Information Retrieval Oriented Mongolian Homograph Pronunciation Recognition

S. Loglo and Sarula
College of Mongolian Studies，Inner Mongolia University, Hohhot, 010021, China
E-mail:sloglo@sina.com

Abstract

Some Mongolian characters are spelt in the same way but pronounced differently. If only spelling was factored in word input, one word would have multiple input methods. One of the main challenges in Mongolian information retrieval is how to correct pronunciations and recognize the pronunciations of homographs. According to statistics that in the text that words have been inputted using Mongolian International Standard Codes, the mispronounced words account for an average of 40% of total number of words, up to 60% in some cases. Under such circumstances, we would be unable to get the information we need if mispronunciations were not corrected. The most difficult task in pronunciation correction is homograph pronunciation recognition. This paper solves this problem with a training method based on conversion and collocation. The accuracy of the pronunciation recognition can reach more than 84%. More importantly, we have designed and implemented an error correction tool with a human-computer interactive training mode, which allows us to continuously improve the recognition accuracy of the software.

Key Words

Mongolian; Information Retrieval; Homographs; Pronunciation Recognition

1．Introduction

1.1 Issues Encountered in Mongolian Text Retrieval

As the progress of global digitalization and application of internet, in China, relevant national authorities have taken timely measures to protect, develop and utilize ethnical language resources. These measures have helped to enrich the online Mongolian text resources. A rough statistics suggests that there are 500 Mongolian-related websites

encompassing such areas as politics, economics, science and technology, sports, military history, culture, religion, medicine and entertainment. These websites have enabled Mongolian language resources to be digitized, networked and shared, and ethical culture to be advocated. But effective Mongolian retrieval has not been achieved due to the following two reasons.

Firstly, there is an absence of uniform coding standard. International standard for Mongolian Coding has been approved for 15 years. But as software developers and websites stick to their own coding systems and none of them is strong enough to dominate the market, no uniform Mongolian coding has been established at this stage. To solve this problem, many businesses, public service units and research institutes have designed and realized relevant conversion applications and algorithm. But these applications and algorithms can only guarantee the accuracy of word forms, instead of the pronunciations of the homographs. For instance, Universal Algorithm for Mongolian Code Conversion (S. Loglo , 2009a) maps the word form ᠳ indiscriminately into the fourth vowel u. But such treatment can only meet the demand of printed publications.

Secondly, there is no standardized input format. Given the difference in people's mastery of Mongolian standard pronunciations and the difference in the habits of input, one code standard could result in a list of words in which character sequences are uncertain. Statistical analysis suggests that in text whose words have been inputted using International Standard for Mongolian Codes, words that are spelt correctly but mispronounced account for an average of 40% of total number of words, up to 60% in some cases. To solve this problem, we have worked out a traditional Mongolian pronunciation correction tool (S. Loglo , 2009b), but that also did not address how to recognize the pronunciations of homographs.

1.2 Relationship between Mongolian Pronunciation and Text Retrieval

There are usually two methods to retrieve Mongolian text, one based on the pronunciations of characters and the other based on glyph. In order to adopt the pronunciation-based method, it is necessary to ensure that all the characters that make up a word are accurate in pronunciation and shape. The glyph-based retrieval method allows us to ignore the pronunciations of all the characters but requires accuracy in each glyph.

In Mongolian, one glyph can correspond to multiple pronunciations, as shown in Table 1 (where only some glyphs are listed). As such, there may be several ways to input one word if only word forms are taken into account. For example, the word ᠤᠨ (edge or plain) may have such input methods as tal_a (correct pronunciation), tel_a, tal_e, tel_e, dal_a, del_a, dal_e, del_e.

