Lexicalized HMM-based Part-of-Speech Tagger for Myanmar Language

Phyu Hninn Myint, Tin Myat Htwe and Ni Lar Thein
University of Computer Studies, Yangon.
phm.ucsy@gmail.com

Abstract

Part-of-speech (POS) tagging is the process of assigning the part-of-speech tag or other lexical class marker to each and every word in a sentence. In many Natural Language Processing (NLP) applications such as word sense disambiguation, information retrieval, information processing, parsing, question answering, and machine translation, POS tagging is considered as the one of the basic necessary tools. Identifying the ambiguities in lexical items is the challenging objective in the process of developing an efficient and accurate POS Tagger. This paper proposes the developments for POS-tagger and POS-tagset of Myanmar language, which is very essential computational linguistic tool needed for many natural language processing (NLP) applications. Since most previous works for HMM-based tagging consider only part-of-speech information in contexts, their models cannot utilize lexical information which is crucial for resolving some morphological ambiguity. In this paper, a simple method to build Lexicalized Hidden Markov Models (L-HMMs) is introduced for improving the precision of part-of-speech tagging in Myanmar language. In experiments, lexicalized models achieve higher accuracy than non-lexicalized models.

1. Introduction

Part-of-speech (POS) tagging has a crucial role in different fields of natural language processing (NLP) including machine translation. POS tagging is defined as the process of assigning to each word in a sentence, a label which indicates the status of that word within some system of categorizing the words of that language according to their morphological and/or syntactic properties. A part-of-speech is a grammatical category, commonly including verbs, nouns, adjectives, adverbs, and so on. POS tagger has to be applied to assign a single best POS to every word. The ambiguity is the key issue that must be addressed and solved while designing a pos tagger since many words belong to more than one lexical class. The different context words behave differently and hence the challenge is to correctly identify the POS tag of a token appearing in a particular context [1]. Because of the importance and difficulty of this task, a lot of work has been carried out to produce automatic POS taggers. Most of the automatic POS taggers, usually based on Hidden Markov Models (HMMs), rely on statistical information to establish the probabilities of each scenario. The statistical data are extracted from previously hand-tagged texts, called pre-tagged corpus. These stochastic taggers neither require knowledge of the rules of the language nor try to deduce them. The context in which the word appears helps to decide which tag is its more appropriate tag and this idea is the basis for most taggers. However, most previous HMM-based taggers consider only POS information in contexts, and so they cannot capture lexical information which is necessary for resolving some morphological ambiguity [9]. The use of lexicalization technique can substantially improve the performance of a HMM-based tagging system.

In this paper, statistical tagging approach is used and it needs pre-tagged training corpus. Therefore, a small corpus has been created manually at first. Using it as a training corpus, a bigram POS-tagger has been built. To get larger-sized corpus, this tagger has run on an untagged corpus. Although most of the words in the corpus have been tagged with their right tags, some words can be tagged with the wrong tags since the training corpus size is small. Thus, manually checking errors and updating with the right tags are performed so that it can be used as a larger-sized training corpus again. LHMM-based POS tagger is proposed for Myanmar language POS tagging. Standard HMMs and lexicalized HMMs models are developed to train the corpus and Viterbi algorithm is used to disambiguate POS tags.

The rest of the paper is organized as follows: Section 2 introduces the POS tagging approaches. Section 3 describes about the standard HMMs, the lexicalized HMMs and Viterbi decoding. A brief description of the proposed approach is presented in Section 4. Finally, performance evaluation and some conclusions on this work are given in Section 5 and Section 6 respectively. The proposed Myanmar Part-of-Speech Tagset and the specific categories are shown at the end of the paper as Appendix.

2. POS Tagging Approaches

POS taggers are broadly classified into three categories called Rule-based approach, Statistical or Stochastic techniques and Hybrid Approach. In case of rule based approach, hand-written rules are used to distinguish the tag ambiguity. Stochastic taggers are either HMM based, choosing the tag sequence which maximizes the product of word likelihood and tag sequence probability, or cue-based, using decision trees or maximum entropy models to combine probabilistic features. The stochastic taggers are further classified in to supervised and unsupervised taggers. Each of these supervised and unsupervised taggers are categorized into different groups based on the particular algorithm used [1].

