

Part-of-Speech Tagging for Bengali

*Thesis submitted to
Indian Institute of Technology, Kharagpur
for the award of the degree*

of

Master of Science

by

Sandipan Dandapat

Under the guidance of
Prof. Sudeshna Sarkar and Prof. Anupam Basu



**Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur
January, 2009**

© 2009, Sandipan Dandapat. All rights reserved.

CERTIFICATE OF APPROVAL

.../.../.....

Certified that the thesis entitled PART-OF-SPEECH TAGGING FOR BENGALI submitted by SANDIPAN DANDAPAT to Indian Institute of Technology, Kharagpur, for the award of the degree of Master of Science has been accepted by the external examiners and that the student has successfully defended the thesis in the viva-voce examination held today.

Signature
Name

(Member of the DSC)

Signature
Name

(Member of the DSC)

Signature
Name

(Member of the DSC)

Signature
Name

(Supervisor)

Signature
Name

(Supervisor)

Signature
Name

(External Examiner)

Signature
Name

(Chairman)

DECLARATION

I certify that the work contained in this thesis is original and has been done by me under the guidance of my supervisors. The work has not been submitted to any other Institute for any degree or diploma. I have followed the guidelines provided by the Institute in preparing the thesis. I have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute. Whenever I have used materials (data, theoretical analysis, figures, and text) from other sources, I have given due credit to them by citing them in the text of the thesis and giving their details in the references. Further, I have taken permission from the copyright owners of the sources, whenever necessary.

Sandipan Dandapat

CERTIFICATE

This is to certify that the thesis entitled **Part-of-Speech Tagging for Bengali**, submitted by **Sandipan Dandapat** to Indian Institute of Technology, Kharagpur, is a record of bona fide research work under my (our) supervision and is worthy of consideration for the award of the degree of Master of Science of the Institute.

(DR. ANUPAM BASU)

Professor

Dept. of Computer Science & Engg.,

Indian Institute of Technology

Kharagpur – 721302, INDIA

Date:

(DR. SUDESHNA SARKAR)

Professor

Dept. of Computer Science & Engg.,

Indian Institute of Technology

Kharagpur – 721302, INDIA

Date:

ACKNOWLEDGEMENT

I wish to express my profound sense of gratitude to my supervisors Prof. Sudeshna Sarkar and Prof. Anupam Basu, for introducing me to this research topic and providing their valuable guidance and unfailing encouragement throughout the course of the work. I am immensely grateful to them for their constant advice and support for successful completion of this work.

I am very much thankful to all the faculty members, staff members and research scholars of the Department of Computer Science and Engineering for their direct or indirect help in various forms during my research work. I would like to thank the co-researchers of Communication Empowerment Laboratory for providing me adequate help whenever required.

Finally I express my special appreciation and acknowledgement to my parents for their constant support, co-operation and sacrifice throughout my research work.

Last but not the least; I thank all my well-wishers who directly or indirectly contributed for the completion of this thesis.

Sandipan Dandapat

Date:

Abstract

Part-of-Speech (POS) tagging is the process of assigning the appropriate part of speech or lexical category to each word in a natural language sentence. Part-of-speech tagging is an important part of Natural Language Processing (NLP) and is useful for most NLP applications. It is often the first stage of natural language processing following which further processing like chunking, parsing, etc are done.

Bengali is the main language spoken in Bangladesh, the second most commonly spoken language in India, and the seventh most commonly spoken language in the world with nearly 230 million total speakers(189 million native speakers). Natural language processing of Bengali is in its infancy. POS tagging of Bengali is a necessary component for most NLP applications of Bengali. Development of a Bengali POS tagger will influence several pipelined modules of natural language understanding system including information extraction and retrieval; machine translation; partial parsing and word sense disambiguation. Our objective in this work is to develop an effective POS tagger for Bengali.

In this thesis, we have worked on the automatic annotation of part-of-speech for Bengali. We have defined a tagset for Bengali. We manually annotated a corpus of 45,000 words. We have used adaptations of different machine learning methods, namely Hidden Markov Model (HMM), Maximum Entropy model (ME) and Conditional random Field (CRF).

Further, to deal with a small annotated corpus we explored the use of semi-supervised learning by using an additional unannotated corpus. We also explored the use of a dictionary to provide to us all possible POS labeling for a given word. Since Bengali is morphologically productive, we had to make use of a Morphological Analyzer (MA) along with a dictionary of root words. This in turn restricts the set of possible tags for a given word. While MA helps us to restrict the possible choice of tags for a given word, one can also use prefix/suffix information (i.e., the sequence of first/last few characters of a word) to further improve the models. For HMM models, suffix information has been used during smoothing of emission probabilities, whereas for ME and CRF models, suffix information is used as features.

The major contribution of the thesis can be outlined as follows:

- We have used HMM model for the Bengali POS tagging task. In order to develop an effective POS tagger with a small tagged set, we have used other resources like a dictionary and a morphological analyzer to improve the performance of the tagger.
- Machine learning techniques for acquiring discriminative models have been applied for Bengali POS tagging task. We have used Maximum Entropy and Conditional Random Field based model for the task.
- From a practical perspective, we would like to emphasize that a resources of 50,000 words POS annotated corpora have been developed as a result of the

work. We have also presented a tagset for Bengali that has been developed as a part of the work.

