyboard: "-bar") and its stem should have been continuously wrote with the Mongolian space (correspond to the keyboard: "-") in the text. However, some text use the common spaces instead of Mongolian space in front of many case suffixes and possessive suffixes. The vocabulary of the corpus in Intermediate characters form is greatly reduced compared to that of the corpus without processing. The reduction ratio of vocabulary and OOV can be seen that many Mongolian words have same presentation forms but different codes.

6.1.2 Evaluation of Language Model

Perplexity is a common metric to measure the performance of a language model. The perplexity PPT (T) of the model $p(w_i|w_{i-n+1}^{i-1})$ is defined as follows:

$$PPT(T) = 2^{H_p(T)} = 2^{-\frac{1}{W_T} log_2 P(T)} \tag{5}$$

where $H_p(T)$ represents the cross entropy for the testing data T in the model $p(w_i|w_{i-n+1}^{i-1})$. It is a basic criterion for evaluating language model performance that the lower perplexity is, the better performance of the language model is.

Table 4. The perplexity of Mongolian language model

Model	Perplexity PPT(T)		
	1-gram	2-gram	3-gram
Trained on Train_Mon and tested on Test_Mon	6130.706	609.0156	238.1425
Trained on Train_IC and tested on Test_IC	**2887.565**	**349.8624**	**139.3985**

Table 4 shows the performance of N-gram language model using SRILM toolkit [16]. We trained 1-gram, 2-gram and 3-gram language model toward the corpus without processing and the corresponding Mongolian Intermediate characters, respectively. It can be seen from Table 4, the perplexity of 1-gram, 2-gram and 3-gram language model was relatively reduced by 52.9%, 41.98% and 41.46% respectively when the training corpus represented in Intermediate characters form. It is clear that the performance of the Mongolian language model has been significantly improved by our proposed approach.

6.2 Speech Recognition

6.2.1 Dataset

This experiment takes the Kaldi [17] speech recognition system as the platform using state-of-the-art acoustic models trained on the Mongolian corpus. The dataset contains approximately 78 hours of speech, in which 70 hours of speech (62794 sentences) is used to train the acoustic model training and which 8 hours of speech (6987 sentences) acts as testing set. The pronunciation dictionary consists of 38235 words.

Mongolian corpus used for constructing the language model is the same as experiment 6.1. In addition, this experiment is performed on the basis of perplexity experiment.

6.2.2 Evaluation of Speech Recognition

When testing, the acoustic model of speech recognition system is kept unchanged and the language model is changed to calculate, compare the WER in the experiment. The evaluation metric is WER defined as follows:

$$WER = (I + D + S)/N \tag{6}$$

where *I, D, S* are the numbers of numeric insertions, deletions and substitutions, respectively. N is the total number of numeric entities in the corpus.

Table 5. The WER under different Mongolian language model

Model	Word error rate WER/%		
	1-gram	2-gram	3-gram
Baseline DNN (Trained on Train_Mon and tested on Test_Mon)	30.66	16.47	12.37
DNN (Trained on Train_IC and tested on Test_IC)	**27.01**	**12.37**	**7.83**
LSTM (Trained on Train_Mon and tested on Test_Mon)	20.0	10.35	9.05
LSTM (Trained on Train_IC and tested on Test_IC)	**18.06**	**8.11**	**5.97**

In Mongolian speech recognition, we compared the WER for the 1-gram, 2-gram and 3-gram language model toward the corpus without processing and the corresponding Mongolian Intermediate characters, respectively. Experimental results are shown in Table 5. We can see that the WER in testing set with the language model trained on the dataset Train_IC is significant lower than that with the language model trained on the dataset Train_Mon. Meanwhile, the WER of DNN+3-gram model [18] and LSTM+3-gram model has been respectively reduced by 36.7%, 34.03% using Intermediate characters, greatly improving the performance of Mongolian speech recognition. It also proves that converting the training corpus into Intermediate characters form can make the language model performs better.