2.1 Rule Based POS tagging

The rule based POS tagging models apply a set of hand written rules and use contextual information to assign POS tags to words. One of the first and widely used English POS-taggers employs rule based algorithms is “Brill’s tagger”. The earliest algorithms for automatically assigning part-of-speech were based on two-stage architecture. The first stage used a dictionary to assign each word a list of potential parts of speech. The second stage used large lists of hand-written disambiguation rules to bring down this list to a single part-of-speech for each word [1]. Rule-based taggers [2][3] tried to assign a tag to each word using a set of hand-written rules. These rules could specify, for instance, that a word following a determiner and an adjective must be a noun. Of course, this means that the set of rules must be properly written and checked by human experts.

2.2. Stochastic based POS tagging

Stochastic taggers generally resolve tagging ambiguities by using a training corpus to compute the probability of a given word having a given tag in a given context. The stochastic approach finds out the most frequently used tag for a specific word in the annotated training data and uses this information to tag that word in the unannotated text. A stochastic approach required a sufficient large sized corpus and calculates frequency, probability or statistics of each and every word in the corpus. Supervised and unsupervised are two broad categories of stochastic based approach. The supervised POS tagging models require pre-tagged corpora which are used for training to learn information about the tagset, word-tag frequencies, rule sets etc. The performance of the models generally increases with the increase in size of this corpus. Hidden Markov Model (HMM) based POS tagging is an alternative to the word frequency approach which is known as the n-gram approach that calculates the probability of a given sequence of tags. It determines the best tag for a word by calculating the probability that it occurs with the n previous tags, where the value of n is set to 1, 2 or 3 for practical purposes. These are known as the Unigram, Bigram and Trigram models. The most common algorithm for implementing an n-gram approach for tagging new text is known as the HMM's Viterbi Algorithm. The Viterbi algorithm is a search algorithm that avoids the polynomial expansion of a breadth first search by trimming the search tree at each level using the best 'm' Maximum Likelihood Estimates (MLE) where 'm' represents the number of tags of the following word. For a given sentence or word sequence, HMM taggers choose the tag sequence that maximizes as in the following formula:

$$P(\text{word} \mid \text{tag}) * P(\text{tag} \mid \text{previous n tags})$$

A bigram-HMM tagger of this kind chooses the tag t_i for word w_i that is most probable given the previous tag t_{i-1} and the current word w_i:

$$t_i = \underset{j}{argmax} \, P(t_j \mid t_{i-1}, w)$$

Unlike the supervised models, the unsupervised POS tagging models do not require a pre-tagged corpus. Instead, they use advanced computational methods like the Baum-Welch algorithm to automatically induce tagsets, transformation rules etc. Based on the information, they either calculate the probabilistic information needed by the stochastic taggers or induce the contextual rules needed by rule-based systems or transformation based systems. The stochastic (probabilistic) approach [5][11] used a training corpus to pick the most probable tag for a word. All probabilistic methods cited above are based on first order or second order Markov models [1].

The stochastic approach is better than rule based tagging to alleviate the manual works. If we have a POS tagged corpus, we are able to obtain a set of syntactic rules easily. Stochastic taggers exploit the power of probabilities and machine learning techniques in order to disambiguate and tag sequences of words [11]. One widely and successfully applied approach to statistical modeling is Hidden Markov Models (HMM). There are a few other techniques which use probabilistic approach for POS tagging, such as the Tree Tagger. For all of the supervised POS Tagging, a pre-tagged corpus is a prerequisite. On the other hand, there is the unsupervised POS tagging [4][6][10] technique, and it does not require any pre-tagged corpora.

