Myanmar Homonym Disambiguation System

Zar Zar Hlaing, Aye Thida
University of Computer Studies, Mandalay
zarzarhlaingucsm@gmail.com, ayethida.royal@ucsm.edu.mm

Abstract

Natural Language Processing (NLP) is one of the most important research areas in Human Language. One ofthe challenges in Natural Language Processing (NLP) is to resolve ambiguous homonyms or homonym errors in sentences. Myanmar Homonym Disambiguation System is the kind of Word Sense Disambiguation System in Natural Language Processing. This system is needed for Myanmar Word Segmentation and Spell Checker System. If the sentence contains incorrect homonyms, this sentence cannot be segmented correctly. Moreover, incorrect usage ofhomonyms is a common problem in Myanmar to English translation. In this paper, Myanmar Homonym Disambiguation System has been described. This system detects homonym errors or ambiguous homonyms and then resolves these errors by using Corpus-Based N-Gram Model. Myanmar Text Corpus is also needed in calculation of N-Gram Model for this system. After resolving homonym errors, the system will output the sentence with correct homonyms.

Keywords**:** *NLP, Homonyms, Corpus-Based N-Gram Model, Myanmar Text Corpus*

1. Introduction

Most Researchers have been researched for spelling checker systems in many languages. However, Myanmar Homonym Disambiguation System or Spelling Checker System for Myanmar homonyms is still in its infancy. Solving Myanmar homonyms can support in the research area of Natural Language Processing. Incorrect homonyms are ambiguous for poor reader. The common reasons for misusing homonyms caused homonym errors. All homonym errors can be context errors, but all context errors cannot be homonym errors. Myanmar Homonym Disambiguation System can solve these homonym errors. Myanmar Homonym Disambiguation System is one of the most vital roles in NLP tasks. This system is used to increase the success rate of Natural Language Processing (NLP) applications. This system is defined as the task of using the correct homonym word in specific context. Human language translation is a difficult task in natural language because there has language ambiguity.

The remaining parts of this paper are organized as follows. In section 2, Myanmar Language is described. The related work is explained in section 3. Moreover, Homonym description is displayed in section 4. And also, Corpus and Corpus Based N-Gram Model are described in section 5 and section 6. Then, the overview of the proposed system and step by step processing of Myanmar Homonym Disambiguation System are shown in section 7. In section 8, Experimental Results are discussed. Conclusion is described in the last section 9.

2. Myanmar Language

Myanmar language is the official language of the Republic of the Union of Myanmar. MLC (Myanmar Language Commission) standardized that it is made of 9 parts of speech (noun, pronoun, adjective, verb, adverb, post-positional marker, particle, conjunction and interjection) in Myanmar Grammar. Myanmar Language is written from left to right with no spaces between words. But informal writing form often contains space after each word. Myanmar Language has end of the sentence with boundary marker. It follows the SOV (subject, object and verb) order and it is a free word order. It is hard to tokenize because it cannot tokenize by space. Word tokenization and segmentation are the essential parts in the field of Natural Language Processing (NLP). In the proposed system, tokenization of Myanmar texts is also used.

3. Related Work

Many approaches have been applied for Homonym Disambiguation. Most of the studies have been researched to English and other languages. There is no researcher who researched to Myanmar Homonym Disambiguation System.

In the paper written by Hendrik J. Groenewald and Marissa van Rooyen, frequency-based approach and tree-based approach are used for Afrikaans Homophone Disambiguation. To check homophones that are used in the wrong context, the texts in a document are needed to check. Sentencised document is used in this process. It continues on a per-sentence basis. The sentence in the sentencised document is then checked to decide if it contains a homophone word. If it doesn't contain, the next sentence in this sentencised document is also checked. If a homophone word is contained in a sentence, the sentence is applied to the summation module. This summation module is used to do comparison of between the words in the sentence and the constraint words of the involved homophone. The normalized frequency value of all the constraint words that are found in the sentence is summed. If sum is above a certain threshold value, the homophone is flagged and the other homophone in the pair is suggested **[1]**.

