Assigning automatically Part-of-Speech tags to build tagged corpus for Myanmar language

Phyu Hninn Myint
University of Computer Studies, Yangon.
phyuhm@gmail.com

Abstract

A variety of Natural Language Processing (NLP) tasks, such as named entity recognition, stemming, question answering and machine translation, benefit from knowledge of the words syntactic categories or Part-of-Speech (POS). POS taggers must be successfully applied to assign a single best POS to every word in a corpus.This paper presents to develop Part-of-Speech tagged text corpora by employing Bigram part-of-speech tagger. POS tagging is a process of assigning appropriate syntactic categories to each word in a sentence. As applying bigram model for automated tagging process we have provided an adequate annotated corpus from scratch. We have used customized POS tagset to annotate the words in a Myanmar sentence. Our Bigram tagger has two phases: training with Hidden Markov Models (HMM) and decoding with Viterbi algorithm.

Keywords: Natural Language Processing, Part-of-Speech Tagging, Hidden Markov Models and Viterbi

1. Introduction

POS tagged corpus is a processed textual database that serves as a reference material for further research in NLP as well as a learning database for machine translation algorithms and other software applications. Building syntactically classified corpus requires a sequence of procedures such as text preprocessing, tokenizing sentences and POS tagging. Also it is influenced on all areas of NLP such as information retrieval, text-to-speech, parsing, information extraction and any linguistic research for corpora [1].

While many words can be unambiguously associated with one POS or tag, other words match multiple tags, depending on the context that they appear in [9]. Therefore, the accuracy of a tagger depends on its learning database or its training data. The larger the corpus size, the better the accuracy for tagging. Also, an automatic part-of-speech tagger is necessarily requested a large corpus because hand annotating is complicated task and also assigning POS tags to each word is very time consuming [8].

In this paper, we start by hand annotating raw text to build a tagged corpus. Then we process by preparing training data from the manually tagged corpus. Next, we automatically assign POS tags to each word of raw text using Bigram part-of-speech tagger. Then, we analyze result of tagged text and refine manually. We conclude with the result that POS tagged corpus for Myanmar Language is annotated by stochastic method of POS tagging.

2. Literature Review

Different approaches have been used for Part-of-Speech (POS) tagging, where the notable ones are rule-based, stochastic, or transformation-based learning approaches. Rule-based taggers [5,6] try 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. The stochastic (probabilistic) approach [3,15] uses 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.

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

[15]. 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. Finally, 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 [5,6]. All of the approaches discussed above fall under the rubric of supervised POS Tagging, where a pre-tagged corpus is a prerequisite. On the other hand, there is the unsupervised POS tagging [7,10,12] technique, and it does not require any pre-tagged corpora.

3. Motivations

Part-of-Speech tagged corpora are essential for developing state-of-the-art POS Tagger for our mother language, Myanmar. To analyze Myanmar Language for Myanmar to English Machine Translation System, assigning POS tags to every token in the text is indispensable as a basic processing step. The capability for a computer to automatically tag a sentence is very essential for further analysis in many approaches to the field of NLP.

4. Methodology

There are several steps to create tagged corpus using stochastic method. The following list demonstrates steps needed corpus building.

- Collecting raw text
- Hand-annotating and preparing training data
- Assigning POS tag automatically
- Analyzing and refining POS tags

At the first, we collect and normalize raw text from online journals, newspaper and e-books. Next, we assign tags in un-annotated text manually and have training data for statistical method.

When we have appropriate amount of training data, decoding phase can be processed. In this phase, untagged text are inputted and then assigned all possible tags from tagged corpus we have. Then, Viterbi algorithm is used to disambiguate all tags and assign the right tag to each word.

After generating tagged text, we have to analyze and refine manually to unknown tag and wrong tag. Finally, we can use these correct texts in the corpus so that our corpus size can be larger.

The system architecture is demonstrated in the following figure, Figure 1.

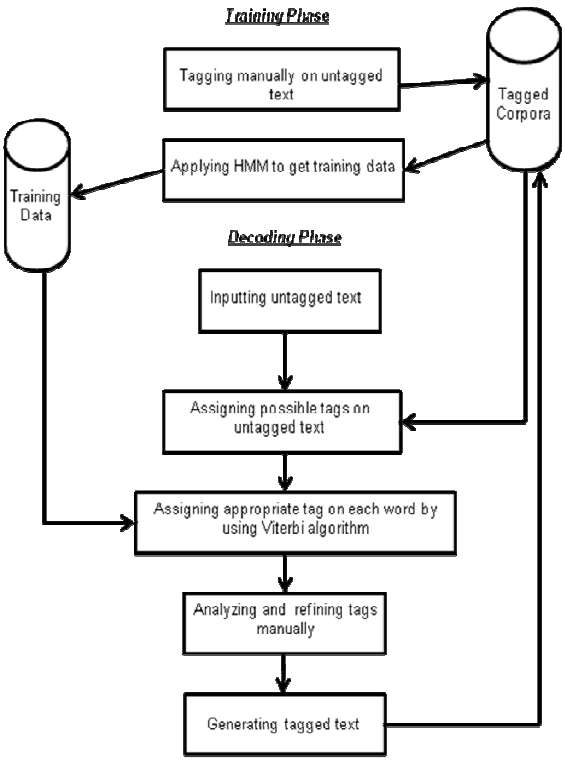

Figure 1. System Architecture

5. Collecting raw text

A great deal of raw text must be assembled from a variety of sources. Also raw text is checked morphological and syntactic errors and normalized to be ready to annotate.

