Lexical Analyzer for Myanmar Language

Soe Lai Phyue
University of Computer Studies, Mandalay
soelaiphyue@gmail.com

Abstract

The lexical analysis within the context of language processing is to connect each word with its corresponding label in a lexicon. However, many words have more than one meaning, ambiguity word, which may make it impossible to choose the correct meaning of the word considering only the highlighted word in its context. Beside then there is also unknown word, a word does not have in the lexicon, which may need to handle for tagging and updating this word in the lexicon to improve the coverage of lexicon. For this reason, this paper proposes the lexical analyzer to solve the ambiguity of known words and to tag the unknown words of Myanmar language by using rule based approach and decision tree induction method. Moreover, to support the lexical analyzer, segmentation and pattern merging algorithm is also proposed by using the Myanmar-English computational lexicon. The propose system is effective for Myanmar language lexical analysis and can improve the coverage of the lexicon.

Keywords-lexical analysis; Natural Language Processing; Machine Learning; decision tree induction; computational lexicon

1. Introduction

Nowadays, the development of context of Natural Language Processing (NLP) in general is rapidly growth as computational linguistic field. The task of language technology is to develop efficient, high accuracy software modules that perform NLP tasks or subtasks [1]. The optimism about the marriage of ML and NLP stems from the observation that most NLP problems can be viewed as classification problems. Modern statistical machine translation (SMT) models implicitly incorporate source language context. In general, linguistic problems fall into two types of classification: (a) Disambiguation, i.e., determine the correct category from a set of possible categories and (b) Segmentation, i.e., determine the correct boundary of a segment from a set of possible boundaries [7].

In Myanmar linguistic tradition there is not a clear-cut, well-defined analysis of the inventory of parts of speech in Myanmar. It has been classified by linguists as a monosyllabic or isolating language with agglutinative features [4]. Also the lexical information of Myanmar language has not been as widely investigated. Therefore, we investigated the lexical analysis of Myanmar sentence which is important portion of Myanmar NLP applications using rules based approach that is finites state theory and statistical approached to define the finite POS.

The analyzer collaborated with bilingual computational lexicon [10] which covered the inflectional form and semantic meaning. The tagger determines word POS through rule-based analysis which rule are generated by manually. The ambiguous word and unknown word is also determined by the decision tree method. The decision tree for each source sentence is built extracted from the training data.

This paper is organized as follows. In the next section we give information about the relationship between the machine learning approach and lexical analysis techniques. Then, the framework of our approach for lexical analysis is described in section 3. Section 4 presents the experimental result of our system. Finally, section 5 provides some concluding remarks and future directions of research.

2. Overview of Machine Learning Techniques in Lexical Analysis

Machine learning is concerned with acquiring knowledge from an environment in a computational manner, in order to improve the performance. Also a greater demand for natural language based applications, are three important factors. (i) NLP require a substantial amount of knowledge (ii) a common NLP problem can be represented as a classification problem (iii) ML approach reduce the dependence of manually embedding knowledge into NLP systems, and to let learning algorithms acquire the knowledge from available data .

Linear classifiers are all relatively simple to understand and are computationally efficient. But it has particularly with 2-class problems. For example, linear threshold algorithms can calculate a weighted sum from input features, and then depending whether that sum is greater or smaller than a certain threshold, a decision can be made as its class. Thresholds are determined through the use of training data [2].

Memory-based learning is a method of classifying by having a full memory of previously seen examples at its disposal [10]. The essence of this method is that learn by comparing similarities of past experiences with new ones, as opposed to rule-based models. Traditionally, what is referred to as the performance stage of a MBL system, i.e., the classification stage, is often descended from the simple k-nn (k nearest neighbors) algorithm.

The TiMBL (Tiberg Memory-Based Learning) system, which can be found in Daelemans *et al.*, developed at Tilberg University has enjoyed large success in this domain. The TiMBL system is more complex than the basic description of MBL. It employs an algorithm to compress examples into a decision-tree like structure. This optimization step means that the classification process does not have to examine all examples in memory, and instead only focuses only nodes with important feature relevant to the instance being classified [3].