Jun-Su Kim, Wang-Woo Lee, Chang-Hwan Kim and Cheol-young Ock have been developed A Korean Homonym Disambiguation System based on Statistical Model to which Bayes' theorem is used, and suggested a model that established the weight of sense rate and the weight of distance to the adjacent words to increase the accuracy **[2]**.

There are many approaches that have been applied to Homonym Disambiguation Systems. Moreover, Text Corpus must also be used for these systems. For Myanmar Homonym Disambiguation System, Corpus Based N-Gram Model is more suitable than the other approaches because Corpus-Based N-Gram model is a type of probabilistic language model to predict the next item in a given sequence.

4. Homonym

Homonym is a word that pronounces the same as another word but these words have different meanings, where spelling is same or not. A more restrictive definition sees homonyms as words that are simultaneously homographs and homophones. The relationship between a set of homonyms is called homonymy. The term "homonym" may be used to refer to words that are either homographs or homophones.

Table 1. Example of Myanmar Homonyms

1	ကျိုး	ကြိုး
2	ခမ်း	ခန်း
3	ကျား	ကြား
4	စီ	စည်
5	တမ်း	တန်း

Explanation of above Myanmar homonyms:

1. **ကျိုး** - တံတား**ကျိုး**၊ လမ်း**ကျိုး**၊ခဲတံ**ကျိုး**၊သစ်ကိုင်း**ကျိုး** etc.
   ကြိုး-**ကြိုး** (အရာဝတ္ထုများကိုချည်ရန်)
2. **ခမ်း** - အ**ခမ်း**အနား
   ခန်း - တိုက်ခန်း၊စာသင်ခန်း၊ စာကြည့့်**ခန်း** etc.
3. **ကျား** - **ကျား**(သတ္တဝါ)**ကြား** - အသံကြား၊သီချင်းသံကြား etc.
4. **စီ** - စာအုပ်စီ၊တန်းစီ၊စီစီရီရီ etc.
   စည် - စည်ကား၊ စည်စည် ကားကား etc.
5. **တမ်း** - **တမ်း**တ
   တန်း - စီတန်း၊အတန်းလိုက် etc.

The words (**two**, **to**, **too**) are pronounced identically for the sentence "The **two** boys want **to** play **too**.", but spellings of these words are different. These three words are homonyms. There are many pairs of homonyms in the English language. Some examples of English homonyms are described as follows:

Table 2. Example of English Homonyms

1	**Accept**	**Except**
2	**Aid**	**Aide**
3	**Complement**	**Compliment**
4	**Discrete**	**Discreet**
5	**Forth**	**Fourth**

Explanation of above English homonyms:

1. **Accept** - to take something that is given to you
   Except - to leave out
2. **Aid** - help
   Aide - one who helps
3. **Complement** - something that makes a thing whole or perfect
   Compliment - to praise
4. **Discrete** - separate
   Discreet - modest, prudent, unobtrusive
5. **Forth** - forward
   Fourth - an ordinal number

5. Corpus

In linguistics, a corpus (plural *corpora*) or text corpus is a large and structured set of texts (nowadays usually electronically stored and processed). They are used to do statistical analysis and hypothesis testing, checking occurrences or validating linguistic rules within a specific language territory. Texts in a single language (monolingual corpus) or text data in multiple languages (multilingual corpus) may be contained in a corpus **[4]**. Monolingual corpus is used in this system.

Monolingual corpus is the most frequent type of corpus. Only texts in one language can be contained in monolingual corpus. This monolingual corpus is usually tagged for parts of speech and is used by many users for various tasks from highly practical ones, e.g. checking the correct usage of a word or looking up the most natural word combinations, to scientific use, e.g. identifying frequent patterns or new trends in language.

