

A hybrid model for Part-of-Speech tagging and its application to Bengali

Sandipan Dandapat, Sudeshna Sarkar, Anupam Basu

Abstract— This paper describes our work on Bengali Part of Speech (POS) tagging using a corpus-based approach. There are several approaches for part of speech tagging. This paper deals with a model that uses a combination of supervised and unsupervised learning using a Hidden Markov Model (HMM). We make use of small tagged corpus and a large untagged corpus. We also make use of Morphological Analyzer. Bengali is a highly ambiguous and relatively free word order language. We have obtained an overall accuracy of 95%.

Keywords—About four key words or phrases in alphabetical order, separated by commas.

I. INTRODUCTION

Part-of-Speech (POS) tagging is a technique for automatic annotation of lexical categories. Part-of-Speech tagging assigns an appropriate part of speech tag for each word in a sentence of a language. POS tagging is widely used for linguistic text analysis. Part-of-speech tagging is an essential task for all the natural language processing activities. A POS tagger takes a sentence as input and assigns a unique part of speech tag to each lexical item of the sentence. POS tagging is used as an early stage of linguistic text analysis in many applications including subcategory acquisition; text to speech synthesis; and alignment of parallel corpora. There are a variety of techniques for POS tagging. Two approaches to POS tagging are

1. Supervised POS Tagging
2. Unsupervised POS Tagging

Supervised tagging technique requires a pre tagged corpora where as unsupervised tagging technique do not require a pre tagged corpora. Both supervise and unsupervised tagging can

Manuscript submitted on November 04, 2004.

Sandipan Dandapat, Indian Institute of Technology – Kharagpur, West Bengal, India; e-mail: sandipan_242@yahoo.com

Prof. Sudeshna Sarkar, Dept. of Computer Sc. and Engg., Indian Institute of Technology-Kharagpur

Prof. Anupam Basu, Dept. of Computer Sc. and Engg., Indian Institute of Technology-Kharagpur

be of two types Rule based and stochastic.

Rule based system needs context rule for POS tagging. Typical rule based approaches use contextual information to assign tags to unknown or ambiguous words. These rules are often known as *context frame rules*.

Stochastic tagging technique makes use of a corpus. The most common stochastic tagging technique uses a Hidden Markov Model (HMM). The states usually denote the POS tags. The probabilities are estimated from a tagged training corpus or an untagged corpus in order to compute the most likely POS tags for the word of an input sentence. Stochastic tagging techniques can be of two types depending on the training data. Supervised stochastic tagging techniques use only tagged data. However the supervised method requires large amount of tagged data so that high level of accuracy can be achieved. Unsupervised stochastic techniques, on the other hand, are those which do not require a pre-tagged corpus but instead use sophisticated computational methods to automatically induce word groupings (i.e. tag sets), and based on these automatic groupings, they calculate the probabilistic values needed by stochastic taggers.

Our approach is a combination of both supervised and unsupervised stochastic techniques for training a HMM. We are using a Morphological Analyzer for Bengali in our POS tagging technique. The Morphological Analyzer takes a word as input and gives all possible POS tags for the word.

II. LINGUISTIC CHARACTERISTICS OF BENGALI

Present day Bengali has two literary styles. One is called "Sadhubhasa" (elegant language) and the other "Chalitbhasa" (current language). The former is the traditional literary style based on Middle Bengali of the sixteenth century. The later is practically a creation of the present century, and is based on the cultivated form of the dialect spoken in Kolkata by the educated people originally coming from districts bordering on the lower reaches of the Hoogly. Our POS tagger deals with Chalitbhasa.

Bengali is a relatively free word order language compare with European languages. For example:

Consider the simple English sentence

I eat rice ? PRP VB NN

The possible Bengali equivalents of the above English sentence are

Ami bhAta khAi (I rice eat) ? PRP NN VB
Ami khAi bhAta (I eat rice) ? PRP VB NN
bhAta Ami khAi (Rice I eat) ? NN PRP VB
bhAta khAi Ami (Rice eat I) ? NN VB PRP
khAi Ami bhAta (Eat I rice) ? VB PRP NN
khAi bhAta Ami (Eat rice I) ? VB NN PRP

Part of speech tagging using linguistic rules is a difficult problem for such a free word order language. A HMM model can capture the language model from the perspective of POS tagging.