Neural networks (NNs) consist of units that are connected by links. The links are assigned a value, known as a weight. Units perform simple computations based on the inputs, and pass on their output. Networks of these units can be connected together in a suitable topology for a given task. Units in between in the input and output units are called hidden units. Once layers of hidden units are present in a NN, it is possible to represent complex problems. The learning step is achieved through training, whereby using an algorithm such as back-propagation, it is possible to update the weights within the NN to boost its performance [8].

Genetic algorithms/evolutionary computing have proved to be a successful technique in optimizing solutions, especially for difficult problems with large search spaces. GAs are prone to getting stuck at local maxima, as it is a greedy algorithm that evolves for the largest short term gain [6].

Decision trees have long been considered as one of the most practical and straightforward approaches to classification [5]. Strictly speaking, induction of decision trees is a method that generates approximations to discrete-valued functions and has been shown, experimentally, to provide robust performance in the presence of noise. Decision trees are used to partition large samples of data into a hierarchical structure. Commonly associated as a tool for classification, they are also capable of generalizing seen data into sets of rules. They are generic enough to be POS tagging.

3. Framework of Lexical Analysis

Lexical Analysis (LA) is determining the meaning of individual words, and identifying non-word tokens and Part-of-Speech (POS) tagging. This system developed the Myanmar language lexical analyzer as shown in Figure 1. To understand the morphology of each word, first tokenize the sentence and determine the word relationships. It is working together with the Myanmar-English computational lexicon [10]. The portion of the system holds all specific attributes to each word of the source sentence. Basic method of lexical analysis is the word

lookup in a lexicon and it has some problem which is word-level ambiguity that words may have several meanings, and the "correct" one cannot be chosen based on the word itself for example: the word "သွား" it may become verb, noun and particle. To resolve on the spot (i.e. POS tagging), or pass on the ambiguity we first use rule based approach and if it has still ambiguity, we solve the problem with decision tree induction using statistical method to define the definite POS of the word.

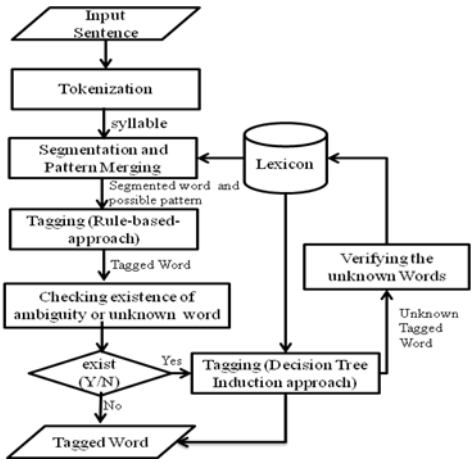

Figure 1. Overview of Lexical Analysis

3.1. Bilingual Computational Lexicon Design

The bilingual computational lexicon [10] is constructed for the further NLP application and now used in lexical analysis. For natural language systems, knowledge acquisition is a main bottleneck. The information contained in the computational lexicon is highly relevant for NLP systems and it is important resource for further NLP research. This bilingual lexicon is built base on the Myanmar WordNet lexical database [9]. Therefore the design is greatly depend on Myanmar WordNet lexical database structure and information. In this lexicon defined the noun as 26 tag set, verb as 15 tag set, adjective as 3 tag set and adverb has 2 tag set, proposition as 17 and conjunction as 8 tag set. Beside then particle are used as indicator for defining the definite POS and produced the inflected form of word. The computational lexicon contains the following information as shown in Table 1.

Table 1. Features in Bilingual Computation Lexicon

	Noun	Verb	Adjective	Adverb	Pronoun	Preposition	Conjunction
Myanmar	•	•	•	•	•	•	•
English word	•	•	•	•	•	•	•
categories	•	•	•	•	•	•	•
lexname	•	•	•	•	•	•	•
POS description	•	•	•	•	_	_	_
Definitions (Synsets)	•	•	•	•	_	_	_
sample	•	•	_	_	_	_	_
Adj-modifier	_	_	•	_	_	_	_
Semantic-Frame	_	•	_	_	_	_	_
Lexical-Frame	_	•	_	_	_	_	_
Inflected Sense	•	•	•	_	_	_	_

3.2. Word Tokenization for Myanmar Syllable

To process text computationally, syllables have to be determined first. Since Myanmar linguistic tradition there is not a clear-cut, we need to pass the tokenize syllable as a one of the step. Word tokenization is done by rules based approach.