The corpus may contain written language, spoken language or both. Spoken corpus consists of audio recordings. A corpus may be open or closed. An open corpus that does not claim to contain all data from a specific area but a closed corpus claims to contain all or approximately all data from a particular field. For example, Historical corpora are closed because there can be no further input data. Grammarians, lexicographers, and other interested parties with better descriptions of a language are provided by a corpus. Computer-processable corpora allow linguists to adopt the principle of total accountability, retrieving all the occurrences of a particular word or structure for inspection or randomly selected samples. For the development of NLP tools, corpora are also needed to use. Corpus is the essential part of NLP applications spell-checking, grammar-checking, speech recognition and so on.

Either corpus-based approach or rules-based approach is used in most of the researches in linguistics. Corpus-based approach is also applied in this system. So, Myanmar Text Corpus is needed to use. This corpus is created manually. The data for this corpus is collected from magazines and news such as sports news, health news, political news, education news. Total number of sentences in this corpus is 37708. Maximum length of sentence is 46 and minimum length of sentence is 7. There are 138,442 tokens in this corpus. The larger size of corpus is used, the more accurate result is obtained in this system.

6. Corpus Based N-Gram Model

This N-Gram Model is a model which is used to calculate probability of character sequence that occurs as a word or probability of word sequence that occurs as a sentence. Probability of character can be estimated from source of data. N-Gram sizes are varied depending on how large programmer would set. It can be from 1 to (n). In this N-Gram model, the length of characters and word sequences are different (2-3 Gram and 4 Gram) **[3]**.

The number n in ***N-gram helps*** in naming the method. If a single word is used then it is known as a ***unigram***. If two word sequences are used then it is called ***bigram,*** if three word sequences are used then it is called a ***trigram*** *and so on*. ***N-gram*** model relates to data in a corpus and the performance of this model also depends on the sized of corpus. The higher the value of n, the more accurate result is obtained.

For sequences of words “the girl is beautiful”, the trigrams are “#the girl”, “the girl is”, “girl is beautiful”, and “is beautiful #”.N- gram model is also used in probability, commutation theory, computational linguistic, computational biology, and data compression. In natural language processing task and text mining, n-grams of texts are widely used. They are basically a set of co-occurring words within a given window. When computing the n-grams, word can be moved one word forward. Bigrams, trigrams, four grams, five grams and six grams calculations are used in this system. For the sentence “စည်းကမ်းကိုလိုက်နာပါ”,

If N=2 (known as bigrams), then the n grams would be:

- စည်းကမ်း
- ကမ်းကို
- ကိုလိုက်
- လိုက်နာ
- နာပါ

The formula of Bigrams Model is as follows:

$$P(W_{\text{n}}|W_{\text{n-1}}) = \frac{\text{C}(W_{\text{n-1}}\,W_{\text{n}})}{\text{C}(W_{\text{n-1}})} \qquad (1)$$

If N=3(known as trigrams), the n-grams would be:

- စည်းကမ်းကို
- ကမ်းကိုလိုက်
- ကိုလိုက်နာ
- လိုက်နာပါ

The formula of Trigram Model is as follows:

$$P(W_{\text{n}}|W_{\text{n-2}}W_{\text{n-1}}) = \frac{\text{C}(W_{\text{n-2}}W_{\text{n-1}}\,W_{\text{n}})}{\text{C}(W_{\text{n-2}}W_{\text{n-1}})} \qquad (2)$$

If N=4(known as four-grams), the n-grams would be:

- စည်းကမ်းကိုလိုက်
- ကမ်းကိုလိုက်နာ

- ကိုလိုက်နာပါ

The formula of Four-Grams Model is as follows:

$$P(W_n|W_{n-3}W_{n-2}W_{n-1}) = \frac{C(W_{n-3}W_{n-2}W_{n-1}\ W_n)}{C(W_{n-3}W_{n-2}W_{n-1})} \qquad (3)$$

If N=5(known as five-grams), the n-grams would be:

