2011 International Conference on Information Communication and Management
IPCSIT vol.16 (2011) © (2011) IACSIT Press, Singapore

Bigram Part-of-Speech Tagger for Myanmar Language

Phyu Hninn Myint, Tin Myat Htwe and Ni Lar Thein

University of Computer Studies, Yangon, Myanmar

Abstract. A variety of Natural Language Processing (NLP) tasks, such as machine translation, benefit from knowledge of the words syntactic categories or Part-of-Speech (POS). Since there is no state-of-the-art POS tagger for Myanmar language and POS tagging is a necessary process for Myanmar to English machine translation system, the development of a Bigram POS tagger is described in this paper. This tagger uses the customized POS tagset which is divided into two groups: basic and finer. Our Bigram tagger has two phases for disambiguating the basic POS tags: training with Hidden Markov Models (HMM) using Baum-Welch algorithm and decoding with Viterbi algorithm. Before disambiguation, word boundaries must be identified because words are not separated by spaces and there is no standard break among words in Myanmar Language. Therefore, the process for identifying each word has to be done in advance. After disambiguation, to produce the better output, normalization rules are used in order to form tagged words with the finer POS tags. This paper proposes an approach that will segment the input sentence to build meaningful words and tag these words with appropriate POS tags. By experiments, this approach has the best performance for known words with a few ambiguous tags. Experimental results show that our approach achieves high accuracy (over 90%) for different testing input.

Keywords: Natural language processing, Part of Speech Tagging, HMM, Baum-Welch, Viterbi

1. Introduction

Part-of speech (POS) tagging is a basic task in Natural Language Processing (NLP). It is the process of labelling a part of speech or other lexical class marker to each and every word in a sentence. It aims to determine which tag is the most likely lexical tag for a particular occurrence of a word in a sentence. It is a difficult problem by itself, since many words belong to more than one lexical class. While many words can be unambiguously associated with one POS tag, e.g. noun, verb or adjective, other words match multiple tags, depending on the context that they appear in [6]. POS tagger has to be applied to assign a single best POS to every word. A dictionary simply lists all possible grammatical categories for a given word. It does not tell us which word is used in which grammatical category in a given context. Hence, ambiguity resolution is the key challenge in tagging. 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.

In this paper, statistical tagging approach is used and it needs pre-tagged training corpus. Therefore, we have created a small corpus manually at first. Using it as a training corpus, a bigram POS-tagger has been built. To get larger-sized corpus, this tagger runs on an untagged corpus. After that, 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. For Myanmar language, word segmentation must be done before POS tagging, because there is no distinct boundary such as white space to separate different words.

The rest of the paper is organized as follows: Section 2 presents the methodology in detail. Section 3 describes the implementation methods. A brief description of the customized POS tagset is described in Section 4. Section 5 presents our training corpus. Finally, experimental results, some conclusions on this work and references are given in Section 6, Section 7 and Section 8 respectively.

2. Methodology

This paper presents about a system which accepts Myanmar sentences as input and its output is classified words with POS tags and categories. Procedure of the system has four steps; sentence level identification, word identification and basic POS tag with category tagging, disambiguation and defining the right tag, normalization and forming finer tag.

2.1. Sentence level identification

First of all, each sentence from the input text has to be extracted by recognizing the sentence marker " ။ " (pote ma) of Myanmar text.

2.2. Word identification and basic pos tag with category tagging

Secondly, each meaningful word must be identified and annotated with all possible basic POS tags and categories using Myanmar Lexicon. In order to form meaningful words, the system uses maximum matching to the input sentence. Myanmar words are comprised with one or more syllable. Maximum matching means comparing the whole input sentence with all words in the lexicon by each syllable from left to right. If one word exactly matches with the sentence, this word and its length are recorded. And then, the whole sentence is compared again until another word with longer length is found. If it is found, this word and its length are recorded again. This step is repeated again until no longer length word is found. When no more longer word is found, the lengths of the recorded words are compared and the word with longest length is extracted. It is also removed from the input sentence and one meaningful word is identified then. The rest of the input sentence has to be compared with all words in the lexicon again. And, words with maximum length are removed from the input sentence until no more syllables is left in it. If there is no matching one or more syllable, it is noted that unknown words and removed from the sentence. After that, the system annotates each word with all possible basic POS tags and categories from the lexicon. If the input word is unknown in the lexicon, it is annotated with all basic tags.