Glyph	Pronunciation	Glyph	Pronunciation	Glyph	Pronunciation	Glyph	Pronunciation
ᠠ	a,e	ᠣ	o,u	ᠤ	t,d	ᠵ	gi,xi
ᠡ	a,e,n	ᠤ	o,u	ᠥ	t,d	ᠶ	x ö,xü,g ö,gü
ᠢ	a,e	ᠥ	o,u, ö,ü	ᠦ	e,w	ᠷ	x ö,xü,g ö,gü
ᠨ	a,e,n	ᠦ	ö,ü	ᠧ	ge,xe	ᠸ	x ö,xü,g ö,gü
ᠩ	i,y	ᠧ	ö,ü	ᠨ	ge,xe	ᠹ	x ö,xü,g ö,gü
ᠪ	i,y	ᠫ	a,x,g	ᠬ	gi,xi	……	……

Table 1: Correspondence between glyph and its pronunciation

When the pronunciation-based retrieval method is adopted, it is necessary to correct the errors in the text and key words beforehand since words with correct glyphs but wrong pronunciations are frequently seen in Mongolian Text. When such words are non-homographs, the pronunciation errors can be corrected using dictionary. When they are homographs, however, the pronunciation errors can only be recognized and corrected through context.

When the glyph-based retrieval method is employed, the text or key words must first be converted into glyph code sequences using coding conversion. But such a method cannot differentiate homographs, because glyph codes do not contain information for pronunciation but correspond to different glyphs. To illustrate, if users want to search for the word form "ᠮᠠᠯ", the system may produce results like "…… ᠮᠠᠯ (mal, animal)ᠪᠣᠭᠤᠴᠠ……" and "……ᠮᠡᠯ（mel, adverb）ᠵᠢᠷᠤᠭ……". But these are not what the users want, because they normally only search for "ᠮᠠᠯ" (mal, animal) or "mel ᠮᠠᠯ" (adverb) separately. In such circumstances, it is necessary to identify the pronunciations of the key words in the retrieved results and take out those that do not meet the requirements of retrieval.

Therefore, homograph pronunciation recognition and correction is the key technology in Mongolian text retrieval whether pronunciation-based or glyph-based retrieval method is adopted. It is worth noting, however, that such pronunciation recognition is different from most of the Grapheme-to-Phoneme (G2P) needs in voice recognition. The difference lies in that instead of converting graphme to phoneme, this technology maps each character of the retrieving word to its corresponding pronunciation, based on context.

In the field of information processing oriented Mongolian homograph, Zhang Jianmei studied pronunciation recognition conditions for 140 high frequency homographs (Zhang, 2007), Shu-Qin constructed a "Mongolian Homograph Knowledge Bank" (Shu, 2010) which summarized the types and grammatical attributes of Mongolian homographs, and

Hasi designed and implemented a management platform for “Mongolian Homograph Information Dictionary (Hasi and Nasun-Urt, 2010). But so far there is not a uniform pronunciation algorithm and software system for Mongolian homographs.

2. Mongolian Homographs

In Mongolian information processing, the first problem to solve in the study of Mongolian and applied linguistics is to recognize homographs and polysemous words. Without a solid research on recognition of homographs, understanding the natural language of Mongolian is impossible. Homographs are closely related to but are not identical with polysemous words and multiple-category words. Homograph is named from the perspective of spelling. Polysemous words are named from the angle of semantic and share the same pronunciation, spelling and etymology. Multiple-category words are named from the perspective of cross-category words (Jirannige, 2007). Since polysemous words and multiple-category words are not directly related to pronunciation retrieval, this paper only discusses and studies the pronunciation recognition approaches for homographs.

Homographs account for 7% of total words in electronic texts of traditional Mongolian. Structure wise, they are categorized into root homographs, root plus “root+suffix” homographs and “root+suffix” plus “root+suffix” homographs. In Mongolian Orthography Dictionary (Second Version), the first category has 1071 words, the second category contains 1577 words, and the third category contains an indefinite number of words. The first category and the secondary category of homographs are called static homographs hereunder. The third category of homographs is called dynamic homographs hereunder. Table 2 presents the statistics for the first two categories.