Myanmar Syllable can be defined as consonant (C) such as က – အ, medial (M) such as ျ ြ ွ ှ etc., ending character (E) း ့ etc., single stand character(S) such as ၏ ၌ ၍, vowel sign character (V) such as ု ူ ိ ီ etc., digit (D). Myanmar syllable consists of one initial consonant, zero or more medial, zero or more vowels sign and optional dependent various signs. Single stand character and digits can act as standalone syllables. Therefore, Myanmar language has stand a single alone character and the special ending character are defined first.

And then we defined the syllable such as regular rule expression as

Syllable(W)=C{M}{V}|C{M}V*M
|C{M}{V}CM[F]|S|D.

For example

Sentence (1): "ကလေးများ ကစားကွင်း သို. သွားသည်" After tokenize sentence (1): က + လေး + များ + က + စား + ကွင်း + သို. + သွား + သည်

Sentence (2): "ကလေးများ ကန်တော်ကြီး သို. သွားသည်"

After tokenize sentence (2): က + လေး + များ + ကန် + တော် + ကြီး + သို. + သွား + သည်

3.3. Word Segmentation and Possible Pattern Merging

In linguistics, a word is a basic unit of language that carries meaning and can be spoken or written [1]. It can consist of one or more morphemes that are linked more or less tightly together. Typically, a word will consist of a root or stem and zero or more affixes. Without a word segmentation solution, no NLP application (such as Part-of-Speech (POS) tagging and translation) can be developed. Words can be combined to form phrases, clauses and sentences. A word consisting of two or more stems joined together is known as a compound word.

```
1.  Input: sentence, syllable_list
2.  Output :segmented_words, possible_patterns
3.  Begin
4.    while syllable_list is empty do
5.       while  syllable_list  is empty do
6.             Form possible_Combinition_list;
7.       end
8.    end
9.  while possible_Combination_list ((i=1,2,..) is empty do
10.      if(possible_Combination_list [i] is in lexicon) then
11.           add Meaningful-word-list;
12.      else
13.           add undefined_word;
14.      end if;
15.   end
16.   Restructure the original sentence by  using
        Meaningful-word -list and undefined_word;
17. End
```

Figure 2. Proposed Algorithm for Word Segmentation and Possible Pattern Merging

Segmentation and pattern merging is done by proposed algorithm, Figure 2, which produce possible word pattern for sentence. The merging of tokenized word is use as input to the algorithm. The output of the algorithm is the segmentation word and possible pattern. They are sent to the part of speech tagging of first phase (rule based tagging process). The merging of tokenized word is as shown in table 2.

Table 2. An example sentence tagged by the lexicon

က	verb { motion | creation}
လေး	noun {act |artifact | quantity} adjective{all} verb {change | motion |perception | contact |communication| creation}
ကလေး	noun {person}
များ	particle
က *(ကန်)*	verb { motion | creation} *(noun{act| communication| object| feeling| event| cognition}* *verb{communication|competation|contact|motion|cognition|consumption})*
စား *(တော်)*	verb { consumption | cognition | emotion | contact | change} *(noun{attribute| artifact)}adj{all})*
ကစား	verb {motion | creation | competition | consumption |change |emotion| contact | body |social | stative}
ကွင်း *(ကြီး)*	noun {object |artifact | location | process | shape | cognition| communication |group | phenomenon| act} *(adj{all})*
ကစားကွင်း	noun {artifact | location}
သို.	Preposition
သွား	noun {artifact | body} verb {motion | creation | competition | consumption |change |emotion| contact | body |social |stative}
သည်	Preposition

The patterns are form with known word. Other patterns which form contain with unknown word are ignore in rule based approach. The possible sentence patterns for an example sentence 1 and 2 are as followed.