We are considering 40 different tags for POS tagging. POS tagger is the most essential tool for design and development of Natural Language Processing application. A major problem of NLP is word sense disambiguation. A larger tag set reduces the ambiguity problem but it also reduces the parsing complexity. An important task in natural language processing is parsing. Given a POS tagged sentence, local word groups are easier to identify if we have a large number of tags. A large tag set also facilitates shallow parsing. Our goal is to achieve high accuracy using a large tag set.

III. BACKGROUND WORK

There are different approaches have been used for Part-of-speech tagging. Some previous work has focused on rule based linguistically motivated Part-of-Speech tagging worked by Brill (1992, 1994) [1]. Brill's tagger uses a two-stage architecture. The input tokens are initially tagged with their most likely tags. It employs an automatically acquired set of lexical rules to identify unknown words. TNT is a stochastic HMM tagger which uses a suffix analysis technique to estimate lexical probabilities for unknown tokens based on properties of words in the training corpus which share the same suffix.

Recent stochastic methods achieve high accuracy in part-of-speech tagging tasks. They resolve the ambiguity on the basis of the most likely interpretation. Markov model has been widely used to disambiguate part-of-speech category. There have been two types of work – one using tagged corpus and other using untagged corpus.

The first model uses a pre-tagged corpus. A bootstrapping method for training was designed by Deroult and Merialdo [Deroult and Merialdo, 1986] [2]. In this model they used a small pre-tagged corpus to determine the initial model. This initial model is used to tag more text. The tags are manually corrected to retrain the model. Church used Brown corpus to estimate the probabilities [Church, 1988] [3]. Existing methods assume a large annotated corpus and/or a dictionary. It is often the case that we have no annotated corpus or a small corpus at the time of developing a part-of speech tagger for new language.

The second model uses an untagged corpus. Supervised methods are not always applicable when a large annotated corpus is not available. There have been several works that

have used unsupervised learning to learn a HMM model for POS tagging. Baum-Welch algorithm [Baum, 1972] [4] can be used to learn a HMM from un-annotated data. The maximum entropy model is powerful enough to achieve accuracy in tagging task [Ratnaparkhi, 1996] [5]. It uses a rich feature representation and generates a tag probability distribution for each word.

[Cutting et al., 1992] [6] used a Hidden Markov Model for Part of speech tagging. The HMM model use a lexicon and an untagged corpus. The methodology uses a lexicon and some untagged text for accurate and robust tagging. There are three modules in this system – tokenizer, training and tagging. Tokenizer identifies an ambiguity class (set of tags) for each word. The training module takes a sequence of ambiguity classes as input. It uses Baum-Welch algorithm to produce a trained HMM. Training is performed on a large corpus. The tagging module buffers sequence of ambiguity classes between sentence boundaries. These sequence are disambiguated by computing the maximal path through the HMM with the Viterbi algorithm. In our POS tagging for Bengali we are using Baum-Welch algorithm for learning from an untagged corpus. But instead of learning completely from the untagged data we are also using a tagged data to determine the initial HMM model. Like Cutting we are also taking help of ambiguity class. But our ambiguity class is taken from the Morphological Analyzer. Instead of using ambiguity class both at the time of learning and decoding we are using the ambiguity class only at the time of decoding.

Another model is designed for the tagging task by combining unsupervised Hidden Markov Model with maximum entropy [kazama et al, 2001] [7]. The methodology uses unsupervised learning of an HMM and a maximum entropy model. Training an HMM is done by Baum-Welch algorithm with an un-annotated corpus. It uses 320 states for the initial HMM model. These HMM parameters are used as the features of Maximum Entropy model. The system uses a small annotated corpus to assign actual tag corresponds each state.

IV. HIDDEN MARKOV MODELING

Hidden Markov Models (HMMs) have been widely used in various NLP task. Hidden Markov Model is a probabilistic finite state machine having a set of states (Q), an output alphabet (O), transition probabilities (A), output probabilities (B) and initial state probabilities (?).

$Q = \{q_1, q_2, \dots, q_n\}$ is the set of states and $O = \{o_1, o_2, \dots, o_3\}$ is the set of observations.

$A = \{a_{ij} = P(q_j \text{ at } t+1 \mid q_i \text{ at } t)\}$, where $P(a \mid b)$ is the conditional probability of a given b , $t = 1$ 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 .

$B = \{b_{ik} = P(o_k | q_i)\}$, where o_k belongs to O . b_{ik} 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.