2.3. Disambiguation and defining the right tag

Thirdly, in order to disambiguate all possible tags to produce the right tag for each word, since supervised tagging method is used, Myanmar pre-tagged Corpus must be trained with HMM model using Baum-Welch algorithm. After that, Viterbi tagging algorithm has to be applied to find out the best probable path (best tag sequence) for a given word sequence. Ambiguous words have more than on POS tags in the lexicon. The sample ambiguous word “ သွား <tooth or go>” may be found with different POS tags and categories (VB.Common and NN.Body) in the corpus as the following sentences ::

- “သွား <go> (VB.Common) သည် ”
- “သွား <tooth> (NN.Body) တိုက် သည် ”

2.4. Normalization and forming finer tag

Finally, normalization step is needed to form more meaningful words and annotate with more appropriate finer POS tags and categories. In our language, Myanmar, there are many "Particles" in the text. These can be appeared in binding with Noun, Verb, Adjective and Adverb in the text. This may cause some changes in the type of POS tag, that is, Noun attached with some particles can become Verb or Adjective. Also, Verb or Adjective with some particles can create new POS tag, which is Adjective with superlative or comparative degree. There are the same pattern and particle to transform from one POS tag to another. Therefore, some lexical rules have to be developed to deduce more finer and standard POS tag.

3. Implementation Methods

This section presents the implementation on bigram based HMM tagging method. The intuition behind HMM and all stochastic taggers is a simple generalization of the "pick the most likely tag for this word" approach. A bigram is called a first-order Markov model and basic bigram model has one state for each word. Bigram taggers assign tags on the basis of sequences of two words. Therefore, the bigram tagger considers the probability of a word for a given tag and the surrounding tag context of that tag. For a given sentence or word sequence, HMM taggers choose the tag sequence that maximizes P (word | tag) * P (tag | previous tag).

3.1. Hidden markov model

Hidden Markov Models (HMMs) have been widely used in various NLP task to disambiguate Part Of Speech category. This is a probabilistic finite state machine having a set of sates (Q), an output alphabet (O), transition probabilities (A), output probabilities (B) and initial state probabilities (Π). $Q = \{q_1, q_2 \ldots q_n\}$ is the set of states and $O = \{o_1, o_2 \ldots o_3\}$ is the set of observations. $A = \{a_{ij} = P(q_j \text{ at } t+1 \mid q_i \text{ at } t)\}$, where P(a | b) is the conditional probability of a given b, $t \geq 1$ is time, and q_i belongs to Q. a_{ij} is the probability that the next state is q_j given that the current state is q_i. $B = \{b_{ik} = P(o_k \mid q_i)\}$, where o_k belongs to O. b_{jk} is the probability that the output is o_k given that the current state is q_i. $\Pi = \{p_i = P(q_i \text{ at } t=1)\}$ denotes the initial probability distribution over states.

Most common stochastic tagging technique states usually denote the POS tags. Probabilities are estimated from a tagged training corpus in order to compute the most likely POS tags for the word of an input sentence. The Markov model for tagging described above is known as a bigram tagger because it makes predictions based on the preceding tag, i.e. the basic unit considered is composed of two tags: the preceding tag and the current one.

3.2. Training with forward-backward (baum-welch) algorithm

Forward-Backward Algorithm is an Expectation Maximization (EM) algorithm invented by Leonard E. Baum and Lloyd R. Welch and capable of solving the Learning Problem. From a set of training samples, it can iteratively learn values for the parameters transition and emission probabilities of an HMM. It repeats until convergence while computing forward probabilities and backward probabilities, that is, re-estimation $\mathbf{P(w_i|t_i)}$ and $\mathbf{P(t_i|t_{i-1})}$.

3.3. Decoding with viterbi algorithm

The most likely sequence of tags given the observed sequence of words has to be found. The Markov assumption is used and problem is that of finding the most probable path through a tag-word lattice. The solution is Viterbi decoding or dynamic programming. HMM only produces output observations $O = (o_1, o_2, o_3 \ldots o_t)$. The precise sequence of states $S = (s_1, s_2, s_3 \ldots s_t)$ that led to those observations is hidden. We can estimate the most probable state sequence $S = (s_1, s_2, s_3 \ldots s_t)$ given the set of observations $O = (o_1, o_2, o_3 \ldots o_t)$. This process is called decoding. The Viterbi algorithm is a simple and efficient decoding technique. It is used to compute the most likely tag sequence. It means finding the best sequence of the maximum product of transition probability and emission probability.

3.4. Normalization rules

After disambiguation, lexical rules have to be created for finer POS tagging and, using these rules, finer and standard POS tags can be produced for some words. These finer tags are able to be applied in the later steps of NLP applications. It is possible that word with finer tag can be directly translated to other language. We have to analyze "Particles" which are functional words to develop most of the lexical rules.