For Sentence 1:

Pattern 1: က {V} + လေး {N/Adj/V} + များ {Pa} + က {V} + စား {V} + ကွင်း {N} + သို. {Pr} + သွား {N/V} + သည် {Pr}

Pattern2: ကလေး {N} + များ {Pa} + က {V} + စား {V} + ကွင်း {N} + သို. {Pr} + သွား {N/V} + သည် {Pr}

Pattern3: က {V} + လေး {N/Adj/V} + များ {Pa} + ကစား {V} + ကွင်း {N}+ သို. {Pr} + သွား {N/V} + သည် {Pr}

Pattern4: ကလေး {N} + များ {Pa} + ကစား {V} + ကွင်း {N} + သို. {Pr} + သွား {N/V} + သည် {Pr}

Pattern5: က {V} + လေး {N/Adj/V} + များ {Pa} + ကစားကွင်း {N} + သို. {Pr} + သွား {N/V} + သည် {Pr}

Pattern 6:ကလေး {N} +များ {Pa} +ကစားကွင်း {N} +သို. {Pr} + သွား {N/V} + သည် {Pr}

For sentence 2:

Pattern 1: က {V} + လေး {N/Adj/V} + များ {Pa} + ကန် {N,V} + တော် {N,Adj} + ကြီး {Adj} + သို. {Pr} + သွား {N/V} + သည် {Pr}

Pattern 2:ကလေး {N} + များ {Pa} + ကန် {N,V} + တော် {N,Adj} + ကြီး {Adj} + သို. {Pr} + သွား {N/V} + သည် {Pr}

After the sentence 1 and 2 are segmented as words and these words are merged as the possible pattern, it has contained the ambiguity words. Therefore we need to define this word with definite POS using the rule base approach.

3.4. Rule based POS tagging of Myanmar language

Part-of-speech tagging (POS tagging or POST), also called grammatical tagging, is the process of marking up the words in a text as corresponding to a particular, based on both its definition, as well as its context i.e. relationship with adjacent and related words in a phrase, sentence, or paragraph. The widespread interest in tagging is founded on the belief that many Natural Language Processing (NLP) applications will benefit from syntactically disambiguated text. This is the ultimate motivation for part-of-speech tagging. There are many approaches to automated part of speech tagging. Rule based tagging is the old line research but it is more efficient time consuming. CFG grammars are widely used in linguistics. Most modern linguistic theories of grammar incorporated some notion from context free grammar.

1.

- ဝါကျ ⟶ စကားစု
- ဝါကျ ⟶ နာမ်ပုဒ်စု(ကံပုဒ်စု)+ကြိယာပုဒ်စု
- နာမ်ပုဒ်စု ⟶ နာမ်+နာမ်ဝိဘတ်
- နာမ်ပုဒ်စု ⟶ နာမ်စား+ဝိဘတ်
- နာမ်ပုဒ်စု ⟶ နာမ်စား+နာမ်+ဝိဘတ်
- နာမ်ပုဒ်စု ⟶ နာမ်စား+ပိုင်ဆိုင်ခြင်းပြဝိဘတ်+ နာမ်ပုဒ်စု
- နာမ်ပုဒ်စု ⟶ နာမ်+ပိုင်ဆိုင်ခြင်းပြဝိဘတ်+ နာမ်ပုဒ်စု
- နာမ်ပုဒ်စု ⟶ နာမ် +နာမ်ပစ္စည်း+ နာမ်ဝိဘတ်
- နာမ်ပုဒ်စု ⟶ နာမ်စား+ နာမ်ပစ္စည်း+ ဝိဘတ်
- နာမ်ပုဒ်စု ⟶ နာမ်စား+နာမ်ပစ္စည်း+ပိုင်ဆိုင်ခြင်းပြဝိဘတ်+နာမ်ပုဒ်စု
- နာမ်ပုဒ်စု ⟶ နာမ်+နာမ်ပစ္စည်း+ပိုင်ဆိုင်ခြင်းပြဝိဘတ်+ နာမ်ပုဒ်စု
- နာမ်ပုဒ်စု ⟶ နာမဓိသေသန+နာမ်ပုဒ်စု
- ကြိယာပုဒ်စု ⟶ အချိန်ပြပုဒ်စု+ကြိယာပုဒ်စု
- ကြိယာပုဒ်စု ⟶ နေရာပြပုဒ်စု+ကြိယာပုဒ်စု
- ကြိယာပုဒ်စု ⟶ နာမ်ပုဒ်စု+ကြိယာပုဒ်စု
- ကြိယာပုဒ်စု ⟶ ကြိယာဝိသေသန+ ကြိယာပုဒ်စု
- ကြိယာပုဒ်စု ⟶ နာမဓိသေသန+ ကြိယာပုဒ်စု
- ကြိယာပုဒ်စု ⟶ ကြိယာ + ဝိဘတ်
- ကြိယာပုဒ်စု ⟶ ကြိယာ+ ကြိယာထောက်ပစ္စည်း+ ဝိဘတ်
- နေရာပြပုဒ်စု ⟶ နေရာပြပုဒ်စု+ ဝိဘတ်
- နေရာပြပုဒ်စု ⟶ နေရာပြပုဒ်
- အချိန်ပြပုဒ်စု ⟶ အချိန်ပြပုဒ်စု + ဝိဘတ်
- အချိန်ပြပုဒ်စု ⟶ အချိန်ပြပုဒ်

