CHAPTER 3

Part of Speech Tagging

This chapter presents an overview of part of speech tagging, its different paradigms and standard approaches. It also presents some applications of part of speech tagging in the field of computational linguistics.

3.1 Introduction

The process of marking up morphosyntactic categories of each lexical item including punctuation mark in a text document according to the context is called part of speech tagging. It is the same process as tokenization of computer languages, although tags for natural languages are much more ambiguous. Part of speech tagging is the common form of corpus annotation. It plays an increasingly important role in speech recognition, information retrieval, text to speech, linguistics research for corpora and higher level NLP tasks like parsing, machine translation and many more.

Ambiguity is the nature of natural languages and it appears in various levels of natural language processing task. There are numbers of words or lexical items with multiple tags. The correct tag depends on the context of the sentence. Consider the following English and Manipuri sentence.

1. Book a room on the top floor.

2. অচৌবা গী য়ুম অচৌবা অমা লৈ।

achouba gi yum achouba ama lei.

Achouba has a big house.

There are lots of ambiguity in the sentences given above which should be resolved after understanding the sentences. For instance in example (1), the word ‘book’ can be a noun or a verb and the word ‘top’ can either be an adjective or a noun. Similarly, in example (2), the word ‘achouba’ can either be a proper noun or an adjective. The meaning of the word ‘achouba’ is ‘big’ in English but it can also be a typical Manipuri name of a male person. In general, POS ambiguity can be resolved by examining the context of the surrounding words. Figure 3.1 shows a detailed analysis of the POS ambiguity of an English sentence considering only the basic 8 tags. The box with single line indicates the correct tag for a particular word where no ambiguity exists i.e. only one tag is possible for the word. On the contrary, the boxes with the double line indicate the correct POS tag of a word from a set of possible tags.

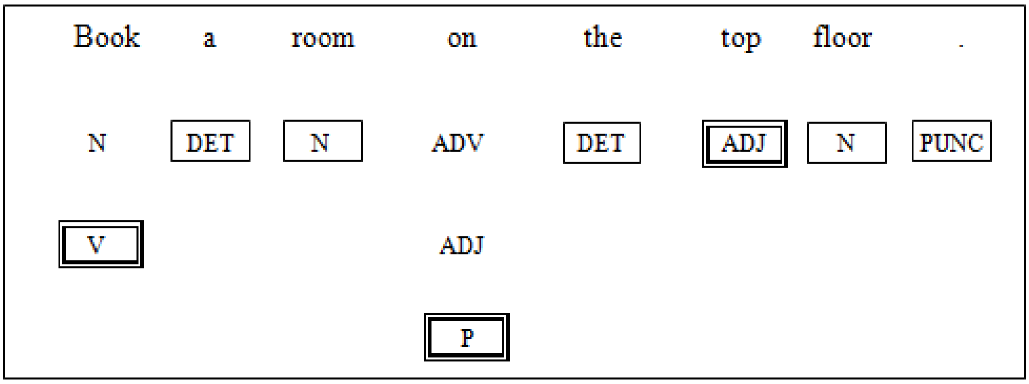

Figure 3.1: POS ambiguity of an English sentence with eight basic tags

Figure 3.2 illustrates the ambiguity in lexical items of a Manipuri sentence as per the tagset used for our experiment. As we are using a fine grained tagset compare to the basic 6 tags, the number of possible tags for a word increases.