In Myanmar language, there are many particles which can be called affixes of the word and can cause the changes of sense or type of that word. The prefixes are "မ-"(ma-), "အ"(a-) and "တ"(ta-). The prefix "မ-" (ma-) is an immediate constituent of the verb, which is the head of the word construction as in: ma-swa: မ-သွား : 'not go'; ma-kaung: မ-ကောင်း : 'not good'. It changes the positive sense to negative sense of the word. The scope of verbal negation extends to the whole compound of a compound verb, as in ma-tang pra: မ-တင်ပြ : : 'not submit'; ma-saung-ywat မ-ဆောင်ရွက် : 'not carry out'. Another pattern of negation is possible with verb compounds or verb phrases by individualized negation of each portion of the compound, as in: ma-ip ma-ne : မ-အိပ် မ-နေ : 'not sleep at all'; ma-tang ma-kya: မ-တင် မ-ကျ : 'noncommittal'.

The prefix "အ-" (a-) is a type converter which is the head word of the verb or adjective as in: a-lote: အ-လုပ် : 'work or job'; a-hla : အ-လှ : 'beauty'. The prefix "တ-" (ta-) can also be seen as a type converter, as in ta-lwal ta-chaw: တ-လွဲ တ-ချော် : 'wrongly'.

The postfixes are " -မှု " (-mhu), "- -ခြင်း " (-ching), "- -ချက် " (-chat), "- ရေး " (-yay), "- နည်း " (-nee), "- စွာ " (-swar), "- သော" (-thaw), " -သည့် " (-thi), "- မည့် " (-myi), etc. The postfixes " မှု " (-mhu), "- ခြင်း " (-ching), "- ချက် " (-chat), "- ရေး " (-yay), "- နည်း " (-nee) change the type of the previous POS tag from verb or adjective or adverb to noun. The words ended with these postfixes are in the noun form. Also, the postfixes "-သော " (-thaw), " သည့် " (-thi), "- မည့် " (-myi) convert to the adjective form from adjective or adverb or verb. The postfixes "- စွာ " (-swar) alters the type of adjective or verb or adverb to form adverb. In noun form, the postfixes "- များ " (-myar), "- တို့ " (-doh) change the singular noun to plural noun. Moreover, in adjective, if JJ tag is lied between two affixes "အ " (-a) and "- ဆုံး " (-sone), this tag JJ become to JJS (superlative degree), i.e., " အ JJ ဆုံး " is equal to "JJS". Sample normalization rules are depicted in figure 1.

JJ	->	JJ (သော | သည့် | မည့်)
JJS	->	အ JJ ဆုံး
JJC	->	(ပို၍ | သာ၍ | ပို) JJ
RB	->	RB စွာ
NNR	->	NN (များ | တို့)
NNR	->	PRN (များ | တို့)
NN	->	အ VB
NN	->	VB (မှု | ခြင်း | ချက် | ရေး | နည်း)
VB	->	VB.Common | VB.Compound

Fig. 1: Sample normalization rules.

The sample input text from the POS tagged corpus and output of the normalization step are shown in figure 2.

Before Normalization,

▪ " ကျန်းမာ <*health*> /**VB.Common** # ခြင်း <*-chin*> /**PART.Common** # သည် <*-thi*> /PPM.Subject # လာဘ် <*invaluable things*>/NN.Common # တစ် <*-one*> /PART.Number # ပါး <*-par*>/PART.Type # ဖြစ် <*is*> /VB.Common # သည် <*-thi*>/SF "

After Normalization,

▪ " ကျန်းမာခြင်း <*health*>/**NN** # သည် <*-thi*>/PPM.Subject # လာဘ် <*invaluable things*> /NN.Common # တစ် <*-one*> /PART.Number # ပါး <*-par*> /PART.Type # ဖြစ် <*is*>/VB.Common # သည် <*-thi*> /SF "

Meaning in English is "*Healthy is one of the invaluable things.*"

Fig. 2: Example for normalization.

4. Customized POS Tagset

The customized POS tagset of this tagger uses only 20 POS tags: 14 for basic tags and 6 for finer tags. To obtain more accurate lexical information together with POS tag, category of a word has to be added according to Myanmar grammar. This category can be applied in further NLP applications. The category for a word can be constructed from the features of that word. For instances of POS tag with category, " မိန်းကလေး " <girl> word must be tagged with NN.Person (Person category of Noun tag), " သို့ " <to> with PPM.Direction (Direction type of Postpositional Marker), " သူ" <he> with PRN.Person (Person type of Pronoun), "လှ "<beautiful> with JJ.Dem (Demonstrative sense of Adjective), " အလွန် " <very> with RB.State (State of Adverb) and so on. Moreover, as Myanmar sentences have some sentence final words and these are always used in the end of the sentences, we have classified these words in one class, SF (Sentence Final).