Figure 3. Context Free Grammar Rules for Simple Sentence Pattern

CFG is an abstract model for associating structures with strings but it is not intended as model of how humans produce sentences. Sentences that can be derived by a grammar G belong to the formal language defined by G, and are called grammatical sentences with respect to G. Sentences that cannot be derived by G are ungrammatical Sentences with respect to G. The language L_G defined by grammar G is the set of

strings composed of terminal symbols that are derivable from the start symbol:

$L_G = \{w \mid w \in T \text{ and } S \text{ derives } w\}$

The generation of CFG rules has two steps.

2. Lexical rules recognizing POS from the Myanmar words are generated.

CFG rules recognizing phrase from POS are generated.

CFG rule of simple sentence for Myanmar language is as shown in Figure 3.To defined the POS of each word, we used the CFGs as rules which parsing is start with sentence and left to right parsing structure. The sample parsing of sentence 1 is shown in Figure 4.

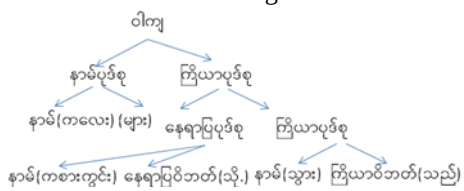

Figure 4. Parsing of example sentence1 with CFG rule

According an example, sentence 1 met CFG rule and which can tag each word in this sentence as ကလေး{N} + များ{Pa} + ကစားကွင်း{N} + သို့{Pr} + သွား{V} + သည်{Pr}. By applying the sentence 1 with CFG rule, we can easily tag of ambiguity words.

However, the example sentence 2 of possible pattern does not meet with CFG rule. For this case we need to use all pattern of these sentence, contain unknown word, to tag these words using with decision tree induction method.

3.5 Decision Tree Induction and POS Tagging

Although the hardest part of the tagging process is accomplished by a computation lexicon, a POS tagger cannot solely consist of a lexicon due to: (i) morphosyntactic ambiguity (e.g.," သွား" as verb or noun or particle) and (ii) the existence of unknown words (e.g., proper nouns, place names, compounds, etc.). When the lexicon can assure high coverage, unknown word guessing can be viewed as a decision taken upon the POSs of open-class.

If we do not meet the POS with rule based approach, we considered the unknown word and ambiguity word in the possible pattern. In the example sentence 2, we consider the unknown word pattern as

Pattern3: က{V} + လေး{N/Adj/V} + များ{Pa} + ကန်တော်{UN} + ကြီး{Adj} + သို့.{Pr} + သွား{N/V} + သည်{Pr}

Pattern5: က {V} + လေး{N/Adj/V} + များ{Pa} + ကန်တော်ကြီး{UN} + သို့.{Pr} + သွား{N/V} + သည်{Pr}

Pattern 6:ကလေး{N} + များ{Pa} + ကန်တော်ကြီး{UN} + သို့.{Pr} + သွား{N/V} + သည်{Pr}

The role of decision trees now becomes evident. According to the tagging performed by the lexicon, a word belonging to *n* POSs receives *n* tags (typically *n* is two or three). Each of the *n* tags contains a different POS value. The goal is to keep the tag with the contextually appropriate POS and discard the rest. When a word with two or three tags appears, its ambiguity scheme is identified and the corresponding decision tree is selected. The tree is traversed according to the results of tests performed on contextual tags. This traversal returns the contextually appropriate POS. The ambiguity and unknown is resolved by eliminating the tag(s) with different POS than the one returned by the decision tree.