7 Conclusion

This paper presents a method that combining different presentation form using Intermediate characters to build Mongolian language model. The experimental results show that this method decreases the vocabulary by 41% and reduce the perplexity of 3-gram language model by 41.46%. Meanwhile, the WER for the 3-gram language model decrease around 30% when comparing with the language model trained without processing in the Mongolian speech recognition. This approach not only effectively improves the performance of Mongolian language model, but also greatly enhances the accuracy of Mongolian speech recognition. It is of great significance to related technological development of Mongolian natural language processing.

In future, we will investigate the processing approach of this kind of word in order to improve the retrieval and statistics performance of the Mongolian words.

Acknowledgements. This research was partially supported by the China National Nature Science Foundation (No.61263037 and No.61563040), Inner Mongolia nature science foundation (No.2014BS0604) and the program of high-level talents of Inner Mongolia University.

Reference

1. Feilong Bao, Guanglai Gao, Xueliang Yan, Weihua Wang.: Segmentation-based Mongolian LVCSR Approach. In: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 8136-8139 (2013)
2. Guanglai Gao, Wei Jin, Feilong.: Mongolian Text Retrieval: Language Specialities and IR Model. In: Journal of Computational Information Systems, pp. 1561-1568 (2009)
3. Feilong Bao, Guanglai Gao, Xueliang Yan, Hongwei Wang.: Language Model for Cyrillic Mongolian to Traditional Mongolian Conversion. In: The 2nd Conference on Natural Language Processing & Chinese Computing (NLPCC 2013), Chongqing, pp. 13–18 (2013)
4. Zhao Jun.: Design and Implementation of Mongolian Word Analyzing and Correcting Based on Statistical Language Method. Hohhot: Inner Mongolia University (2007)
5. Chuanjie Su, Hongxu Hou, Yangping, Huarui Yuan.: Based on the Statistical translation framework of the Monglian automatic spelling correction method. In: Journal of Chinese Information Processing, pp. 175-179 (2013)
6. Sloglo.: A Proofreading Algorithm of Mongolian Text Based on Nondeterministic FiniteAutomata. In: Journal of Chinese Information Processing, pp. 110-115 (2009)
7. Bule Jiang.: Research on Rule-based the method of Mongolian automatic correction. Hohhot: Inner Mongolia University (2014)
8. GB 25914-2010: "information technology of traditional Mongolian nominal characters, presentation characters and control characters using the rules", 2011-01-10, publish (2011)
9. Chengqing Zong.: Natural Language Processing (Second Edition). Beijing: Tsinghua University press, 83-104 (2013)

10. Daniel Jurafsky and James Martin.: Speech and Language Processing, 2nd Ed. Prentice Hall, (2009)
11. James, Frankie.: Modified Kneser-Ney Smoothing of n-gram Models. In: RIACS Technical Report (2000)
12. Jianliang Meng, Junwei Zhang, Haoquan Zhao.: Overview of the Speech Recognition Technolog, In: International Conference on Computational & Information Sciences (2012)
13. Mikolov T, karafiat M, Burget L, et al.: Recurrent Neural Network Based Language Model In: Proc of 11th Annual Conference of the International Speech Communication Association. Makuhari, Japan, 1045-1048 (2010)
14. Mikolov T, Kombrink S, Burget L, et al.: Extensions of Recurrent Neural Network Language Model. In: Proc of the IEEE International Conference on Acoustics, Speech and Signal Processing. Prague, Czech Republic, pp. 5528-5531 (2011)
15. Forney Jr., G.D.: The Viterbi algorithm. Proc. In: IEEE 61(3), pp. 268-278 (1973)
16. Stolcke, A., et al.: SRILM --An Extensible Language Modeling Toolkit. In: Proc. Intl. Conf. on Spoken Language Processing, vol. 2, pp. 901-904, Denver (2002)
17. D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely.: The kaldi speech recognition toolkit. In: IEEE 2011 Workshop on Automatic Speech Recognition and Understanding. IEEE Signal Processing Society (2011)
18. Hui Zhang, Feilong Bao, Guanglai Gao.: Mongolian Speech Recognition Based on Deep Neural Networks. In: 2015 Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data (CCL), pp. 180-188 (2015)