5. Training Corpus

In our training corpus, Myanmar words are segmented and tagged with their respective POS tags and categories. "#" is word break and "/" is put between word and its POS tag and category. Each sentence is ended with carriage return. We have limited resource for annotated corpus and lexicon till now. However, we have created a pre-tagged corpus with 1000 sentences for experiments. Figure 3 shows the sample corpus format.

▪ သူ <*he*> /PRN.Person # သည် <*-thi*> /PPM.Subject # စာ <*lesson*> /NN.Common # ကို <*-thi*> /PPM.Object # အလွန် <*very*> /RB.State # ကြိုးစား <*try*> /VB.Common # သည် <*-thi*> /SF

Meaning in English is "*He tries hard his lesson very much.*"

Fig. 3: Sample corpus format.

6. Experimental Results

In order to measure the performance of the system, we have tested many experiments using our approach on different untagged corpora till we get the best accuracy. The training corpus has 1000 Myanmar sentences and average sentence length is about 10 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, unknown words and ambiguous words for the tagger. "Known words" means the words including in the lexicon and "Unknown Words" means the words that are not pre-inserted in the lexicon. "Ambiguous words" means the known words which can be tagged with more than one POS tags and it is necessary to solve for disambiguating which tag is the particular tag for these words. Some ambiguous words have a few numbers of POS tags (around 5 tags) and some have many POS tags (up to 10 tags). For unknown words, the tagger has to annotate these words with all basic tags and has to disambiguate for all tags. There are 14 basic tags and 9 tags of these have specific categories (52 categories in total). One unknown word has to be tagged with all 57 tags. Disambiguating unknown words makes reduction in the accuracy of the tagger.

The performance of this tagger is evaluated in terms of precision, recall and F-measure. Precision (P) is the percentage of POS tags correctly predicted by the system. Recall (R) is the percentage of correct POS tags predicted by the system. F-score is the harmonic mean of recall and precision, that is, F=2PR/(P+R). Three testing corpora are used for evaluation in order to calculate the precision, recall and F-score of the tagger and each corpus contains 300 untagged sentences. First corpus (A) has "Known Words", but most of the words have a few numbers of ambiguous tags (around 5 tags). Second corpus (B) has "Known Words", but most of the words have many ambiguous tags (up to 10 tags). Third corpus (C) has "Unknown Words". Table 1 shows the experimental results of POS tagging according to our approach on different types of text.

Table 1: Experimental results

Testing Corpora	Precision (%)	Recall (%)	F-score (%)
A	97.16	98.78	97.96
B	95.62	95.93	95.77
C	93.89	93.92	93.91

7. Conclusions

This paper proposes an implementation of bigram POS tagger using supervised learning approach for Myanmar Language. For disambiguating POS tags, HMM model with Baum-Welch algorithm is used for training and Viterbi algorithm is used for decoding. And then, lexical rules have to be applied to normalize some words and tags in order to produce accurate and finer tags. For the POS tagging, a Myanmar POS tagged corpus has to be used. The annotation standards for POS tagging include 20 tags for POS and many categories. "Myanmar Dictionary" and "Myanmar Grammar" books published by Myanmar Language Commission are used as references for POS tagging of Myanmar words. One of the improvements to be done is adding more lexical rules in order to do more accurate normalization. Also, Myanmar lexicon is used for tagging a word with its all possible tags. Therefore, another that is necessary here is to go through the lexicon manually and add all the possible tags that a word can take so that unknown words in the lexicon are reduced. And then, in order to develop larger pre-tagged corpus size, 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 accuracy for our 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.

8. References

[1] Anwar, W., Wang, X., LuLi and Wang, X., Hidden Markov Model Based Part of Speech Tagger for Urdu, *Information Technology Journal*, 2007.

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

[3] Dandapat, S., Sarkar, S. and Basu , A., A Hybrid Model for Part-of-Speech Tagging and its Application to Bengali, *Transactions on engineering, computing and technology v1*, December 2004 ISSN 1305-5313.

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

[5] Jurafsky, D. and Martin, JH., Tagging with Hidden Markov Models. Viterbi Algorithm. Forward-backward algorithm, http://www.cse.unt.edu/~rada/CSCE5290/Lectures/

[6] Manning, CD. and Schütze, H., *Foundations of Statistical Natural Language Processing*, Cambridge, Mass, 1999.