Decision trees are built top-down. One selects a particular attribute of the instances available at a node, and splits those instances to children nodes according to the value each instance has for the specific attribute. This process continues recursively until no more splitting along any path is possible, or until some splitting termination criteria are met. After splitting has ceased, it is sometimes an option to prune the decision tree (by turning some internal nodes to leaves) to hopefully increase its expected accuracy.

Given a concrete node of a decision tree with its associated set of examples *X*, the probability of a certain tag *t* is straightforwardly estimated by MLE as the proportion of examples that have tag *t* over the total number of examples, that is:

$$\hat{p}(t \mid X) = \frac{f(t \mid X)}{\| X \|} \qquad (1)$$

In order to smooth the MLE probability estimates, one can consider the following general formulation for discounting some probability mass from frequently seen events to redistribute it among the less frequent events:

$$\hat{p}(t \mid X) = \frac{f(t \mid X) + \lambda}{\| X \| + \lambda K} \qquad (2)$$

where *K* is the number of possible tags, and A is a positive real value. When A = 1 the above formula is known as Laplaces's law of succession. As some authors observe, the appropriate value for A in general problems of language modelling is significantly lower than 1. We have set this value to A = *(K—l)/K*, which depends on the number of possible tags. It starts in 0.5 for two-tag ambiguity classes and increases asymptotically to 1 (Laplace's law) as the number of possible tags increase.

The splitting process requires some effort to come up with informative attribute tests. This paper relaxes the classical definition of the value of an attribute and allows an instance to have a set of values for some attribute. As presented earlier, this deviation is absolutely critical for the POS tagging task. Set-valued attributes require extra care in how they are handled, as the usual splitting criteria may have to be modified. The training pattern would look like Figure 4.

Figure 5. Example training pattern

Specifically, when instances, during training are allowed to follow more than one branch out of a node, it may turn out that the usual entropy-based metrics deliver loss rather gain of information. Needless to say this requires exceptional handling. In our example the preposition "သို့" is evident the unknown tag of "ကန်တော်ကြီး" as a place of Noun tag. Therefore, we got the tagging pattern as pattern 6: ကလေး{N} + များ{Pa} + ကန်တော်ကြီး {N(Place)} + သို့.{Pr} + သွား{V} + သည်{Pr}.

4. Experimental Results

The System is more correct and performance is depended on the word exists in the lexicon. The more word in the lexicon, the better of the system's performance. Execution time is measured as the time taken from getting input text file to generating analyzed text. That means it is time spent for the entire system execution. Syllable count and corresponding average execution time with various words is shown in Figure 5.

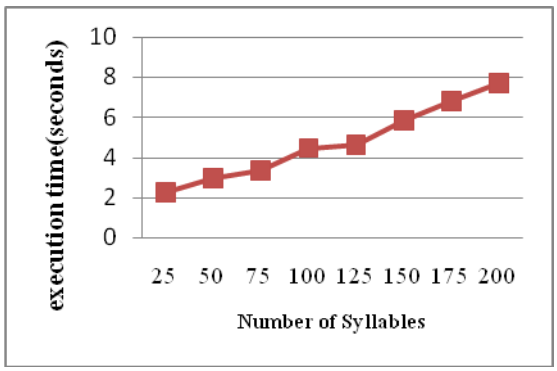

Figure 6. Execution Time of Segmentation and Pattern Merging Versus Number of Syllables

To evaluate our method, we conducted a translation experiment was made as follows. We implemented the system with java programming language on Core i5 processor laptop with 2 GB of RAM.