2.3. Transformation-based POS tagging

The transformation-based approach combines the rule-based approach and statistical approach. It picks the most likely tag based on a training corpus and then applies a certain set of rules to see whether the tag should be changed to anything else. It saves any new rules that it has learnt in the process, for future use. One example of an effective tagger in this category is the Brill tagger [2][3].

In general, the supervised tagging approach usually requires large sized pre-annotated corpora for training, which is difficult for most of the cases. But recently, good amount of work has been done to automatically induce the transformation rules. One approach to automatic rule induction is to run an untagged text through a tagging model and get the initial output. A human then goes through the output of this first phase and corrects any erroneously tagged words by hand. This tagged text is then submitted to the tagger, which learns correction rules by comparing the two sets of data. Several iterations of this process are sometimes necessary before the tagging model can achieve considerable performance. The transformation based approach is similar to the rule based approach in the sense that it depends on a set of rules for tagging [1].

3. Hidden Markov Models of Supervised POS Tagging

3.1. Standard Hidden Markov Models

From the statistical point of view, POS tagging can be defined as a maximization problem. Let $T = \{t_1,t_2,...,t_N\}$ be a set of POS tags and let $W = \{w_1,w_2,...,w_m\}$ be the vocabulary of the application. Given an input sentence of length L, $W = w_1,...,w_L$, the process consists of finding the sequence of POS tags of maximum probability, that is:

$$\hat{T} = \underset{T}{argmax} \, P(T|W) = \underset{T}{argmax} \frac{P(W|T)P(T)}{P(W)} \qquad (1)$$

In equation (1), the contextual probabilities or language model, P(T), represent the possible or probable sequences of POS tags. The lexical probabilities, P(W|T), represent the relation between the vocabulary and the POS tags. To solve this equation, the Markov assumptions should be made in order to simplify the problem. For first-order Markov models (bigrams) and taking into account the Markov assumptions, the problem is reduced to solving the following equation (2):

$$\hat{T} = \underset{T}{argmax} \prod_{i=1}^{n} P(t_i|t_{i-1})P(w_i|t_i) \qquad (2)$$

In standard HMMs, the appearance of current word w_i depends only on current tag t_i during known word tagging, and the assignment of current tag t_i depends only on its previous tag t_{i-1}. $P(w_i \mid t_i)$ is the so-called lexical probability; and $P(t_i \mid t_{i-1})$ denotes the contextual tag probability.

3.2. Lexicalized HMMs

In the lexicalized HMMs, the appearance of current word w_i is assumed to depend not only on current tag t_i but also its previous word and the assignment of current tag t_i is supposed to depend both its previous word w_{i-1} and its previous tag t_{i-1}. Thus, we have the lexicalized HMMs for POS tagging as follows:

$$\hat{T} = \underset{T}{argmax} \prod_{i=1}^{n} P(t_i|w_{i-1}, t_{i-1})P(w_i| w_{i-1}, t_i) \qquad (3)$$

In comparison with the standard HMMs, the lexicalized HMMs can provide richer contextual information for the assigning of tags to known words, including both contextual words and contextual tags, which will result in improvement of accuracy in unknown word tagging.

For simplification, the maximum likelihood estimation (MLE) has to be applied to estimate the parameters in above equations. In MLE, parameters are estimated with their relative frequencies that are extracted directly from the manual corpus for training. The MLE of HMMs and LHMMs is formulated respectively in equation (4) and (5).

$$\begin{cases} P(w_i \mid t_i) = \dfrac{Count(w_i\ ,\ t_i)}{Count(t_i)} \\ P(t_i \mid t_{i-1}) = \dfrac{Count(t_{i-1}\ ,\ t_i)}{Count(t_{i-1})} \end{cases} \qquad (4)$$

$$\begin{cases} P(w_i \mid w_{i-1}, t_i) = \dfrac{Count(\ w_{i-1}, w_i\ ,\ t_i)}{Count(w_{i-1}, t_i)} \\ P(t_i \mid w_{i-1}, t_{i-1}) = \dfrac{Count(w_{i-1}, t_{i-1}\ ,\ t_i)}{Count(w_{i-1}, t_{i-1})} \end{cases} \qquad (5)$$