Number of Pronunciations	Two	Three	Four	Five
Number of Words	2447	176	23	2

Table 2: Statistics of Pronunciations of Mongolian Homographs

When establishing a vocabulary list, we treat the three categories of homographs using different strategies. For the first category, we extract a word from dictionary and compare it with other words in terms of spelling to get its homographs. For the second category, we adopt an Finite Automaton-based Recognition Algorithm (S. Loglo , 2009c) because “root+suffix”(Verb) forms are not available in the dictionary. Specifically, we extract a word from the vocabulary list and recognize it against the “root+suffix” form (Among the roots of the dictionary, there are only verb roots, and the suffixes are only in the form of supplementary elements to verb configuration). If recognition is successful, the current

word will be added to the homograph vocabulary list. We did not create an initial vocabulary list for the third category of homographs. Alternatively, we add them one by one to the vocabulary list when such words are encountered in proofreading a specific text. Specifically, in the process of proofreading, if the morphological changes of a verb correspond to more than one form of "root + affix (...)" and differ in pronunciations, we would treat them as homographs and add them to the homograph vocabulary list. Table 3 shows a sample of a homograph vocabulary list.

Spelling	Pronunciation Sequence
	ayimagta#N@0&Yes|ayima/gda#V@0&Yes
	ayimagtan#N@0&Yes|ayima/gda/n#V@0&Yes
	anggi#N@0&Yes|angxi#V@0&Yes
	anggin_a#N@0&Yes|angxi/n_a#V@0&Yes
	abiyasuraxu#V@0&Yes|abiyasurxaw#A@0&Yes
	aburaltan#N@0&Yes|aburalda/n#V@0&Yes
	aburaltu#A@0&Yes|abura/ldu#V@0&Yes
...	...

Table 3: Homograph Vocabulary List Sample

Please note that we use “|” to separate different pronunciations in the pronunciation sequence. The character between “#” and “@” behind each pronunciation denotes part-of-speech, the character between“@” and “&” is the frequency and the character between “&”and “|” indicates whether the current pronunciation is valid.

3. Pronunciation Recognition Algorithm for Homographs

1,557 of the 2,648 static homographs in traditional Mongolian have pronunciations corresponding to different parts-of-speech. As such their pronunciation recognition is a de facto process of multiple-category word disambiguation. The remaining 1,081 static homographs and all dynamic homographs have pronunciations that are wholly or partially corresponding to the same parts-of-speech. Recognizing these pronunciations is the most difficult. The following presents two approaches to recognize pronunciations for these two categories of homographs.

3.1 Pronunciation Recognition for Homographs with Different Parts-of-speech

Such homographs account for 58.8% of dynamic homographs. They can be easily recognized by the parts-of-speech information of their neighboring words. Of the 1,557

such homographs, 1,034 have noun part-of-speech. Nouns have rich morphological forms, and their parts-of-speech can be easily judged using morphological rules. The training and testing corpora chosen for this paper already has detailed parts-of-speech information. Due to these reasons, we adopt the conversion-based and error-driven pronunciation recognition algorithm (Hearst, 1991; Eric, 1995).

In fact, the pronunciation recognition for homographs with different parts-of-speech is a part-of-speech tagging process. After correctly tagging the parts-of-speech, the pronunciation can be determined easily. For example: Mongolian homograph "ᠮᠠᠯ" has two different parts-of-speech “N” (Noun) or “D” (adverb), when the parts-of-speech is "N", the corresponding pronunciation is "mal", when the parts-of-speech is "D", the corresponding pronunciation is "mel". If we using the conversion-based and error-driven method to tagging the parts-of-speech, a tagging set, a POS tagged corpus and an initial tagger are needed. In this paper, we used national standard "Information Technology-Mongolian Word and Expression Marks for Information Processing" (GBT 26235-2010) as tagging set, 1 million words "modern Mongolian corpus" as POS tagged corpus and Mongolian lexical analyzing system “Lexical” for initial tagger. Rule template used in the algorithm as follows:

Rewrite rules: rewrite the parts-of-speech *x* to *y*

Activation environment:

(1) If use w_i to represent the current word, the parts-of-speech for w_{i-1} or w_{i+1} is *z*

(2) If use w_i to represent the current word, the parts-of-speech for w_{i-2} or w_{i+2} is *z*

(3) The morphological form for current word is *z*

(4) The morphological form for previous word is *z*

The main steps of conversion-based and error-driven methods are:

Step1: C_0= 1 million words "modern Mongolian corpus" (POS tagged)

Step2: Use the initial tagger *Lexical* to handle C_{0-raw}(non POS tagged edition of 1 million words "modern Mongolian corpus”) and get POS tagged corpus C_1. Through comparison of C_0 andC_1, get a series of conversion rules T_j(j=0, 1, 2, …, n).