We have achieved higher accuracy than the naive baseline model. However, the performance of the current system is not as good as that of the contemporary POS-taggers available for English and other European languages. The best performance is achieved for the supervised learning model along with suffix information and morphological restriction on the possible grammatical categories of a word.

Content

List of Figures	x
List of Tables.....	xi
CHAPTER 1	1
Introduction	1
1.1. The Part-of-Speech Tagging Problem	3
1.2. Applications of POS Tagging.....	5
1.3. Motivation	6
1.4. Goals of Our Work	8
1.5. Our Particular Approach to Tagging	8
1.6. Organization of the Thesis	10
CHAPTER 2	12
Prior Work in POS Tagging.....	12
2.1. Linguistic Taggers.....	13
2.2. Statistical Approaches to Tagging.....	14
2.3. Machine Learning based Tagger	15
2.4. Current Research Directions	17
2.5. Indian Language Taggers	20
2.6. Acknowledgement.....	24
CHAPTER 3	25
Foundational Considerations	25
3.1. Corpora Collection	26
3.2. The Tagset.....	26
3.3. Corpora and Corpus Ambiguity	30
CHAPTER 4	34
Tagging with Hidden Markov Model	34
4.1. Hidden Markov Model	34
4.2. Our Approach.....	37

4.3.	Experiments.....	48
4.4.	System Performance.....	49
4.5.	Conclusion.....	55
CHAPTER 5		56
Tagging with Maximum Entropy Model.....		56
5.1.	Maximum Entropy Model.....	57
5.2.	Our Particular Approach with ME Model.....	59
5.3.	Experiments.....	68
5.4.	System Performance.....	70
5.5.	Conclusion.....	74
CHAPTER 6		76
Tagging with Conditional Random Fields.....		76
6.1.	Conditional Random Fields.....	77
6.2.	Experimental Setup.....	81
6.3.	System Performance.....	82
6.4.	Conclusion.....	85
CHAPTER 7		87
Conclusion.....		87
7.1.	Contributions.....	90
7.2.	Future Works.....	92
List of Publications.....		94
References		96
Appendix A		106
Lexical Categories (Tags) for Bengali.....		106
Appendix B		117
Results obtained by Maximum Entropy based Bengali POS Tagger		117

List of Figures

Figure 1: POS ambiguity of an English sentence with eight basic tags.....	4
Figure 2: POS ambiguity of a Bengali sentence with tagset of experiment	4
Figure 3: POS tagging schema.....	9
Figure 4: Vocabulary growth of Bengali and Hindi	32
Figure 5: General Representation of an HMM	36
Figure 6: The HMM based POS tagging architecture	37
Figure 7: Uses of Morphological Analyzer during decoding	45
Figure 8: The accuracy growth of different supervised HMM models.	50
Figure 9: The accuracy growth of different semi-supervised HMM tagging models.....	50
Figure 10: Known and Unknown accuracy under different HMM based models .	51
Figure 11: The ME based POS tagging architecture	59
Figure 12: The Potential Feature Set (F) for the ME model	61
Figure 13: The Beam search algorithm used in the ME based POS tagging model	63
Figure 14: Decoding the most probable tag sequence in ME based POS tagging model.....	65
Figure 15: Search procedure using MA in the ME based POS tagging model.....	67
Figure 16: The overall accuracy growth of different ME based tagging model	70
Figure 17: The known and unknown word accuracy under different ME based model.....	71
Figure 18: Graphical structure of a chain-structured CRF for sequences.....	78
Figure 19: The overall accuracy growth of different CRF based POS tagging model.....	83
Figure 20: Known and unknown word accuracies with the CRF based models ...	84

List of Tables

Table 1: Summary of the approaches and the POS tagging accuracy in the NLP AI machine learning contest.....	23
Table 2: Summary of the approaches and the POS tagging accuracy in the SPSAL machine learning contest.....	23
Table 3: The tagset for Bengali with 40-tags.....	29
Table 4: Tag ambiguity of word types in Brown corpus (DeRose , 1988).....	31
Table 5: Tag ambiguity of word types in Bengali CIIL corpus.....	31
Table 6: Corpus ambiguity, Tagging accuracy and percentage of unknown word (open testing text) for different language corpora used for POS tagging	33
Table 7: Tagging accuracies (%) of different models with 10K, 20K and 40K training data. The accuracies are represented in the form of <i>Overall Accuracy (Known Word Accuracy, Unknown Word Accuracy)</i>	52
Table 8: Five most common types of errors	54
Table 9: Feature used in the simple ME based POS tagging.....	69
Table 10: Tagging accuracies (%) of different models with 10K, 20K and 40K training data. The accuracies are represented in the form of <i>Overall Accuracy (Known Word Accuracy, Unknown Word Accuracy)</i>	72
Table 11: Tagging Accuracy with morphology as a feature in ME based POS tagging model.....	72
Table 12: Five most common types of errors with the ME model	74
Table 13: Tagging accuracies (%) of different models with 10K, 20K and 40K training data. The accuracies are represented in the form of <i>Overall Accuracy</i>	84

Chapter 1

Introduction

Part-of-Speech (POS) tagging is the process of automatic annotation of lexical categories. Part-of-Speech tagging assigns an appropriate part of speech tag for each word in a sentence of a natural language. The development of an automatic POS tagger requires either a comprehensive set of linguistically motivated rules or a large annotated corpus. But such rules and corpora have been developed for a few languages like English and some other languages. POS taggers for Indian languages are not readily available due to lack of such rules and large annotated corpora.