- စည်းကမ်းကိုလိုက်နာ
- ကမ်းကိုလိုက်နာပါ

The formula of Five-Grams Model is as follows:

$$P(W_n|W_{n-4}W_{n-3}W_{n-2}W_{n-1}) = \frac{C(W_{n-4}W_{n-3}W_{n-2}W_{n-1}\ W_n)}{C(W_{n-4}W_{n-3}W_{n-2}W_{n-1})} \quad (4)$$

If N=6(known as six-grams), the n-grams would be:

- စည်းကမ်းကိုလိုက်နာပါ

The formula of Six-Grams Model is as follows:

$$P(W_n|W_{n-5}W_{n-4}W_{n-3}W_{n-2}W_{n-1}) = \frac{C(W_{n-5}W_{n-4}W_{n-3}W_{n-2}W_{n-1}\ W_n)}{C(W_{n-5}W_{n-4}W_{n-3}W_{n-2}W_{n-1})} \qquad (5)$$

7. Overview of the Proposed System

This system is intended to resolve ambiguous homonyms in Myanmar text. Myanmar sentence is only accepted as an inputted sentence by this system. The system extracts ambiguous homonyms in inputted sentence and then resolves these homonyms. Corpus Based N-Gram Model is used to resolve these ambiguous homonyms in sentence. After resolving all detected ambiguous homonyms, the system will output the sentence with correct homonyms. The overall architecture of the system is shown in Figure1.

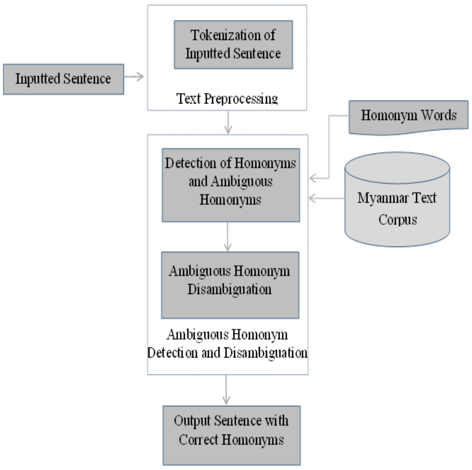
Inputted Sentence
Tokenization of Inputted Sentence
Text Preprocessing
Homonym Words
Detection of Homonyms and Ambiguous Homonyms
Myanmar Text Corpus
Ambiguous Homonym Disambiguation
Ambiguous Homonym Detection and Disambiguation
Output Sentence with Correct Homonyms

Figure 1. Proposed System Design

7.1. Preprocessing Step

This section describes the preprocessing step of the system. Tokenization of inputted sentence acts as a preprocessing step for the proposed system. It is useful in word sense disambiguation (WSD) of Natural Language Processing. Inputted text is tokenized according to the paper **[5]**. In English language, words are delimited by spaces. But, tokenization is hard to tokenize the texts for the languages (Myanmar, Thai, Chinese and Japanese languages). If the inputted sentence is "ကျောင်းစည်းကန်းများကိုလိုက်နာပါ။", the tokenized inputted sentence for this system is "ကျောင်း, စည်း,ကန်း, များ,ကို, လိုက်,နာ,ပါ ,။,".

7.2. Homonyms Detection

Detecting homonyms and ambiguous homonyms acts as the second step of this system. The system detects homonyms and ambiguous homonyms according to the following pseudo codes.

```
Begin
1. Take tokenized text;
2. Convert this tokenized text to string array tokens[];
3. Read the text file containing homonym words and convert to string array textWords[];
4. Check each value of array tokens[] is homonym word or not;
            if (each value of array tokens[] equals values of array textWords[] )
  this value is homonym word and concatenates this homonym word to the String tempHomonyms;
5. Homonym words in tempHomonyms String are solved by using Corpus Based  N-Gram Model to get
   ambiguous homonyms;
   If the system finds the ambiguous homonyms then print these ambiguous homonyms else print
   homonyms else print the original text;
End
```

Figure 2. Homonyms and Ambiguous Homonyms Detection Algorithm

Detecting homonyms and ambiguous homonyms acts as the second step of this system. The above pseudo code is used to detect the homonyms and ambiguous homonyms in the inputted sentence. For instance, if the tokenized inputted sentence is "ကျောင်း,စည်း,ကန်း, များ, ကို,လိုက်,နှာ,ပ◌ါ,။,", the system finds the homonym words "ကျောင်း, စည်း, ကန်း, များ, ကို, နှာ" for this inputted sentence.