In our HMM model, states correspond to part-of-speech tags and observations correspond to words. We aim to learn the parameter of the HMM using our corpus. The HMM will be used to assign the most probable tag to the word of an input sentence. We use a bi-gram model. We tried supervised learning from the tagged corpus. But, possibly because the corpus size is so small we have achieved accuracy of 65%. Therefore we decide to use a raw corpus in addition to the tagged corpus.

The HMM probabilities are updated using both tagged as well as the untagged corpus. For the tagged corpus, sampling is used to update the probabilities. When using untagged corpus the EM algorithm is used to update the probabilities.

V. A HYBRID TAGGING MODEL

We will first outline our training method. The training module is based on partially supervised learning. It makes use of some tagged data and more untagged data. We are estimating the transition and emission probabilities from the partially supervised learning.

A. Training

In training module we use both types of sentences – tagged and untagged.

Tagged Data: Five hundred tagged sentences for supervised learning.

Untagged Data: Raw data for re-estimating parameter (50,000 words)

First we describe how we learn using tagged data and then we will outline the learning process from untagged data.

Our algorithm runs on a number of iterations. First we process the tagged data by supervised learning then in each iteration it processes the untagged data and updates the transition probabilities i.e. $p(\text{tag} | \text{previous tag})$ and emission probabilities i.e. $p(\text{word} | \text{tag})$ for the Hidden Markov Model. Using tagged data each word maps to one state as the correct part-of-speech is known. But using untagged data each word will map to all states because part-of-speech tags are not known i.e. all states we considered possible. In supervised learning, we calculate the bi-gram counts of a particular tag given a previous tag from the tagged corpus.

We use untagged data (50,000 words) to re-estimate the bi-gram counts from tag to tag and also re-estimate the unigram counts of a word given a particular tag. This re-estimation of

counts from untagged data is achieved using the Baum-Welch algorithm. In each iteration of the Baum-Welch algorithm we get some expected counts and add them to the previous counts. For the first iteration previous counts are actually the counts from the tagged data. In the second iteration the previous counts are the counts after first iteration. Finally Baum-Welch algorithm ends up by holding training plus raw counts. We use of ten iterations of the algorithm for modifying the initial counts estimated from tagged data.

We calculate the transition probabilities ‘A’ and emission probabilities ‘B’ from the above counts. We calculate the transition probability of next state given the current state. The transition probability is calculated simply by the following formula.

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

Where t_i is the current tag and t_{i-1} is the previous tag.

For calculating emission probability we calculate the unigram of a word along with its tag assigned in the tagged data. We are also calculating the emission probability of a word given a particular tag by using the above formula where t_i is the tag and t_{i-1} is the word. We are also using add one smoothing for avoiding zero transition and emission probabilities.

B. Decoding

The decoding module finds the best probable tag sequence of a sentence. We use Viterbi algorithm to calculate the best probable path (best tag sequence) for a given word sequence (sentence). Instead of considering all possible tags for each word in the test data we consider the most possible tags given by the Morphological Analyzer. We feed each word to our Morphological Analyzer that outputs all possible part-of-speech of that word. Considering all possible tags from the tagset increases the number of paths. But the use of Morphological Analyzer reduces the number of paths as given in following figure. For example we are considering a sentence “Ami ekhana chA khete yAba”.

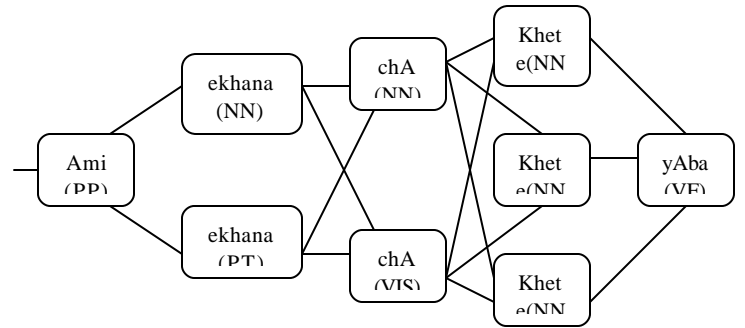


Figure 1: Possible tags are taken from Morphological Analyzer

A word is unknown to the HMM if it has not occurred during the training. However even for an unknown word the Morphological Analyzer gives all possible tags of the word. These possible part-of-speech tags are used during training. In fig.1, each word has different possible tags given by Morphological Analyzer. For example word *chA* has two different tags NN and VIS. Using the above restriction on tags for each word and the transition and emission probabilities from a partially supervised model we are finding the best probable path (best tag sequence) for a given word sequence is found out by using the Viterbi tagging algorithm. The best probable path is calculated by the following formula.

$$\text{argmax} = \underset{i=1}{n} p(t_i | t_{i-1}) p(w_i | t_i)$$

This approach offers an overall high accuracy even if a small set of tagged corpus is used for the purpose.