Step3: Use conversion rules T_j(j=0, 1, 2, …, n) to handle C_i(i> 0) and get POS tagged corpus $C_i^{T_j}$(j=0, 1, 2, …, n)

Step4: form $C_i^{T_j}$ (j=0, 1, 2, …, n) choose a corpus C_i^{top}that has the highest annotation accuracy. And then, save the corresponding conversion ruleT_j.

Step5: if accuracy of C_i^{top} > accuracy of C_i， go to step6，else goes to step7.

Step6: $C_{i+1} = C_i^{top}$; $i = i + 1$; go to step3.

Step7: end

Through the above machine learning process, we will get the conversion rule set T_k(k=0, 1, 2, …, m).

3.2 Pronunciation Recognition for Homographs Having the Same Part-of-speech

Homographs of this type are often recognized using collocation. When two words' syntactic functions are identical or similar in a text, their morphological features and parts-of-speech information can no longer disambiguate. Instead words before or after these words or words at a greater distance may disambiguate because ambiguity seldom arises on lexical level. For example,  has two pronunciations, (bodoju, thinking) and (buduju, coloring), but their parts-of-speech are both transitive verbs with the same syntactic functions. As such the two pronunciations can only be seperated using their collocation with words before and after it. Collocation can be described by collocation vocabulary list or collocation rules. But this description method would need people to manually or semimanually search phrases and collocation rules. This means a high workload, and an algorithm whose sophistication and robustness are not good. For these reasons, we adopt a collocation probability model. Forumula (1) shows this model.

Statistical pronunciation recognition model turns disambiguation into a process of optimization. In other words, this model calculates the probability for each possible pronunciation, and finds the most probable pronunciation *d** from context. This process is such that:

$$d^* = \text{argmax}P(d|f) = \text{argmax}\, P_{d\in D}(d|f) \quad (1)$$

where $\sum_{d\in D} P(d|f) = 1$, *d** denotes the most probable pronunciation, *d* is a production for word form *f*, and *D* is all the pronunciations of morph *f*. To reduce data sparseness of the lexical information in the training corpus, in calculating the probabilities of pronunciations we have included the word before and after the current word, as well as the semantic classification of the second word before the current word and the word before and after the current word.

$$P(d_i) = P(d_i|W, Dist) \quad (2)$$

where d_i denotes the i^{th} pronunciation of word form *f*, *W* is the root or punctuation

symbol in collocation with d_i, and Dist is the linear distance between *W* and (for example, Dist==-1 indicates that *W* is the first word before d_i). This formula is used to calculate the frequency of collocations between Word *W* and the pronunciation d_i of word form *f*.

$$P(d_i) = P(d_i|S_w, Dist) \tag{3}$$

where S_w denotes collocation word *W*'s semantic classification information. The other symbols in this function are the same as those in the above function.

$$P(d_i) = P(d_i|f) \tag{4}$$

This function is used to calculate the probability of word form *f* given d_i.