7.3. Ambiguous Homonyms Detection

Homonyms in inputted sentence are solved to get the ambiguous homonyms by using Corpus Based N-Gram Model. Six homonym words are detected in the tokenized inputted sentence. The detected homonym words are "ကျောင်း, စည်း, ကန်း, များ, ကို, နှာ". The system solves the detected homonym words from left to right. Firstly, the system solves the first detected homonym word "ကျောင်း". Another homonym word of "ကျောင်း" is "ကြောင်း". The probabilities of each homonym word are calculated in this system. After calculating these probabilities, the probability of "ကျောင်း" is 0.01399 and the probability of another homonym word "ကြောင်း" is 0.0. The system takes the largest probability of homonym word "ကျောင်း". This homonym word and the first detected homonym word in inputted sentence are the same. So, the inputted sentence is "ကျောင်းစည်းကန်းများကိုလိုက်နှာပါ။" because there is no need to replace the first detected homonym word in the inputted sentence.

The second detected homonym word "စည်း" is also solved as the solution of the first homonym word "ကျောင်း". Another homonym words of "စည်း" are "စည်း, ဆီး, ဆည်း".The probability of homonym word "စီး" is 0.0 , the probability of homonym word "စည်း" is 0.5, the probability of homonym word "ဆီး" is 0.0 and the probability of homonym word "ဆည်း" is 0.0. The system takes the largest probability of homonym word "စည်း". After solving second detected homonym word, the inputted sentence is "ကျောင်းစည်းကန်းများကိုလိုက်နှာပါ။".

The remaining homonym words "ကန်း, များ, ကို, နှာ" are also solved as the solution of first and second detected homonym words. After solving these remaining homonym words, the inputted sentence becomes "ကျောင်းစည်းကမ်း များကိုလိုက်နှာပါ။". After solving all detected homonym words, this system check homonym words are equal or not in the original inputted sentence and the sentence obtained by solving all detected homonym words. If each homonym word in these sentences is equal, the system outputs the original sentence. Otherwise, this homonym word is needed to disambiguate. For example sentence "ကျောင်းစည်းကန်းများကိုလိုက် နှာပါ။", the system checks homonym words are equal or not in the original inputted sentence "ကျောင်းစည်းကန်းများကိုလိုက်နှာပ◌ါ" and the sentence "ကျောင်းစည်းကမ်းများကိုလိုက်နှာပါ။" obtained by solving all detected homonyms.

After checking each homonym word, the homonym word "ကန်း" in the original inputted sentence and the homonym word "ကမ်း" in the sentence obtained by solving all detected homonyms is not equal. Therefore, the system outputs the homonym word "ကန်း" as the ambiguous homonym word.

7.4. Homonym Words Disambiguation

This section describes homonym words disambiguation step of the system by using Corpus Based N-Gram Model. The example inputted sentence is "ကျောင်းစည်းကန်းများကိုလိုက်နှာပါ။". First step, the system tokenizes the inputted sentence. After tokenizing the inputted sentence, the tokenized sentence becomes "ကျောင်း, စည်း, ကန်း, များ, ကို, လိုက်, နှာ, ပါ , ။,". Second step, the system detects the homonyms and ambiguous homonyms by using *Homonyms and Ambiguous Homonyms Detection Algorithm.* After doing this second step, the system finds the ambiguous homonym "ကန်း".