Though the MLE has the advantage of simpleness, it can yield zero probabilities for any cases that are not observed in the training data. In the implementation, we employ the linear interpolation smoothing technique to avoid this problem of data sparseness. HMMs are smoothed as shown in equation (6).

$$\begin{cases} P'(w_i \mid t_i) = \lambda\, P(w_i \mid t_i) + \dfrac{1-\lambda}{Count(\ t_i)} \\ P'(t_i \mid t_{i-1}) = \mu\, P(t_i \mid t_{i-1}) + (1-\mu)P\,(\,t_i) \end{cases} \qquad (6)$$

In smoothing the lexicalized HMMs, we use non-lexicalized probabilities to smooth the relevant lexicalized probabilities. This process is given in detail in equation (7) :

$$\begin{cases} P'(w_i|w_{i-1}, t_i) = \lambda P(w_i|w_{i-1}, t_i) + (1-\lambda)P(w_i|t_i) \\ P'(t_i|w_{i-1}, t_{i-1}) = \mu P(t_i|w_{i-1}, t_{i-1}) + (1-\mu)P(t_i|t_{i-1}) \end{cases} \qquad (7)$$

The sample lexical and contextual probabilities of our lexicalized models are shown in the following tables Table 1 and Table 2 respectively. In lexical probabilities, current word (w_i) depends on its previous word (w_{i-1}) and current tag (t_i).

Table 1. Sample Lexical Probabilities

w_i	w_{i-1}	t_i	$P(w_i \mid w_{i-1}, t_i)$
တွင်	ကျောင်း	PPM.Place	0.0882
တွင်	နာရီ	PPM.Time	0.0909
သည်	ဖြစ်	SF.Declarative	0.9098
သည်	ခဲ့	SF.Declarative	0.919
သည်	သူ	PPM.Subj	0.869

In contextual probabilities, current tag (t_i) depends on its previous word (w_{i-1}) and tag (t_{i-1}).

Table 2. Sample Contextual Probabilities

t_i	w_{i-1}	t_{i-1}	$P(t_i \mid w_{i-1}, t_{i-1})$
SF. Declarat-ive	ဖြစ်	VB. Common	0.7544
SF. Declarat-ive	ခဲ့	Part. Support	0.6296
PPM. Direction	မြို့	NN. Location	0.4333

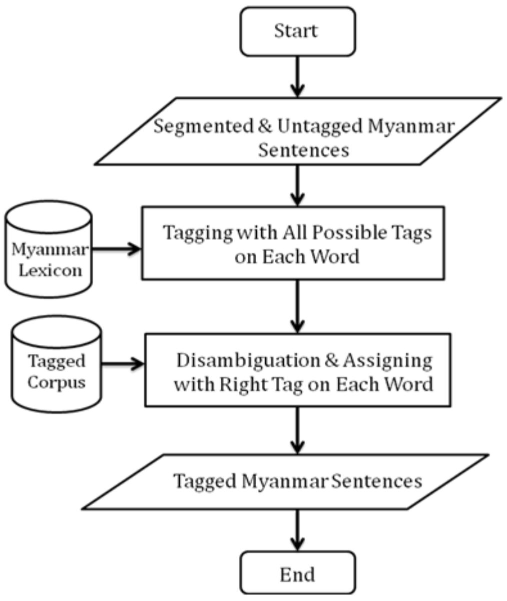
3.3. Decoding Model

The Viterbi search algorithm is used for HMM decoding. The best probable path (best tag sequence) for a given word sequence is found out by using the Viterbi tagging algorithm. It is calculated by the following formula [8].

for i from 1 to n:

$$argmax = P(t_i \mid w_{i-1}, t_{i-1}) P(w_i \mid w_{i-1}, t_i)$$

4. The Proposed Approach



Start
Segmented & Untagged Myanmar Sentences
Myanmar Lexicon
Tagging with All Possible Tags on Each Word
Tagged Corpus
Disambiguation & Assigning with Right Tag on Each Word
Tagged Myanmar Sentences
End