The linguistic approach is the classical approach to POS tagging was initially explored in middle sixties and seventies (Harris, 1962; Klein and Simmons, 1963; Greene and Rubin, 1971). People manually engineered rules for tagging. The most representative of such pioneer tagger was TAGGIT (Greene and Rubin, 1971), which was used for initial tagging of the Brown Corpus. The development of ENGTWOL (an English tagger based on constraint grammar architecture) can be considered most important in this direction (Karlsson et al., 1995). These taggers typically use rule-based models manually written by linguists. The advantage of this model is that the rules are written from a linguistic point of view and can be made to capture complex kinds of information. This allows the construction of an extremely accurate system. But handling all rules is not easy

and requires expertise. The context frame rules have to be developed by language experts and it is costly and difficult to develop a rule based POS tagger. Further, if one uses of rule based POS tagging, transferring the tagger to another language means starting from scratch again.

On the other hand, recent machine learning techniques makes use of annotated corpora to acquire high-level language knowledge for different tasks including PSO tagging. This knowledge is estimated from the corpora which are usually tagged with the correct part of speech labels for the words. Machine learning based tagging techniques facilitate the development of taggers in shorter time and these techniques can be transferred for use with corpora of other languages. Several machine learning algorithms have been developed for the POS disambiguation task. These algorithms range from instance based learning to several graphical models. The knowledge acquired may be in the form of rules, decision trees, probability distribution, etc. The encoded knowledge in stochastic methods may or may not have direct linguistic interpretation. But typically such taggers need to be trained with a handsome amount of annotated data to achieve high accuracy. Though significant amounts of annotated corpus are often not available for most languages, it is easier to obtain large volumes of un-annotated corpus for most of the languages. The implication is that one may explore the power of semi-supervised and unsupervised learning mechanism to get a POS tagger.

Our interest is in developing taggers for Indian Languages. Annotated corpora are not readily available for most of these languages, but many of the languages are morphologically rich. The use of morphological features of a word, as well as word suffixes can enable us to develop a POS tagger with limited resources. In the present work, these morphological features (affixes) have been incorporated in different machine learning models (Maximum Entropy, Conditional Random Field, etc.) to perform the POS tagging task. This approach can be generalized for use with any morphologically rich language in poor-resource scenario.

The development of a tagger requires either developing an exhaustive set of linguistic rules or a large amount of annotated text. We decided to use a machine learning approach to develop a part of speech tagger for Bengali. However no tagged corpus was available to us for use in this task. We had to start with creating tagged resources for Bengali. Manual part of speech tagging is quite a time consuming and difficult process. So we tried to work with methods so that small amount of tagged resources can be used to effectively carry out the part of speech tagging task.

Our methodology can be used for the POS disambiguation task of any resource poor language. We have looked at adapting certain standard learning approaches so that they can work well with scarce data. We have also carried on comparative studies of the accuracies obtained by working with different POS tagging methods, as well as the effect on the learning algorithms of using different features.

1.1. The Part-of-Speech Tagging Problem

Natural languages are ambiguous in nature. Ambiguity appears at different levels of the natural language processing (NLP) task. Many words take multiple part of speech tags. The correct tag depends on the context.

Consider, for instance, the following English and Bengali sentence.

1. *Keep the book on the top shelf.*
2. সকালবেলায় তারা ক্ষেতে লাঙল দিয়ে কাজ করে।
sakAlabelAYa tArA kShete lA~Nala diYe kAja kare.
Morning they field plough with work do.
They work in the field with the plough in the morning.

The sentences have lot of POS ambiguity which should be resolved before the sentence can be understood. For instance in example sentence 1, the word ‘*keep*’ and ‘*book*’ can be a noun or a verb; ‘*on*’ can be a preposition, an adverb, an adjective; finally, ‘*top*’ can be either an adjective or a noun. Similarly, in

Bengali example sentence 2, the word ‘তারা(/tArA/)’ can be either a noun or a pronoun; ‘দিয়ে(/diYe/)’ can be either a verb or a postposition; ‘করে(/kare/)’ can be a noun, a verb, or a postposition. In most cases POS ambiguity can be resolved by examining the context of the surrounding words. Figure 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 double line indicate the correct POS tag of a word form a set of possible tags.

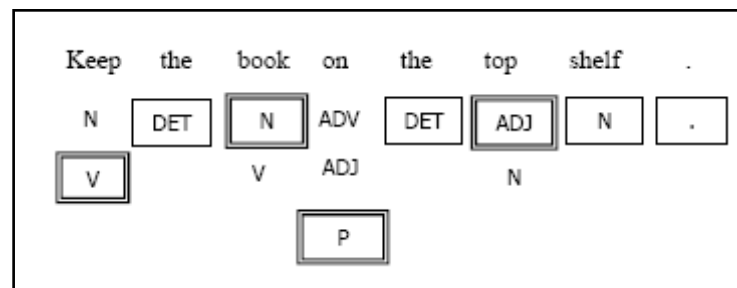


Figure 1: POS ambiguity of an English sentence with eight basic tags

Figure 2 illustrate the detail of the ambiguity class for the Bengali sentence as per the tagset used for our experiment. As we are using a fine grained tagset compare to the basic 8 tags, the number of possible tags for a word increases.