Next step, this ambiguous homonym word "ကန်း" is to be solved. Another homonym word of "ကန်း" is "ကမ်း". After calculating probabilities of these two homonym words, the probability of homonym word "ကန်း" is 0.0 and the probability of homonym word "ကမ်း" is 0.5. The system takes the largest probability of homonym word "ကမ်း" and replaces it in the inputted sentence. After solving this homonym word, the inputted sentence becomes "ကျောင်းစည်းကမ်းများကိုလိုက်နှာပါ။". Finally, the system outputs the sentence "ကျောင်းစည်းကမ်းများကိုလိုက်နှာပါ။" with the correct homonym word "ကမ်း".

8. Experimental Results

The accuracy of detecting ambiguous homonyms and disambiguating the detected ambiguous homonyms in the inputted sentence is calculated by using three measures of assessments:

precision, recall and f-measure. 1000 inputted sentences are used as testing data. The formulas of precision, recall and f-measure for the experimental result of detecting and disambiguating the ambiguous homonyms are:

$$\text{Precision}=\frac{\text{Total True Positive}}{\text{Total True Positive}+\text{Total False Positive}} \quad (6)$$

$$\text{Recall} = \frac{\text{Total True Positive}}{\text{Total True Positive}+\text{Total False Negative}} \quad (7)$$

$$\text{F-Measure} = \frac{2(\text{Precision} \times \text{Recall})}{(\text{Precision} + \text{Recall})} \quad (8)$$

Where,

Precision = percent correctly detected or disambiguated ambiguous homonyms

Recall = coverage of correctly detected or ambiguous homonyms

F-Measure = the ratio between recall and precision

The overall performance of the system is calculated by an expert user. As a result, it can be concluded that the system provides 97.73% of precision, 97.27% of recall and 97.50% of f-measure for detecting ambiguous homonyms in the inputted sentences and 94.90% of precision, 100% of recall and 97.38% of f-measure for disambiguating the detected ambiguous homonyms.

9. Conclusion

Using incorrect homonyms is a major problem in Myanmar to English Translation. Myanmar words segmentation can also be incorrect because of incorrect usage of homonyms. Moreover, consequences of Myanmar words segmentation in NLP applications can be caused. Homonym errors are needed to solve in most of the spelling checker systems. If Homonym errors can be solved, the performance of spell checker will improve. Therefore, for the above reasons, Myanmar Homonym Disambiguation System is an essential role in Natural Language Processing. This Myanmar Homonym Disambiguation System can be applied in Myanmar NLP applications. The main limitation of the system is that it can accept either Myanmar-3 Unicode format or Zawgyi-One format. Stacked consonant homonyms such as "ဝိဇ္ဇာ, ပဏ္ဍိ, ကဏ္ဍ, တက္က, အဂ္ဂ, သိပ္ပံ, သဏ္ဍာန်, etc." can't be resolved in this system.

References

[1] Hendrik J. Groenewald and Marissa van Rooyen, "Afrikaans Homophone Disambiguation".

[2] Jun-Su Kim, Wang-Woo Lee, Chang-Hwan Kim, Cheol-young Ock, Dept. of Computer Engineering & Information Technology, University of Ulsan, "A Korean Homonym Disambiguation System Based on Statistical Model Using weights ", San29, Mugeo-dong, Nam-gu Ulsan, Korea, pages 680-749.

[3] "N-Gram Wikipidea", https://en.wikipedia.org/wiki/N-gram.

[4] "Text Corpus Wikipedia", https://en.wikipedia. org /wiki/Text_corpus.

[5] Zin Maung Maung, Yoshiki Mikami, Management Information Systems Engineering Department Nagaoka University of Technology 1603-1 Kamitomioka, Nagaoka, Japan, "A Rule-based Syllable Segmentation of Myanmar Text", Proceedings of the IJCNLP-08 Workshop on NLP for Less Privileged Languages, Hyderabad, India, January 2008, pages 51–58.