4.1 Performance of Tagger

Precision and recall for overall function tagging process is calculated. For the context of function tagging, the precision and recall of grammatical function tagger is calculated by using the following equations.

$$precision = \frac{\text{No: of Correct function Tags}}{\text{Total Function Tags}} * 100 \qquad (3)$$

$$recall = \frac{\text{No: of Correct function Tags}}{\text{No:of Actural Existiong Function Tags}} * 100 \quad (4)$$

The results of precision and recall of grammatical function tagger is as shown in Table4. In our case, it is a measure of agreement between the target tagged word and the tagged of word using our system. The evaluation results demonstrate that the proposed lexical analyzer is effective to define the accurate POS.

Table 3. Result of Grammatical Function Tagging on Testing Set

Type of Sentence	Recognized	Actual	Correct	% of Precision	% of Recall
Ambiguity +Known	170+309 (479)	163+309 (472)	155+309 (464)	96.86	98.3
Unknown +Known	117+253 (370)	107+253 (360)	96+253 (349)	94.32	96.94
Ambiguity &Unknown+ Known	47+205 (252)	42 +205 (247)	39+205 (244)	96.83	98.79
Total	1101	1079	1057	96	97.96

7. Conclusion

Lexical analysis is a first step of a NLP applications and this step is fundamental due bad results at this step are transmitted to later steps, growing the errors exponentially. In this paper, we proposed the framework for lexical analysis for Myanmar language. Myanmar Language has not delimiter or word boundary. Therefore tokenization and word segmentation is important for further Myanmar NLP application.

To accurate the word boundary and to solve the word ambiguities and unknown in POS tagging, the rule based context free grammar or regular expressions and decision tree induction method which is probabilistic model are combined to finite tagging process. Also the unknown tagged word is updated to the lexicon. As a result, the proposed system was effective for tagging process. Moreover, it can improve the reliable and wide coverage for lexicon.

References

[1] A.Roberts, Machine Learning in Natural Language Processing, CiteSeerX-Scientific Literature Digital Library and Search Engine, October 16, 2003

[2] D.Roth and D.Zelenko, Part of speech tagging using a network of linear separators,. In Proceedings of the joint 17th International Conference on Computational Linguistics and 36th Annual Meeting of the Association for Computational Linguistics (COLING-ACL) Montr´eal, Canada, pp 1136-1142, 1998

[3] Daelemans, W., Van den Bosch, A., Zavrel, J., Veenstra, J., Buchholz, S. and Busser, G. ,Rapid development of NLP modules with memory-based learning, In Proceedings of ELSNET in Wonderland, pp 105-.113, 1998

[4] F.K.L. Chit Hlaing, Burmese, Cognitive Science 15, pp 271-391, 1991

[5] G.Orphanos, D.Kalles, T. Papagelis and D.Christodoulakis, Decision Trees and NLP: A case strudy in POS Tagging, ACAI'99, 1999.

[6] Kazakov, Dimitar, Unsupervised learning of naive morphology with genetic algorithms, In Workshop Notes of the ECML/MLnet workshop on empirical learning of Natural Language Processing Tasks, 1997

[7] M. Anwar, et.al , Syntax Analysis and Machine Translation of Bangla Sentences, IJCSNS International Journal of Computer Science and Network Security, VOL.9 No.8, August 2009

[8] M.Qing and I.Hitoshi, A multi-neuro tagger using variable lengths of contexts, In Proceedings of the joint 17th International Conference on Computational Linguistics and 36th Annual Meeting of the Association for Computational Linguistics (COLING-ACL), Montr´eal, Canada, pp 802-.806, 1998

[9] S.L.Phyue, Construction Myanmar WordNet Lexcial Database, 9th International Conference on IEEE Student of Conference on Research and Development (IEEE-SCOReD'11), Malaysia, 2011

[10] S.L.Phyue, Building the Myanmar Language Resources like Myanmar WordNet and Bilingual Computational Lexicon, Applied Information and Computational Technology (AICT'11), Mandalay, Myanmar, 2011.

[11] T.M.Cover, and P.E.Hart, Nearest neighbor pattern classification, IEEE Transactions on Information Theory, pp 21-.27, 1967