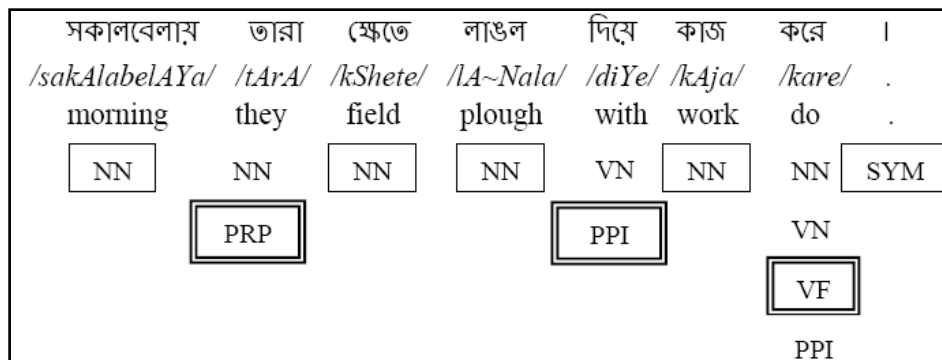


Figure 2: POS ambiguity of a Bengali sentence with tagset of experiment

POS tagging is the task of assigning appropriate grammatical tags to each word of an input text in its context of appearance. Essentially, the POS tagging

task resolves ambiguity by selecting the correct tag from the set of possible tags for a word 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 $W=w_1 \dots w_n$, we want to find the corresponding sequence of tags $T=t_1 \dots t_n$, drawn from a set of tags $\{T\}$, which satisfies:

$$S = \arg \max_{t_1 \dots t_n} P(t_1 \dots t_n | w_1 \dots w_n) \quad \text{Eq. 1}$$

1.2. Applications of POS Tagging

POS disambiguation task is useful in several natural language processing tasks. It is often the first stage of natural language understanding following which further processing e.g., chunking, parsing, etc are done. Part-of –speech tagging is of interest for a number of applications, including – speech synthesis and recognition (Nakamura et al., 1990; Heeman et al., 1997), information extraction (Gao et al., 2001; Radev et al., 2001; Argaw and Asker, 2006), partial parsing (Abney, 1991; Karlsson et al., 1995; Wauschkuhn, 1995; Abney, 1997; Vuoltilainen and Padro, 1997; Padro, 1998), machine translation, lexicography etc.

Most of the natural language understanding systems are formed by a set of pipelined modules; each of them is specific to a particular level of analysis of the natural language text. Development of a POS tagger influences several pipelined modules of the natural language understanding task. As POS tagging is the first step towards natural language understating, it is important to achieve a high level of accuracy which otherwise may hamper further stages of the natural language understanding. In the following, we briefly discuss some of the above applications of POS tagging.

- *Speech synthesis and recognition*, Part-of-speech gives significant amount of information about the word and its neighbours which can be useful in a

language model for speech recognition (Heeman et al., 1997). Part-of-speech of a word tells us something about how the word is pronounced depending on the grammatical category (the noun is pronounced *OBject* and the verb *obJECT*). Similarly, in Bengali, the word ‘করে(/kare/)’ (postposition) is pronounced as ‘kore’ and the verb ‘করে(/kare/)’ is pronounced as ‘kOre’.

- *Information retrieval and extraction*, by augmenting a query given to a retrieval system with POS information, more refined information extraction is possible. For example, if a person wants to search for document containing ‘book’ as a noun, adding the POS information will eliminate irrelevant documents with only ‘book’ as a verb. Also, patterns used for information extraction from text often use POS references.
- *Machine translation*, the probability of translating a word in the source language into a word in the target language is effectively dependent on the POS category of the source word. E.g., the word ‘দিয়ে(/diYe/)’ in Bengali will be translated as either *by* or *giving* depending on its POS category, i.e. whether it is a *postposition* or *verb*.

As mentioned earlier, POS tagging has been used in several other application such as a processor to high level syntactic processing (noun phrase chunker), lexicography, stylometry, and word sense disambiguation. These applications are discussed in some detail in (Church, 1988; Ramshaw and Marcus, 1995; Wilks and Stevenson, 1998).

1.3. Motivation

A lot of work has been done in part of speech tagging of several languages, such as English. While some work has been done on the part of speech tagging of different Indian languages (Ray et al., 2003; Shrivastav et al., 2006; Arulmozhi et al., 2006; Singh et al., 2006; Dalal et al., 2007), the effort is still in its infancy. Very little work has been done previously with part of speech tagging of Bengali.

Bengali is the main language spoken in Bangladesh, the second most commonly spoken language in India, and the seventh most commonly spoken language in the world.

Apart from being required for further language analysis, Bengali POS tagging is of interest due to a number of applications like speech synthesis and recognition. Part-of-speech gives significant amount of information about the word and its neighbours which can be useful in a language model for different speech and natural language processing applications. Development of a Bengali POS tagger will also influence several pipelined modules of natural language understanding system including: information extraction and retrieval; machine translation; partial parsing and word sense disambiguation. The existing POS tagging technique shows that the development of a reasonably good accuracy POS tagger requires either developing an exhaustive set of linguistic rules or a large amount of annotated text. We have the following observations.