In actual calculation, Dist can be -1，-2 and 1. Since the semantic category information the second word in front of the current word is not counted, only 6 functions are needed.

$$\begin{aligned}
P_1(d_i) &= \frac{Count(d_i,w,Dist=-1)}{Count(f)} \\
P_2(d_i) &= \frac{Count(d_i,w,Dist=1)}{Count(f)} \\
P_3(d_i) &= \frac{Count(d_i,w,Dist=-2)}{Count(f)} \\
P_4(d_i) &= \frac{Count(d_i,S_w,Dist=-1)}{Count(f)} \\
P_5(d_i) &= \frac{Count(d_i,S_w,Dist=1)}{Count(f)} \\
P_6(d_i) &= \frac{Count(d_i,)}{Count(f)}
\end{aligned} \tag{5}$$

To improve the effectiveness of the algorithm, the counting results of training corpus are stored in the Finite Automaton-based dictionary files. The recognition route in the computer includes the numerators of the six functions (their denominators are not included because they are all the same). The strings of numerators are recognized to obtain the probability value that stored in the terminal state.

When doing specific recognition, we find the above seven items by looking through each pronunciation of each homograph and then obtain the probability value *P* through smooth interpolation.

$$P = \lambda_1 p_1 + \lambda_2 p_2 + \lambda_3 p_3 + \lambda_4 p_4 + \lambda_5 p_5 + \lambda_6 p_6 \quad (6)$$

where $\lambda_1 = \lambda_2 = 0.4; \lambda_3 = 0.05;\ \lambda_4 = \lambda_5 = 0.15;\ \lambda_6 = 0.05$. The initial values of the parameters are set according to experience and are fine-tuned according to recognition accuracy in pronunciation recognition experiment.

3.3 Realization of Pronunciation Recognition Algorithm for Homographs

We have designed and realized Mongolian Pronunciation Proofreading and Error Correction Software System Editor using "dictionary+rules" and a conversion-and-collocation-based learning method. The designation and realization of this Editor is based on International Standard for Mongolian Codes, China's National Standard for Traditional Mongolian Word Form Norms (consistent with Mongolian Orthographic Dictionary) and Mongolian morphological and word-formation rules. Figure 1 shows its text processing and training process. This paper describes the automatic recognition process for homographs in the Editor, as shown in the lower right dashed box below. For details on pronunciation correction, please refer to references (S. Loglo , 2009b and 2009c).

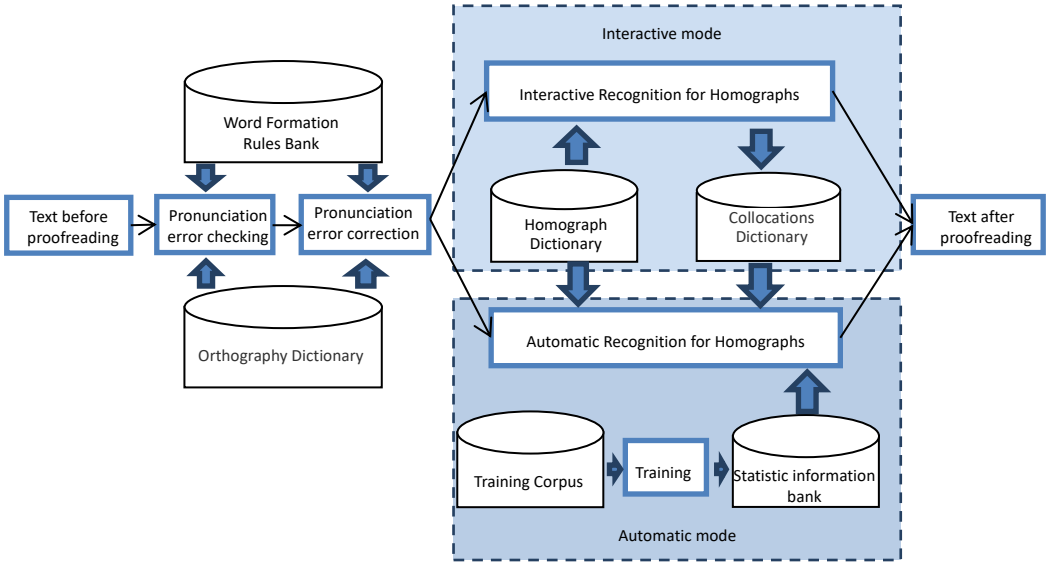
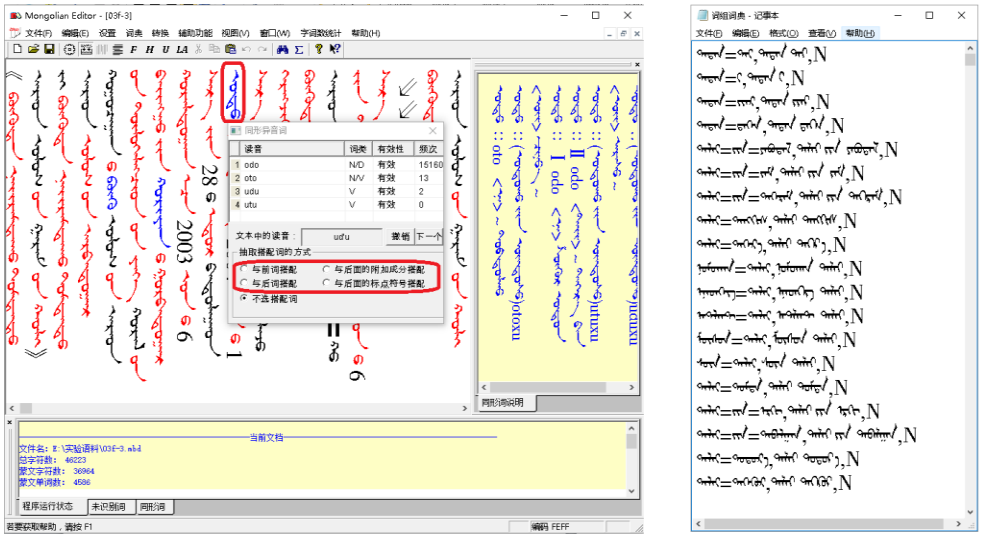