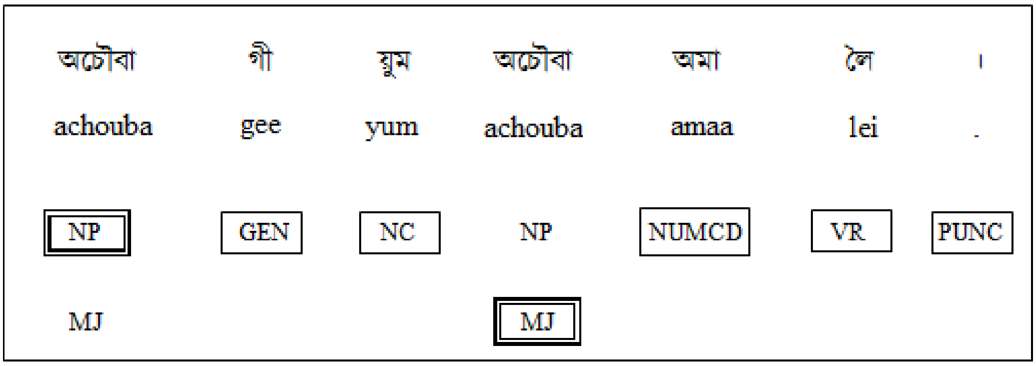

Figure 3.2: POS ambiguity of a Manipuri sentence with tagset of experiment

Part of speech tagging task resolves ambiguity by selecting the correct tag from the set of possible tags for a word or a lexical item in a sentence. Thus the problem can be viewed as a classification task.

More formally, the statistical definition of POS tagging can be stated as follows. Given a sequence of words W = $w_1.....w_n$, we want to find the corresponding sequence of tags T = $t_1...... t_n$, drawn from a set of tags {T}, which satisfies:

$$S = \arg\max_{t_1....t_n}(t_1.....t_n \mid w_1.....w_n)$$

3.2 Different types of Part of Speech Tagging

There are different models for part of speech tagging. It can be classified as Supervised and Unsupervised. Both the supervised and unsupervised model can be classified as rule-based and stochastic model.

3.2.1 Supervised and Unsupervised Part of Speech Tagging

The supervised part of speech tagging model requires a pre-tagged corpus which is used for training to learn information about the tagset, word-tag frequencies, rule sets etc [48]. The performance of the model generally increases with the increase in size of the corpus.

In contrast to the supervised model, the unsupervised model of part of speech tagging does not require a pre-tagged corpus. On the contrary, unsupervised model uses the advanced computational methods like Baum-Welch algorithm, transformation rules etc. to automatically induce tagsets.

3.2.2 Rule based Part of Speech Tagging

Rule based part of speech tagging is the approach that uses handwritten rules for tagging. Rule based tagger depends on dictionary or lexicon to get possible tags for each word to be tagged. Hand-written rules are used to identify the correct tag when a word has more than one possible tag. These rules are often known as context frame rules. Disambiguation is done by analysing the linguistic features of the word, its preceding word, its following word and other aspects. For example, if the preceding word is an adjective then the word in question must be adjective or noun. This information is coded in the form of rules [39].

3.2.3 Stochastic Part of Speech Tagging

The instinct behind the stochastic part of speech tagging is to find out the most frequently used tag for a specific word in the annotated training data and uses the same information to tag the word in the unannotated text. The drawback of this approach is that it comes with sequences of tags for sentences that are not satisfactory according to the grammatical rules of a language.

Another approach substitute to the word frequency approach is known as the n-gram approach. N-gram calculates the probability of a given sequence of tags. It determines the best tag for a word by calculating the probability that 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 unigram, bigram and trigram models respectively. The most common algorithm for implementing n-gram approach for tagging new text is known as the Viterbi Algorithm [38], which is a search algorithm that avoids the polynomial expansion of a breath 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.

There are different models that can be used for stochastic POS tagging, some of which are described below:

3.2.3.1 HMM Part of Speech tagging

Hidden Markov Model (HMM) tagger generally selects a tag sequence for a whole sentence rather than for a single word [26]. For a given sentence or word sequence, HMM taggers choose the tag sequence that maximizes the following formula:

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

For finding the maximum probability HMM uses the Viterbi algorithm.