- Rule based POS taggers uses manually written rules to assign tags to unknown or ambiguous words. Although, the rule based system allows the construction of an extremely accurate system, it is costly and difficult to develop a rule based POS tagger.
- Recent machine learning based POS taggers use a large amount of annotated data for the development of a POS tagger in shorter time.
- However, no tagged corpus was available to us for the development of a machine learning based POS tagger.

Therefore, there is a pressing necessity to develop a automatic Part-of-Speech tagger for Bengali. With this motivation, we identify the major goals of this thesis.

1.4. Goals of Our Work

The primary goal of the thesis is to develop a reasonably good accuracy part-of-speech tagger for Bengali. To address this broad objective, we identify the following goals:

- We wish to investigate different machine learning algorithm to develop a part-of-speech tagger for Bengali.
- As we had no corpora available to use we had to start creating resources for Bengali. Manual part of speech tagging is quite a time consuming and difficult process. So we wish to work with methods so that small amount of tagged resources can be used to effectively carry on the part of speech tagging task.
- Bengali is a morphologically-rich language. We wish to use the morphological features of a word, as well as word suffix to enable us to develop a POS tagger with limited resource.
- The work also includes the development of a reasonably good amount of annotated corpora for Bengali, which will directly facilitate several NLP applications.
- Finally, we aim to explore the appropriateness of different machine learning techniques by a set of experiments and also a comparative study of the accuracies obtained by working with different POS tagging methods.

1.5. Our Particular Approach to Tagging

Our particular approach to POS tagging belongs to the machine learning family, and it is based on the fact that the POS disambiguation task can be easily interpreted as a classification problem. In the POS disambiguation task, the finite set of classes is identified with the set of possible tags and the training examples are the occurrences of the words along with the respective POS category in the context of appearance.

A general representation of the POS tagging process is depicted in the Figure 3. We distinguish three main components. The system uses some knowledge about the task for disambiguation for POS disambiguation. This knowledge can be encoded in several representations and may come from several resources. We shall call this model as *language model*. On the other hand there is a *disambiguation algorithm*, which decides the best possible tag assignment according to the language model. The third component estimates the set possible tags $\{T\}$, for every word in a sentence. We shall call this as *possible class restriction* module. This module consists of list of lexical units with associated list of possible tags. These three components are related and we combine them into a single tagger description. The input to the disambiguation algorithm takes the list of lexical units with the associated list of possible tags. The disambiguation module provides the output consist of the same list of lexical units reducing the ambiguity, using the encoded information from the language model.

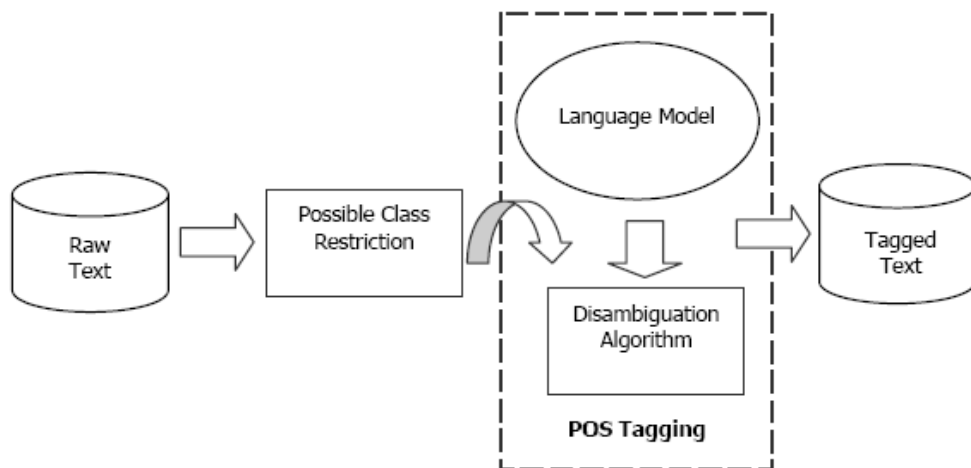


Figure 3: POS tagging schema

We used different graphical models to acquire and represent the language model. We adopt Hidden Markov Model, Maximum Entropy model and Conditional Random Field, which has widely been used in several basic NLP

applications such as tagging, parsing, sense disambiguation, speech recognition, etc., with notable success.

1.6. Organization of the Thesis

Rest of this thesis is organized into chapters as follows:

Chapter 2 provides a brief review of the prior work in POS tagging. We do not aim to give a comprehensive review of the related work. Such an attempt is extremely difficult due to the large number of publication in this area and the diverse language dependent works based on several theories and techniques used by researchers over the years. Instead, we briefly review the work based on different techniques used for POS tagging. Also we focus onto the detail review of the Indian language POS taggers.

Chapter 3 supply some information about several important issues related to POS tagging, which can greatly influence the performance of the taggers, as well as the process of comparison and evaluation of taggers.

Chapter 4 describes our approach of applying Hidden Markov Model (HMM) to eliminate part-of-speech ambiguity. We outline the general acquisition algorithm and some particular implementations and extensions. This chapter also describes the use of morphological and contextual information for POS disambiguation using HMM. Further, we present the semi-supervised learning by augmenting the small labelled training set with a larger unlabeled training set. The models are evaluated against a reference corpus with a rigorous methodology. The problem of unknown words is also addressed and evaluated in this chapter.