In case of this work, bunch of raw text are collected from online journals, newspaper and e-books. Myanmar text are copied and saved in text files. Moreover, we have to make changes in these files in level of headings, paragraphs and others.

6. Manual tagging

After text normalization, a subsequent task is assigning POS tag to each word by hand. Furthermore, when we get a tagged corpus, we apply useful functions such as counting word frequency, searching words by tag, creating words involved particular tag and preparing training data. These functions help us to analyze on tagged corpus.

If the number of tags is very large, it leads to increased complexity during POS tagging which in turn reduces the tagging accuracy. For simple POS level, a tagset which has just the grammatical categories excluding grammatical features is needed.

Tag Name	Description
NN	Singular Noun
NNR	Plural Noun
NNG	Gerund
NNP	Proper Noun
PRN	Pronoun
PRNS	Possessive Pronoun
JJ	Adjective
JJR	Comparative Adjective
JJS	Superlative Adjective
RB	Adverb
VB	Verb
VBD	Past Tense Verb
VBN	Past Participle Verb
VBG	Continuous Tense Verb
CC	Conjunction
PART	Particle
PPM	Postpositional Marker
INJ	Interjection
CRD	Cardinal Number
ORD	Ordinal Number
SF	Sentence Final
UNK	Unknown

Figure 2. Customized POS Tagset

Figure 2 shows the customized tagset we have used to annotate the words.

7. Preparing training data

To get training data, we have to compute bigram probabilities for each tag in the tagged corpus. Since we have developed a model, it produces two results. The results of the training phase are word-and-tag probabilities and bigram of POS tag shown in Figure 3 and 4.

Figure 3 shows the word and tag probability. For example, the word "*သည် (the)*" is appeared in the corpus as SF tag and PART tag. The word tag probability is calculated by dividing the number of word with one tag by the total number of that word containing in the corpus.

သည် (the) SF : 0.9565 PPM : 0.0435
ဏီ (ei) PPM : 1.0

Figure 3. Word and tag probability

Figure 4 shows the tag and tag probability. For example, the tag "*PRN*" is paired with the tags "*NN, VB* and *PPT"* in the corpus. The tag bigram probability is calculated by dividing the number of one tag paired with another tag by the total number of that tag containing in the corpus.

NN PART : 0.6176
VB PART : 0. 5477
VB SF : 0.8765
PRN PPM : 0.9445

Figure 4. POS tag bigram probability

The example sentence for the word "*သည် (the)*" in the corpus is shown in Figure 5.

သူ _ PRN # သည် (the)_ PPM # စာ _ NN # ကို _ PPM # အလွန် _ RB # ကြိုးစား _ VB # သည် (the) _ SF

Figure 5. Example sentence in the corpus

8. Building part-of-speech tagged corpus

The next step for corpus tagging aims to assign POS tag automatically to unprocessed text files in raw corpus. Bigram POS tagger runs and assigns POS tag to each word automatically on the untagged corpus.

8.1. Implementation on Bigram part-of-speech tagging

This section presents the implementation on bigram based HMM tagging method. The intuition behind HMM (Hidden Markov Model) 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 the following formula:

P (word | tag) * P (tag | previous tag)

HHM is the most common stochastic tagging technique. Probabilities are estimated from a tagged training corpus in order to compute the most likely POS tags for the word of an input sentence.

States usually denote the POS tags. Let T be the set of states (POS Tags), W be the set of output alphabets (words), A be the transition probabilities, B be the emission probabilities and π be the set of initial state probabilities.

$$T = \{ t_1, t_2, \ldots, t_n \}$$
$$W = \{ w_1, w_2, \ldots, w_n \}$$

The transition probability is calculated simply by the following formula.

$P(t_i|t_{i-1}) = C(t_{i-1},t_i) / Total\ number\ of\ bigrams\ starts\ with\ t_{i-1}$

where t_i is the current tag and t_{i-1} is the previous tag and $C(t_{i-1},t_i)$ is a function that counts the number of times that t_{i-1},t_i pair is found.

For calculating emission probability, the unigram of a word is calculated along with its tag assigned in the tagged data. The emission probability of a word given a particular tag is calculated simply by the following formula.

$P(t|w) = C(t,w) / Total\ number\ of\ unigrams\ starts\ with\ w$

where t is the tag and w is the word tagged with t and C(t,w) is a function that counts the number of times that a word(w) tagged with t is found.