Figure 1. System Framework

4.1. Segmented & Untagged Myanmar Sentences Insertion

First of all, Myanmar sentences in which every word is segmented and untagged, can be inserted as an input. The system makes sentence level identification and the process proceeds in each sentence. For example,

- ❑ မြမြ သ ည် သူ ၏ သွား ကို နှုတ် ရန် ရန် ဆေးခန်း သို့ သွား ခဲ့ သည် ။
- ❑ ကျန်းမာ ခြင်း သည် လာဘ် တစ် ပါး ဖြစ် သည် ။
- ❑ သံလွင် မြစ် သ ည် မြန်မာပြည် တောင်ပိုင်း သို့ ဦးတည် ဦးတည် စီးဆင်း သွား နေ သည်။

4.2. Tagging with All Possible Tags on Each Word

For tagging with all possible tags on each word in the input sentence, Myanmar Lexicon is applied. It consists of Myanmar stem words tagged with all possible tags for these words. The system annotates each word with all possible basic POS tags and categories by using the lexicon. If the input word is unknown in the lexicon, the training data has to be used to extract the possible tags. Since there is one sentence final tag at the end of a Myanmar sentence and these sentence final words are existed in the lexicon, we can get possible tags for unknown words by guessing from the sentence final tags. In the contextual probabilities, the possible previous tags can be found for all unknown words and they can be annotated with these tags.

Example (1) for tagging all known words with all possible tags is as follows::

- မြမြ _#NNP.Livingthing
- သည်_#SF.Declarative #PPM.Subj
- သူ_#PRN.Person
- ၏_#PPM.Possessive #SF.Declarative
- သွား_#VB.Common #NN.Body
- ကို_#PPM.Obj
- နှုတ်_#VB.Common #NN.Common
- ရန် _#CC.Sent
- ဆေးခန်း _#NN.Location
- သို့ _#PPM.Direction
- ခဲ့ _#Part.Support

Example (2) for tagging unknown words between known words is as follows::

- ကျန်းမာ_# *UNKNOWN*
- ခြင်း_#Part.Common #NN.Object
- လာဘ်_# *UNKNOWN*
- တစ်_#NN.Cardinal #VB.Common
- ပါး_#Part.Type #JJ. Demonstrative
- ဖြစ်_#VB.Common
- သည် _#SF.Declarative #PPM.Subj

In example (2), after tagging known words with all possible tags, unknown words can be tagged with possible tags by using contextual relation from trained data as follows::

- ကျန်းမာ_# *VB.Common # JJ.Demonstrative*
- ခြင်း_#Part.Common #NN.Object
- လာဘ်_# *NN.Common # Part.Negative# Part.Common*
- တစ်_#NN.Cardinal #VB.Common
- ပါး_#Part.Type #JJ. Demonstrative
- ဖြစ်_#VB.Common
- သည် _#SF.Declarative #PPM.Subj

Since we use the basic tags for POS tagging, stem verbs are separately tagged with their basic tags. In a Myanmar sentence, more than one stem verbs can be seen consecutively and more than one support particles between verb and sentence final word. In the next step of POS tagging called chunking, these words can be grouped together as one verb chunk.

Example (3) for tagging consecutive verbs and support particles is as follows ::

- သံလွင်_#NNP.Location
- မြစ် _#NN.Location
- သည် _#SF.Declarative #PPM.Subj
- မြန်မာပြည် _#NNP.Location
- တောင်ပိုင်း_#NN.Location
- သို့ _#PPM.Direction
- ဦးတည်_#VB.Common
- စီးဆင်း _#VB.Common
- သွား_#VB.Common#NN.Body#Part.Support
- နေ _# VB.Common#NN.Common#Part.Support

4.3. Disambiguation and Assigning with Right Tag on Each Word

Since the right tag has to be chosen for each word, disambiguation is needed to be done for the words that possess more than one POS tags. For disambiguation, pre-tagged corpus is used for training data. LHMM and HMM models are developed to train the corpus and Viterbi algorithm is applied to choose the right tag for every words.