Viterbi for Pos tagging:

Initialization Step

```
For i=1 to N do
Seqscore (i, 1) = Prob (w1 | Li)*Prob (Li |ø)
Backptr (i, 1) =0;
```

Iteration Step

```
For t=2 to T
For i=1 to N
```

Seqscore (i, t) = MAX j=1, N (Seqscore (i,t-1)*Prob (L_i | L_j)) * Prob (w_t | L_i)

Backptr (i, t) = index of j that gave the MAX above Sequence

Identification step

C (T) = I that maximizes Sequence (i, T)

For i=T-1 to 1 do

C (i) = Backptr (C (i+1), i+1)

w_1, ...w_T : Word Sequence

L_i, ...L_N : Lexical Categories

Prob (w $_t$ | L_i) : Lexical Probability

Prob (L_i | L_j) : Bigram Probability

3.2.3.2 MEM Part of Speech tagging

Maximum Entropy Model (MEM) is a very flexible process of statistical modelling. The MEM estimates the probabilities based on the imposed constraints. Such constraints are derived from the training data, maintaining some relationship between *history* and *outcomes* i.e. set of allowable tags. MEM allows the computation of P (t | h) for any t from the space of possible outcome T; for every h from the space of possible histories, H. In POS disambiguation task, we can reframe this in terms of finding the probability of a POS tag (t) associated with the token at index i in the test corpus as:

P (t | h_i) = P (t | information derivable from the test corpus at index i)

The computation of in ME depends on a set of possible *features* which are helpful to predict the outcome. Like most current ME modelling efforts, we restrict ourselves to the features which are binary function of history and outcome P (t | h). Given a set of features and the training data, the ME estimation process produces a model in which every feature *fi* is associated with a parameter *λi*. This allows the computation of the conditional probability as follows:

$$P(\mathrm{t} \mid \mathrm{h}) = \frac{\Pi_i \lambda_i^{fi(\mathrm{h},\mathrm{t})}}{\mathrm{Z}\lambda(\mathrm{h})}$$

$$\mathrm{Z}\lambda(\mathrm{h}) = \sum_t \Pi_i \lambda_i^{fi(\mathrm{h},\mathrm{t})}$$

To reframe, the above equation tells us that the conditional probability of the outcome given the history is the product of the weights of all the features, normalized over the products for all the outcomes.

3.2.3.3 MBL Part of Speech tagging

Memory-Based Learning (MBL) is a Machine Learning technique, where the training part consists in storing the training examples without restructuring. During classification the input is compared to the stored examples. *k* most similar examples are retrieved and majority voting is used to pick the class label. The method may be parameterised with the number of neighbours (*k*), a similarity metric between feature values, a scheme of feature weighting and a scheme of weighting neighbours as a function of their distance.

In the case of morphosyntactic tagging, the problem must be first formulated as a classification task. This may be achieved by representing the context of each token as a fixed-width feature vector. The features usually include parts-of-speech of neighbouring tokens, their orthographic forms, sometimes also fixed-width affixes of the word forms [21]. Memory Based Learning is able to train a working tagger using an annotated corpus. The corpus must contain a sequence of tokens and their corresponding MSD (Morphosyntactic Descriptions) tags. Optionally, external features may also be included. During training a lexicon of frequent word forms is generated. The lexicon is a mapping of word forms into ordered sets of encountered MSD tags. Rare forms are treated separately to account for unknown words when tagging. The default set of features includes form suffixes to facilitate guessing of correct tag. A drawback of MBT (Memory Based Tagger) when applied to inflectional languages is that it treats the feature values atomically, hence it cannot reason using the attribute values inferred from tags. This, however, can be altered by introducing additional features directly to the input before running MBT.

3.2.3.4 CRF Model Part of Speech tagging

A Conditional Random Field (CRF) is a framework of probabilistic model to segment and label a sequence of data. A conditional model specifies the probabilities of possible label sequences given an observation sequence. The conditional probability of the label sequence can depend on arbitrary, non-independent features of the observation sequence. The probability of a transition between labels may depend not only on the current observation, but also on past and future observations [48]. The CRF model calculates the probability based on some features, which might include the suffix of the current word, the tags of previous and next words, the actual previous and next etc.