HMM only produces output observations o = (o_1, o_2, o_3 . . . o_t). The precise sequence of states s = (s_1, s_2, s_3 . . . s_t) that led to those observations is hidden. We can estimate the most probable state sequence s = (s_1, s_2, s_3 . . .s_t) given the set of observations o = (o_1, o_2, o_3 . . . o_t). This process is called decoding. The Viterbi algorithm is a simple and efficient decoding technique. 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 [10].

for i from 1 to n:

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

Viterbi decoding algorithm is as follows:

/ Given an observation sequence o[t], a transition matrix a[i,j], and an observation likelihood b[i,o[t]], create a path probability matrix v[i,t].*/*

```
v[0,0] = 1.0
for t = 0 to T do
  for s = 0 to num_states do
    for each transition i from s do
        new_score = v[s,t] * a[s,i] * b(i,o[t])
        if (new_score > v[i,t+1]) then
            v[i,t+1] = new_score
            back_pointer[i,t+1] = s
```

*/*To find best path, choose the highest probability state in the final column of v[] and backtrack. Each time v[i,t+1] is gets a higher score from state s, back pointer[i,t+1] is reset to s, back pointer[s,t] therefore only stores optimal paths*/*

According to above algorithm, we have developed bigram POS tagger in JAVA and it supports Unicode (UTF-8).

As a result of POS tagger, each probability of tag pairs for choosing most appropriate tag must be disambiguated.

In Figure 6, an example of an input sentence is shown as follows.

မောင်မောင် # သည် # A တန်း # ထဲတွင် # A တော်ဆုံး # ကျောင်းသား # တစ် # ယောက် # ဖြစ် # သည် #

Figure 6. Input sentence

မောင်မောင် _ NNP # သည် _ PPM # အတန်း _NN # ထဲတွင် _ PPM # အတော်ဆုံး _ JJS # ကျောင်းသား _ NN # တစ် _ CD # ယောက် _ PART # ဖြစ် _ VB # သည် _SF #

Figure 7.Output POS tagged sentence

Here Figure 7 shows the output that our POS tagging model produces. The output of tagger generator is manually corrected to increase the corpus size.

9. Analyzing and refining POS tags

After building a tagged corpus, we have to find some errors by analyzing on tagged outputs and re-annotating with the right tags. 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.

10. Conclusion and future work

In this paper, we presented an automatic approach for building part-of-speech tagged corpus.

Since only bigram POS tagger is not good approach [2], we need to find out more experiment on stochastic models such as Maximum Likelihood, Baum-Welch algorithms. The next step could be to comparison on different stochastic approaches and to apply best one method for Myanmar Language corpus tagging. Furthermore, we are going to develop chunker which is subdivision of sentences into clusters – called chunks and find out lexical and contextual rules by analyzing tagged corpus. The customized tagset can be added to be useful in further NLP applications such as word sense disambiguation and function tagging. Also lexical information such as word category can be supplied in every tag.

11. References

[1] Amarsanaa Ganbold and Purev Jaimai, *Integrative tools for part-of-speech tagged corpus*, School of IT, National University of Mongolia, Ulaanbaatar, Mongolia.

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

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

[4] D Jurafsky and JH Martin, *Speech and Language Processing*, Upper Saddle River NJ (2000): Prentice Hall.

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

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

[7] E. Brill, *Unsupervised Learning of Disambiguation Rules for Part of Speech Tagging*, In Proceeding of The 5atural Language Processing Using Very Large Corpora, Boston, MA, 1997.

[8] Fahim Muhammad Hasan, Naushad UzZaman and Mumit Khan, *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

[9] Jana Diesner, *Part of Speech Tagging for English Text Data*, School of Computer Science, Carnegie Mellon University, Pittsburgh.

[10] John Fry, *Hidden Markov Models*, San Jos´e State University, Computers and Spoken Language, Fall 2003.

[11] Jurafsky & Martin, *Tagging with Hidden Markov Models. Viterbi Algorithm. Forward-backward algorithm.*

[12] M. Pop, *Unsupervised Part-of-speech Tagging*, Department of Computer Science, Johns Hopkins University, 1996.

[13] R Dale, H Moisl & H Somers, *Handbook of Natural Language Processing*, New York (2000).

[14] R. Prins and G. van Noord, *Unsupervised Pos-Tagging Improves Parsing Accuracy And Parsing Efficiency*, In Proceedings of the International Workshop on Parsing Technologies, 2001.

[15] Stolz, W.S., Tannenbaum, P.H. & Carstensen, F.V. , *Stochastic Approach to the Grammatical Coding of English*, Communications of the ACM.

[16] Steve Renals, *Part-of-speech tagging*, ICL, 2006.

[17] Waqas Anwar, Xuan Wang, LuLi and Xiaolong Wang, *Hidden Markov Model Based Part of Speech Tagger for Urdu*, Information Technology Journal, 2007.