After disambiguation, the sample outputs are as follows::

- မြမြ_#NNP.Livingthing
- သည်_#PPM.Subj
- သူ_#PRN.Person
- ၏_#PPM.Possessive
- သွား_#NN.Body
- ကို_#PPM.Obj
- နှုတ်_#VB.Common
- ရန် _#CC.Sent
- ဆေးခန်း _#NN.Location
- သို့ _#PPM.Direction
- သွား_#VB.Common
- ခဲ့ _#Part.Support
- သည်_#SF.Declarative

- ကျန်းမာ_#VB.Common
- ခြင်း_#Part.Common
- သည်_#PPM.Subj
- လာဘ်_# NN.Common
- တစ်_#NN.Cardinal
- ပါး_#Part.Type
- ဖြစ်_#VB.Common
- သည် _#SF.Declarative

- ❑ သံလွင်_#NNP.Location
- ❑ မြစ် _#NN.Location
- ❑ သည် _#PPM.Subj
- ❑ မြန်မာပြည် _#NNP.Location
- ❑ တောင်ပိုင်း_#NN.Location
- ❑ သို့ _#PPM.Direction
- ❑ ဦးတည်_#VB.Common
- ❑ စီးဆင်း _#VB.Common
- ❑ သွား_#Part.Support
- ❑ နေ _# Part.Support
- ❑ သည့် _#SF.Declarative

4.4. Tagged Myanmar Sentences Generation

After disambiguation, the right tags are tagged to every word. The output is Myanmar sentences in which every word is tagged with its POS tag.

- ❑ မြွေ_NNP.Livingthing # သည်_PPM.Subj # သူ_PRN.Person # ၏_PPM.Possessive # သွား_NN.Body # ကို_PPM.Obj # နှုတ်_VB.Common # ရန်_CC.Sent # ဆေးခန်း_NN.Location # သို့_PPM.Direction # သွား_VB.Common # ခဲ့_Part.Support # သည်_SF.Declarative

- ❑ ကျန်းမာ_VB.Common # ခြင်း_Part.Common # သည်_PPM.Subj # လာဘ်_NN.Common # တစ်_Part.Number # ပါး_Part.Type # ဖြစ်_VB.Common # သည်_SF.Declarative

- ❑ သံလွင်_NNP.Location # မြစ်_NN.Location # သည်_PPM.Subj# မြန်မာပြည်_NNP.Location # တောင်ပိုင်း_NN.Location # သို့_PPM.Direction # ဦးတည်_VB.Common # စီးဆင်း_VB.Common # သွား_Part.Support # နေ_Part.Support # သည်_SF.Declarative

5. Performance Evaluation

5.1. Experiments Data

In order to measure the performance of the tagger, many experiments have been done on different untagged corpora. The training corpus has 1200 Myanmar POS tagged sentences and the average sentence length is about 12 words. The Myanmar lexicon has 3000 words tagged with all possible tags.

The performance of the tagger is evaluated by using testing corpora which comprise different types of words. Testing words can be classified as known words and unknown words. "Known words" means the words including in the lexicon and “Unknown Words” means the words that are not pre-inserted in the lexicon.

Three testing corpora are used for evaluation in order to calculate the accuracy of the tagger and each corpus contains 250 untagged sentences. First corpus (A) contains 16% unknown words of the total words in the corpus, which means most of its words are existed in the lexicon. Second corpus (B) has 39% unknown words and third corpus (C) has 63% unknown words.

5.2. Experimental Results

The performance of the tagger is evaluated in terms of the number of correct tags it can recognize i.e recall. Recall is the percentage of correct POS tags predicted by the system. The following formula is used to get recall percentage.

$$Recall = \frac{\text{Correctly tagged words by the system}}{\text{Total no. of actually correct words}} * 100$$

Figure 2 depicts the experimental results of the POS tagger according to the three different testing corpora by using lexicalized and non-lexicalized approaches. In lexicalized approach, the combination of standard HMMs and lexicalized HMMs are combined for training and in non-lexicalized approach, only standard HMMs are applied to train the data.