Chapter 5 describes our work on Bengali POS tagging using Maximum Entropy based statistical model. In this chapter, we also present the uses of a morphological analyzer to improve the performance of a tagger in the maximum

entropy framework. We also present the uses of different features and their effective performance in the Maximum Entropy model.

Chapter 6 presents our work on Bengali POS tagging using Conditional Random Fields (CRF). We use the same potential features of the Maximum Entropy model in the CRF framework to understand the relative performance of the models. Here, we also use morphological information for further improvement of the tagging accuracy.

Chapter 7 provides general conclusion, summarizes the work and contribution of the thesis, and outline several direction for future work.

Appendixes. Some appendixes have been added in order to cover the complementary details. More precisely, the list included materials are:

Appendix A fully describes the tagset used for tagging the Bengali corpora.

Appendix B includes the detail experimental results with Maximum Entropy based model.

Chapter 2

Prior Work in POS Tagging

The area of automated Part-of-speech tagging has been enriched over the last few decades by contribution from several researchers. Since its inception in the middle sixties and seventies (Harris, 1962; Klein and Simmons, 1963; Greene and Rubin, 1971), many new concepts have been introduced to improve the efficiency of the tagger and to construct the POS taggers for several languages. Initially, people manually engineered rules for tagging. Linguistic taggers incorporate the knowledge as a set of rules or constraints written by linguists. More recently several statistical or probabilistic models have been used for the POS tagging task for providing transportable adaptive taggers. Several sophisticated machine learning algorithms have been developed that acquire more robust information. In general all the statistical models rely on manually POS labeled corpora to learn the underlying language model, which is difficult to acquire for a new language. Hence, some of the recent works focus on semi-supervised and unsupervised machine learning models to cope with the problem of unavailability of the annotated corpora. Finally, combinations of several sources of information (linguistic, statistical and automatically learned) have been used in current research direction.

This chapter provides a brief review of the prior work in POS tagging. For the sake of consciousness, we do not aim to give a comprehensive review of the related work. Instead, we provide a brief review on the different techniques used

in POS tagging. Further, we focus onto the detail review of the Indian language POS taggers.

The first section of this Chapter provides a brief discussion on the work performed around linguistic POS tagging. Section 2 surveys a broad coverage compilation of references about the stochastic POS taggers. The third section discusses the application of general machine learning algorithms to address the POS tagging problem. In the fourth section, we briefly discuss the most recent efforts have been done in this area. Finally, the fourth section contains a detail description of the work on Indian Language POS tagging.

2.1. Linguistic Taggers

Automated part of speech tagging was initially explored in middle sixties and seventies (Harris, 1962; Klein and Simmons, 1963; Greene and Rubin, 1971). People manually engineered rules for tagging. The most representative of such pioneer tagger was TAGGIT (Greene and Rubin, 1971), which was used for initial tagging of the Brown Corpus. Since that time to nowadays, a lot of effort has been devoted to improving the quality of the tagging process in terms of accuracy and efficiency.

Recent linguistic taggers incorporate the knowledge as a set of rules or constraints, written by linguists. The current models are expressive and accurate and they are used in very efficient disambiguation algorithms. The linguistic rules range from a few hundred to several thousands, and they usually require years of labour. The development of ENGTWOL (an English tagger based on constraint grammar architecture) can be considered most important in this direction (Karlsson et al., 1995). The constraint grammar formalism has also been applied for other languages like Turkish (Oflazer and Kuruoz, 1994).

The accuracy reported by the first rule-based linguistic English tagger was slightly below 80%. A Constraint Grammar for English tagging (Samuelsson and Voutilainen, 1997) is presented which achieves a recall of 99.5% with a very

high precision around 97%. Their advantages are that the models are written from a linguistic point of view and explicitly describe linguistic phenomena, and the models may contain many and complex kinds of information. Both things allow the construction of extremely accurate system. However, the linguistic models are developed by introspection (sometimes with the aid of reference corpora). This makes it particularly costly to obtain a good language model. Transporting the model to other languages would require starting over again.

2.2. Statistical Approaches to Tagging

The most popular approaches nowadays use statistical or machine learning techniques. These approaches primarily consist of building a statistical model of the language and using the model to disambiguate a word sequence by assigning the most probable tag sequence given the sequence of words in a maximum likelihood approach. The language models are commonly created from previously annotated data, which encodes the co-occurrence frequency of different linguistic phenomena to simple n-gram probabilities.

Stochastic models (DeRose, 1988; Cutting et al., 1992; Dermatas and Kokkinakis, 1995; Mcteer et al., 1991; Merialdo, 1994) have been widely used POS tagging for simplicity and language independence of the models. Among stochastic models, bi-gram and tri-gram Hidden Markov Models (HMM) are quite popular. TNT (Brants, 2000) is a widely used stochastic trigram HMM tagger which uses a suffix analysis technique to estimate lexical probabilities for unknown tokens based on properties of the words in the training corpus which share the same suffix. The development of a stochastic tagger requires large amount of annotated text. Stochastic taggers with more than 95% word-level accuracy have been developed for English, German and other European languages, for which large labeled data is available. Simple HMM models do not work well when small amounts of labeled data are used to estimate the model parameters. Sometimes additional information is coded into HMM model to achieve high accuracy for POS tagging (Cutting et al., 1992). For example,

Cutting et al (1992) propose an HMM model that uses a lexicon and an untagged corpus for accurate and robust tagging.