Figure 1: Flow Chart of Pronunciation Error Correction and Homograph Pronunciation Process

Dictionary+morphological rules and word formation rules are over 99.90% accurate in

recognizing and auto-correcting pronunciations of non-homographs, but are ineffective for homographs. This is because pronunciations of homographs rely on context. As such we treat homographs using human-computer interaction mode or full automation mode. In the process of text proofreading using human-computer interaction, we build a corpus for collocations by manually selecting correct pronunciations and extracting collocations. This corpus can also be used to judge pronunciation through the full automation mode.



(a) (b)

Figure 2: Editor's homograph processing based on human-computer interaction

Note: Figure 2(a) is the user interface for Mongolian Pronunciation Proofreading and Error Correction Software. After text is processed at the first stage (the left part of dashed box in Figure 1), the homograph is highlighted in blue. The box in the middle of Figure 2(a) is human-computer interactive pronunciation selection tool. The word within the red box above the tool is a homograph whose alternative pronunciations "oto,odo,utu,udu" are listed in the tool window. The red box below the tool window contains collocation options. Four radio buttons respectively indicate the current word collocate with previous word, next word, next suffix and next punctuation. If any of the keys is pressed, a collocation word is extracted into the corpus for collocations while correct pronunciation is chosen. Figure 2(b) shows such corpus.

The statistics-based full automation can recognize and correct pronunciations for homographs after training the linguistics models through the algorithm introduced in section 3.1 and section 3.2.

Users can build a corpus for collocations using Editor's human-computer interaction mode or automatically train the recognition algorithm in the full automation mode.

4. Experiment and Analysis

Language Research Institute of Inner Mongolia University constructed a 1-million-word modern Mongolian corpus in a span of eight years from 1984 to 1991 and expanded it twice into what is now a 10-million-word corpus. But the expanded corpus still hasn't solved the mispronunciations for homographs, and thereby cannot be used to train the linguistics models proposed in training and test text. We adopt for this paper the earliest and the most sophisticated version of the 1-million-word corpus which contains materials from novels (19.6%), textbooks (50.3%), newspapers and periodicals (9.8%) and politics (22.9%). So far this corpus has undergone compound tagging, part-of-speech (POS) tagging and roots and affix segmentation.