VI. EXPERIMENT RESULTS

The system performance is evaluated in two ways. Firstly, the system is tested in one Leave One Out Correctness Validation (LOOCV) method i.e. from N tagged files we use N-1 for training and 1 file for testing. This is done for each individual file from N tagged files. The above technique for evaluation is applied on three approaches to determine the precision. In our POS tagging evaluation we use 20 files each consist of 25 sentences.

$$\text{precision} = \frac{\text{Correctly tagged words by the system}}{\text{Total no. of words in the evaluation set}}$$

We have tested three different approaches of POS tagging.

Method 1: POS tagging using only supervised learning

Method 2: POS tagging using a partially supervised learning and decoding the best tag sequence without using Morphological Analyzer restriction.

Method 3: POS tagging using a partially supervised learning and decoding the best tag sequence without using Morphological Analyzer restriction.

The evaluation results are given in the following table:

Precision	Method 1	Method 2	Method 3
	64.31	67.6	96.28

The above table indicates the high 96.28% accuracy of the Hybrid system. To ensure the correctness of the precision we tried another approach for evaluating the system. We took 100 sentences (1003 words) randomly from the CIIL corpus and tagged it manually; the sentences taken from CIIL corpus being more complex sentences compare to the sentences used in tagged data. The precision is calculated using the above formula.

Precision	Method 1	Method 2	Method 3
	59.93	61.79	84.37

In the above data set the precision is much lower. Many errors are due to incomplete lexicon used in our Morphological Analyzer and also the unavailability a proper noun identifier. Morphological errors are of two types – a particular word is not found in the Morphological Analyzer or Morphological Analyzer does not cover all possible tags of a word. To find out the actual accuracy of our model we manually entered the possible part-of-speech for all the words of the test set that are not covered by the Morphological Analyzer. We also made a list of all possible proper nouns in our test data set. At the time of evaluation we marked all proper nouns from that list. We tested the above modification over Method 3 and we got an average percentage of precision 95.18%

Precision	Method 3
	95.18

VII. CONCLUSION AND FUTURE WORK

This paper presents a model for POS tagging for a relatively free word order language, Bengali. On the basis of our preliminary experiment the system is found to have an accuracy of 95%. The system uses a small set of tagged sentences. It also uses an untagged corpus and a morphological analyzer. The precision is affected by incomplete lexicon in Morphological Analyzer and errors in the untagged corpus. It is expected that system accuracy will increase by correcting the typographical errors in the untagged corpus and also by increasing the accuracy of Morphological analyzer. Some rule-based component can also be applied to the model to detect and correct the existing errors. The POS tagger is useful for chunking, clause boundary identification and other NLP applications.

REFERENCES

- [1] E. Brill, "A simple Rule-Based Part-of-Speech Tagger", University of Pennsylvania, 1992.
- [2] A. M. Deroualt and B. Merialdo, "Natural Language modeling for phoneme-to-text transposition", IEEE transactions on Pattern Analysis and Machine Intelligence, 1986.
- [3] K.W. Church, "A statistical parts program and noun phrase parser for unrestricted text", Proceedings of the second conference on Applied Natural Language Processing (ACL), 1988.
- [4] L. E. Baum, "An inequality and associated maximization technique in statistical estimation on probabilistic functions of a Markov process", Inequalities, 1972.
- [5] A. Ratnaparkhi, "A maximum entropy Part-of-speech tagger", Proceedings of the Empirical Methods in NLP conference, University of Pennsylvania, 1996.
- [6] D. Cutting, "A practical part-of-speech tagger", Proceedings of third conference on Applied Natural Language processing, 1992.
- [7] J. Kazama, "A maximum entropy tagger with unsupervised Hidden Markov Model", NLPRS, 2001
- [8] J. Allen, "Natural Language Understanding", pages {195-203}
- [9] D. Jurafsky and J. H. Martin, "Speech and Language Processing" pages {287-320}, Pearson Edition.