The advantage of the HMM model is that the parameters of the model can be re-estimated with the Baum-Welch algorithm (Baum, 1972) to iteratively increase the likelihood of the observation data. This avoids the use of annotated training corpora or at least reduces the amount of annotated training data to estimate a reasonably good model. The semi-supervised (Cutting et al., 1992; Kupiec, 1992; Merialdo, 1994) model makes use of both labeled training text and some amount of unlabeled text. A small amount of labeled training text is used to estimate a model. Then the unlabeled text is used to find a model which best describe the observed data. The well known Baum-Welch algorithm is used to estimate the model parameters iteratively until convergence.

Some authors have performed comparison of tagging accuracy between linguistic and statistical taggers with favorable conclusion (Chanod and Tapanainen, 1995; Samuelsson and Voutilainen, 1997).

2.3. Machine Learning based Tagger

The statistical models use some kind of either supervised or unsupervised learning of the model parameters from the training corpora. Although the machine learning algorithms for classification tasks are usually statistical in nature, we consider in the machine learning family only those systems which acquire more sophisticated model than a simple n-gram model.

First attempt of acquiring disambiguation rules from corpus were done by Hindle (Hindle, 1989). Recently, Brill's tagger (Brill, 1992; Brill, 1995a; Brill 1995b) automatically learns a set of transformation rules which correct the errors of a most-frequent-tag tagger. The learning algorithm he proposed is called *Transformation-Based Error-Driven Learning* and it has been widely to resolve

several ambiguity problems in NLP. Further Brill proposed a semi supervised version of the learning algorithm which roughly achieve the same accuracy.

Instance based learning has been also applied by several authors to resolve a number of different ambiguity problems and in particular to POS tagging problem (Cardie, 1993a; Daelemans et al., 1996).

Decision trees have been used for POS tagging and parsing as in (Black et al., 1992; Magerman, 1995a). Decision tree induced from tagged corpora was used for part-of-speech disambiguation (Marquez and Rodriguez, 1998). In fact (Daelemans, 1996) can be seen as an application of a very special type of decision tree.

POS tagging has also been done using neural net architecture (Nakamura et al., 1990; Schutze, 1993; Eineborg and Gamback, 1993; and Ma and Isahar, 1998). There also exist some mixed approaches. For example forward backward algorithm is used to smooth decision tree probabilities in the works of (Black et al., 1992; Magerman, 1995a), and conversely, decision trees are used to acquire and smooth the parameter of a HMM model (Schmid, 1995b; Schmid, 1995a).

Support Vector Machines (SVM) has been used for POS tagging with simplicity and efficiency. Nakagawa (Nakagawa et al., 2001), first used the SVM based machine learning technique for POS tagging. The main disadvantage of the system was low efficiency (running speed of 20 words per second was reported). Further, Gimenez and Marquez (Gimenez and Marquez, 2003) in their work proposed a SVM based POS tagging technique which is 60 times faster than the earlier one. The tagger also significantly outperforms the TNT tagger. From the comparison of their paper, it has been observed that the accuracy for unknown word is better for the TnT tagger compared to the SVM taggers.

2.4. Current Research Directions

Recently lot of work has taken place on construction of POS taggers for a variety of languages and also for providing adaptive and transportable POS taggers. Current direction of research also includes the combination of statistical algorithms and the use of more sophisticated language models. Further, work has also been carried out to find out the underling language properties (*features*) for feature based classification algorithms (*e.g. Maximum Entropy Model, Conditional Random fields etc.*) for POS disambiguation. The following describe some of the recent efforts for the POS tagging problem:

2.4.1. POS tagger for large divergence of languages

Researchers are taking into account new problems for the development of a POS tagger for the variety of languages over the world. Due to the different inherent linguistic properties and the availability of language resources required for POS disambiguation, the following issues have been included in the focus of the current research in this area.

1. Learning from small training corpora (Kim and Kim, 1996; Jinshan et al., Padro and Padro, 2004)
2. Adopting very large tag set (Asahara and Matsumoto, ; Rooy and Schafer, ; Ribarvo, 2000)
3. Exploiting morphological features for morphologically rich languages including *highly agglutinative languages* (Dalal et al., 2007; Dandapat et al., 2007; Smriti et al., 2006)
4. Learning from un-annotated data (Biemann, 2007; Dasgupta and Ng, 2007; Kazama et al., 2001; Mylonakis et al., 2007)

In particular, taggers have been described for the following languages: Dutch (Dermatas and Kokkinakis, 1995a; Daelemans et al., 1996), French (Chando and Tapanainen, 1995; Tzoukermann et al., 1995), German (Feldweg, 1995, Lezius et al., 1996), Greek (Dermatas and Kokkinakis, 1995a), Japanese

(Matsukawa et al., 1993; Haruno and Matsumoto, 1997), Italian (Dermatas and Kokkinakis, 1995a), Spanish (Moreno-Torres, 1994, Marquez et al., 1998), Turkish (Oflazer and Kuruoz, 1994) and many more.

2.4.2. Providing adaptive and transportable tagger

The main aim here is to design taggers which can be ported from one domain to another domain without serious hampering tagging accuracy at a very low cost for adapting to new domain. This will require annotated corpus of the new domain, and in some cases new features may have to be considered. This is very much required for domain specific applications. Roth and Zelenko (Roth and Zelenko, 1998) presented the SNOW architecture for the type of task.