Statistical definition of CRF is the probability of a particular label sequence *y* given the observation sequence *x* to be a normalized product of the potential functions, each of the form

$$\exp\left(\sum_j \lambda_j \mathrm{t_j}(\mathrm{y_{i-1}}, \mathrm{y_i}, \mathrm{x}, \mathrm{i}) + \sum_k \mu_k s_k(\mathrm{y_i}, \mathrm{x}, \mathrm{i})\right)$$

Where, t_j (y_i*-1,* y_i*, x, i*) is a transition feature function of the entire observation sequence and the labels at position *i* and *i-1* in the label sequence. S_k (y_i*, x, i*) is a state feature function of the label at position *i* and the observation sequence. λ_j and S_k are the model parameters to be estimated from the training data [51].

3.2.4 TBL Part of Speech tagging

Transformation Based Learning (TBL) is a rule based algorithm of tagging part of speech to the given text. TBL transforms one state to another using transformation rules in order to find the proper tag for each word. TBL allows us to have linguistic knowledge in a readable form. It extracts linguistic information automatically from corpora. The outcome of TBL is an ordered sequence of transformations of the form as shown below.

$$\text{Tag}_i \text{ -> } \text{Tag}_j \text{ in context C}$$

A typical transformation-based learner has an initial state annotator, a set of transformations and an objective function. Initial state annotator is a program to assign tags to each and every word in the given text. It may be one that assigns tags randomly or a Markov model tagger. Usually it assigns every word its most likely tag as indicated in the training corpus. For example, walk would be initially labelled as a verb.

The learner is given allowable transformation types. A tag may change from X to Y if the previous word is W, the previous tag is t_i and the following tag is t_j, or the tag two before is t_i and the following word is W. Consider the following sentence,

The rabbit runs.

A typical TBL tagger (or Brill Tagger) can easily identify that rabbit is noun if it is given the rule, if the previous tag is an article and the following tag is a verb [39].

3.3 Applications

Part of speech tagging has a wide range of applications in the field of Natural Language Processing. It is the preliminary stage of natural language understanding following which the further processing like chunking, parsing etc are normally done. Part of speech tagging is used in a number of applications, including- speech synthesis and recognition, information extraction, partial parsing, machine translation and lexicography etc [47].

Generally, the natural language understanding systems are formed by a set of pipelined modules; each module is specific to a particular level of analysis of the natural language text. As part of speech tagging is the initial step towards the understanding of natural language, it is very important to attain a high level of accuracy which otherwise may hamper further stages of natural language processing. We briefly discuss some of the above applications of POS tagging as follows:

(i) Speech synthesis and recognition: Part of Speech gives significant amount of information about word and its surrounding words which can be useful in a language model for speech recognition. Part of speech of a word indicates how the word is pronounced depending on the grammatical category. [37].

(ii) Information retrieval and extraction: If a query is given with part of speech information to a retrieval system more accurate information can be extracted.

(iii) Machine translation: The possibility of translating a word from the source language to target language is effectively dependent on the part of speech category of the source word.

(iv) Chunking: It is very easy to select the subset of the words after tagging the proper grammatical categories to each word or token of the sentence.

As stated above, part of speech tagging has been used in many other applications such as processor to high level syntactic processing, lexicography and word sense disambiguation etc.[17],[64].

3.4 Chapter Summary

In this chapter, a discussion on some fundamental concepts of part of speech tagging, viz., supervised and unsupervised part of speech tagging, rule based part of speech tagging, stochastic part of speech tagging and transformation based part of speech tagging and well known models of stochastic part of speech tagging like Hidden Markov Model, CRF model, Maxim Entropy model, Memory based part of speech tagging is presented. Further the chapter also presented some applications of part of speech tagging in different domains of NLP.