Given the uneven distribution of homographs across different texts, we extract at a ratio of 4:1 the four categories of texts from the 1-million Mongolian corpus, and build a 800,000-word training set and a 200,000-word test set.

After training and testing, accuracy rates for homographs with different POS and homographs with the same POS have hit 87.2% and 81.6% respectively, averaging 84.88%. The human-computer interaction mode as shown in Figure 1 can improve the accuracy of pronunciation recognition algorithm by building a corpus for collocations. But since the corpus is small, this practice is not adopted in text testing.

As regards the overall performance of pronunciation proofreading and correction, the system has achieved a more than 99% recognition rate for words, with unrecognized words being mainly loan words and proper nouns. Among the recognized words which exclude homographs, pronunciation correction results are fully correct. The correction experiment shows that homographs account for about 7% of Mongolian texts. In view of the above factors, the Editor's pronunciation correction rate has hit 98.01%. This figure is given by the following function.

Accuracy rate for pronunciation correction =word recognition rate (1-proportion of homographs) ×accuracy rate for non-homographs + proportion of homographs ×accuracy rate for homograph pronunciation recognition.

Since the training corpus is relatively small, some low-frequency homographs did not appear in the training corpus. Neither did some special pronunciations of high-frequency homographs. This phenomenon did not have a big impact on the algorithm that was driven by conversion and error of POS information, but resulted in spare data in algorithm that was mainly about collocation relationships.

Supplementary elements of nouns play an important role in the pronunciation recognition of homographs with different POS. When verbs and static words have the same spellings, they simultaneously mainly appear in the form of stem. Verbs that appear in the form of stem play the role of revelation and command, and thus their punctuations marks after these verbs are most likely to be recognized. Integrating the above features into the activated environment of conversion rules would greatly improve the system's ability to correct errors.

5. Acknowledgement

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 13BXW047).

6. References

S. Loglo . Universal Algorithm for Mongolian Code Conversion, Inner Mongolian University Journal (Philosophy and Social Sciences Edition), 2009(2), 133-136.

S. Loglo . Mongolian Pronunciation Error Correction Software Universal Mongolian Code Conversion, Inner Mongolian University Journal (Philosophy and Social Sciences Mongolian Edition), 2009(1), 59-64

Zhang Jianmei. Information Processing-oriented Mongolian Homograph Pronunciation Recognition, Inner Mongolian University Journal (Humanity and Social Sciences Edition), 2007(5),25-28.

Shuqin. Construction of a Mongolian Homograph Knowledge Bank, Inner Mongolian University Degree Thesis, 2010, Huhehaote

Hasi and Nasun-Urt. The Design and Application of Mongolian Homograph Words Information Dictionary[C]，Proceedings of the Fifth International Conference on Intelligent Networks and Intelligent Systems, Tianjin, 2012。

Jirannige. Mongolian Homograph's Statistics, Proceedings of 11th National Ethnic Language Character Information Academic Workshop[C], Xishuangbanna, February 2007, 229-231

S. Loglo . Mongolian Proofreading Algorithm based on Nondeterministic Finite Automaton [J], Journal of Chinese Information Processing, 2009(6), 110-115.

Hearst, M. Noun Homograph Disambiguation using Local Context in Large Text Corpora[C], Proceedings of the 7th Annual Conference of the UW Centre for the New OED and Text Research: Using Corpora, Oxford, 1991.

Eric Brill. Transformation-Based Error-Driven Learning and Natural Language Processing:

A Case Study in Part-of-Speech Tagging [J], Computational Linguistics, 21(4), 543—465.

David Yarowsky. Homograph Disambiguation in Text-to-Speech Synthesis[C], Progress in Speech Synthesis, 1997, 157-172.

Wei Naixing. Corpus-based and Corpus-Driven Collocations, Modern Linguistics, 2002(2), 101-114

Zhang Yangsen. Error Correction Knowledge Bank Building in Chinese Proofreading System and Algorithm for Generation of Error Correction Suggestions, Journal of Chinese Information Processing, 2001(5), 33-39