As shown in the results, lexicalized models achieve higher accuracy than non-lexicalized models for all testing corpus.

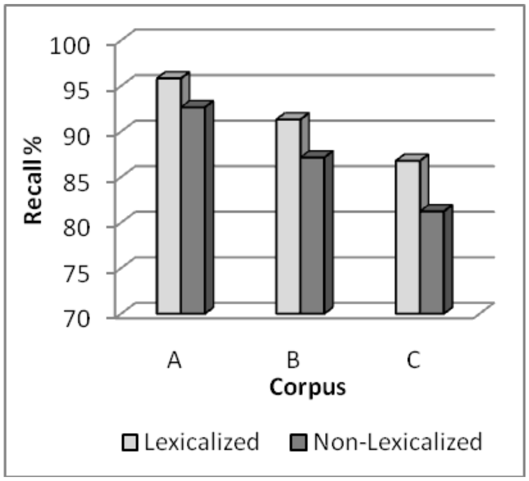


Figure 2. Experimental Results

5.3. Error Analysis

Some errors can occur for some words especially for negative word because unusual pattern of verbal negation is found in the patterns where the second verb of a compound is marked with the negative prefix, as in ne-ma-kaung: နေ-မ-ကောင်း: 'unwell', nar-ma-lal: နား-မ-လည် : 'misunderstand', etc. Although most of the negative verbs can be formed by combining a negative particle "မ-" (ma-) with a verb, some negative verbs can be formed from a combination pattern which has a negative particle "မ-" (ma-) between two words of verb.

To alleviate these errors, these words have to be tagged with their POS tags such as verb1 + negative particle + verb2 in the training corpus. It needs to pre-insert these peculiar pattern words and pre-tag with customized tags verb1, verb2, etc. In our training corpus, about 20 peculiar words are collected and pre-tagged. For unknown negative words solving, it is possible to tag noun to first word before negative particle and verb to next word of the negative particle. Actually, they should be tagged with verb1 and verb2 since the combination of verb1, negative particle and verb2 is one negative verb. Therefore, after this basic POS tagging, standard POS tagging has to be done for combining the basic tags to form the standard POS tags such as negative verb, negative adjective, superlative degree adjective, comparative degree adjective and so on.

Another error can appear when unknown proper nouns are found in the testing corpus. In our mother language, there are very wide ranges for giving the names. Therefore, naming database is needed to develop and now our lexicon stores many names for humans, places, rivers, mountains, months and so on.

6. Conclusion and Future Work

This paper proposes an implementation of bigram POS tagger using supervised learning approach for Myanmar Language. For disambiguating POS tags, lexicalized HMMs model is used for training and Viterbi algorithm is used for decoding. For the POS tagging, a Myanmar POS tagged corpus has to be used. The annotation standards for POS tagging include 12 tags for POS and 81 categories. “Myanmar Dictionary” and “Myanmar Grammar” books published by Myanmar Language Commission are used as references for POS tagging of Myanmar words. Myanmar lexicon is used for tagging a word with its all possible tags. Therefore, it is necessary to go through the lexicon manually and add all the possible tags that a word can have so that unknown words in the lexicon are reduced. And then, in order to develop larger pre-tagged corpus, untagged corpus has to be processed by this tagger and refined by manually checking errors. Then this corpus is ready to use for training phase so that our training data are greater in size and also the accuracy for the tagger.

For future work, we hope to conduct more experiments to examine how different types of input affect the performance. This tagger can be used in a number of NLP applications. In Myanmar to English machine translation system, Chunking, Grammatical Function Assignment, Word Sense Disambiguation, Translation Model and Reordering systems have to use these POS tags for analyzing Myanmar words in order to translate Myanmar text to English text.

References

[1] Antony, P. J. and Soman, K. P., "Parts Of Speech Tagging for Indian Languages: A Literature Survey", *International Journal of Computer Applications* (0975 – 8887) Volume 34– No.8, November 2011.

[2] Brill, E., "A simple rule based part of speech tagger", *Proceedings of the Third Conference on Applied 5atural Language Processing*, ACL, Trento, Italy, 1992.

[3] Brill, E., "Some advances in rule based part of speech tagging", *Proceedings of The Twelfth 5ational Conference on Artificial Intelligence* (AAAI- 94), Seattle, Washington, 1994.

[4] Brill, E., "Unsupervised Learning of Disambiguation Rules for Part of Speech Tagging", *Proceeding of The Natural Language Processing Using Very Large Corpora*, Boston, MA, 1997.

[5] Cutting, D., Kupiec, J., Pederson, J. and Sibun, P., "A practical Part-Of-Speech Tagger", *Proceedings of the Third Conference on Applied Natural Language Processing*, ACL, Trento, Italy, 1992.

[6] Fry, J., "Hidden Markov Models", San Jos´e State University, Computers and Spoken Language, Fall 2003.

[7] Fu, G.H., Luke, K.K., "Chinese unknown word identification as known word tagging", *Proceedings of the Third International Conference on Machine Learning and Cybernetics*, Shanghai, 26-29 August 2004.

[8] Hasan, F.M., UzZaman, N. and Khan, M., "Comparison of Unigram, Bigram, HMM and Brill's POS Tagging Approaches for some South Asian Languages", *Proceeding Conference on Language and Technology* (CLT07), Pakistan, 2007.

[9] Lee, S.Z., Tsujii, J.I. and Rim, H.C., "Lexicalized hidden Markov models for part-of-speech tagging", *Proceeding of The 18th Conference on Computational Linguistics* (COLING 2000), Saarbruken, Germany, pp. 481-487, July 2000.

[10] Murthy, K.N.,"Natural Language Processing - an Information Access Perspective", University of Hyderabad, 2006.

[11] Stolz, W.S., Tannenbaum, P.H. and Carstensen, F.V., "Stochastic Approach to the Grammatical Coding of English", Communications of the ACM 8: 399–405, 1965.

Appendix

Table 3. Part-of-Speech Tagset

No.	Description	Tag Name	Example
1.	Noun	**NN**	ခဲတံ
2.	Proper Noun	**NNP**	မောင်ဘ
3.	Pronoun	**PRN**	ကျွန်တော်
4.	Adjective	**JJ**	လှ
5.	Adverb	**RB**	အလွန်
6.	Verb	**VB**	ပြေး
7.	Conjunction	**CC**	သော်လည်း
8.	Particle	**Part**	သော ၊ ခဲ့
9.	Postpositional Marker	**PPM**	မှာ ၊ တွင် ၊ ဝယ် ၊ က ၊ သို့
10.	Interjection	**INJ**	လားလား
11.	Symbols	**SYM**	( ) / + = × ၊ ။
12.	Sentence Final	**SF**	သည် ၊ မည် ၊ ၏ ၊ ပြီ ၊ လား ၊ လဲ ၊ ပါ

Table 4. Specific Categories for each Part-of-Speech Tag

No.	POS Name	Category Name
1.	Noun	• Common • Person • Animals • Time • Body • Building • Object • Location • Cognition • Attribute

2.	Pronoun	• Person • Distplace • Disttime • Distobj • Question • Reflexive • Possessive
3.	Verb	• Common • Compound
4.	Adjective	• Demonstrative • Distplace • Disttime • Distobj • Quantity • Question
5.	Adverb	• Time • Manner • State • Quantity • Question
6.	Postpositional Marker	• Subject • Object • Leave • Direction • Arrive • Used • Cause • Accept • Place • Time • Agree • Extract • Possessive • Time Start • Time End
7.	Particles	• Type • Common • Number • Support • Interjection • Negative • Quantity • Example
8.	Conjunction	• Sentence • Mean • Chunk
9.	Interjection	• Common