2.4.3. Combination of statistical information

The combination of statistical information has been proposed by several of the statistical based tagger as maintained previously, to obtain more accurate model parameters especially to overcome the problem of the sparseness of the data. However, different techniques of smoothing (*Back-off, linear interpolation, etc.*) were used to deal with the above problem. Recently, some work has been carried out to integrate and combine several sources of information for the POS tagging problem. The following are some examples:

A recent model which handles the sparse data problem is the Maximum Entropy (ME) model (Ratnaparkhi, 1996), which assume maximum entropy (i.e. uniform distribution). Under this model, a natural combination of several features can be easily incorporated, which can not be done naturally in HMM models. In the ME based approach, unobserved events do not have zero probability, but the maximum they can give the observations. Simple HMM models do not work well when small amount of labeled data are used to estimate the model parameters. Incorporating a diverse set of overlapping features in a HMM-based tagger is difficult and complicates the smoothing typically used for such taggers. In contrast, a ME based methods can deal with diverse, overlapping features

The combination of statistical and linguistic/rule based model has been encoded inside the rules/constrain-based environment. Some of the work can be found in (Oflazer and Tur, 1996; Tur and Oflazer, 1998, Tzoukermann et al., 1997).

Another model is designed for the tagging task by combining unsupervised Hidden Markov Model with maximum entropy (Kazama et al., 2001). 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.

2.4.4. Extending the language model inside the statistical approach

Recent works do not try to limit the language model to a fixed n-gram. Different orders of n-grams, long distance n-grams, non-adjacent words etc are constrained in more sophisticated systems. The speech recognition field is very productive in this issue. In particular we find Aggregate Markov Model and Mixed Markov Model (Brown et al., 1992; Saul and Pereira, 1997), Hierarchical Non-emitting Markov Model (Ristad and Thomas, 1997), Mixture of Prediction Suffix Trees (Pereira et al., 1995; Brants, 2000], have applied to POS tagging. Variable memory based Markov Model (Schutze and Singer, 1994) and Mixture of Hierarchical Tag Context Trees (Haruno and Matsumoto, 1997) has been applied to tagging and parsing.

Finally, Conditional Random Field (CRF) (Sha and Pereira, 2003; Lafferty, 2001; Shrivastav et al., 2006) has been applied for POS disambiguation task. Unlike Maximum Entropy model, it finds out the global maximum likelihood estimation. This model also captures the complex information in terms of features as on ME model.

2.4.5. Feature inspection

Recently, considerable amount of effort has been given to find out language specific features for the POS disambiguation task. Discriminative graphical models (e.g. maximum entropy model, CRF etc.) usually integrate different features for the disambiguation task. Some works (Kazama et al., 2001; McCallum et al., 2000; Zhao et al., 2004) report that discriminative model works better than the generative model (e.g. HMM). However, the power of the discriminative models lies in the features that have been used for the task. These features vary from language to language due to the inherent linguistic/grammatical properties of the language. The main contributions in this area are (Ratanaparkhi, 1996; Zavrel and Daelemans, 2004; Toutanova et al., Singh et al., 2006; Tseng et al. ;). Some of the above contributions are specific to Indian languages. The details of some of the experiments and results are described in the next section.

2.5. Indian Language Taggers

There has been a lot of interest in Indian language POS tagging in recent years. POS tagging is one of the basic steps in many language processing tasks, so it is important to build good POS taggers for these languages. However it was found that very little work has been done on Bengali POS tagging and there are very limited amount of resources that are available. The oldest work on Indian language POS tagging we found is by Bharati et al. (Bhartai et al., 1995). They presented a framework for Indian languages where POS tagging is implicit and is merged with the parsing problem in their work on computational Paninian parser.

An attempt on Hindi POS disambiguation was done by Ray (Ray et al. 2003). The part-of-speech tagging problem was solved as an essential requirement for local word grouping. Lexical sequence constraints were used to assign the correct POS labels for Hindi. A morphological analyzer was used to find out the possible POS of every word in a sentence. Further, the follow relation for lexical tag sequence was used to disambiguate the POS categories.

A rule based POS tagger for Tamil (Arulmozhi et al., 2004) has been developed in combination of both lexical rules and context sensitive rules. Lexical rules were used (*combination of suffixes and rules*) to assign tags to every word without considering the context information. Further, hand written context sensitive rules were used to assign correct POS labels for unknown words and wrongly tagged words. They used a very coarse grained tagset of only 12 tags. They reported an accuracy of 83.6% using only lexical rules and 88.6% after applying the context sensitive rules. The accuracy reported in the work, are tested on a very small reference set of 1000 words. Another hybrid POS tagger for Tamil (Arulmozhi et al., 2006) has also been developed in combination of a HMM based tagger with a rule based tagger. First a HMM based statistical tagger was used to annotate the raw sentences and it has been found some sentences/words are not tagged due to the limitation of the algorithm (no smoothing algorithm was applied) or the amount of training corpus. Then the untagged sentences/words are passed through the rule based system and tagged. They used the same earlier tagset with 12 tags and an annotated corpus of 30,000 words. Although the HMM tagger performs with a very low accuracy of 66% but, the hybrid system works with 97.3% accuracy. Here also the system has been tested with a small set of 5000 words and with a small tagset of 12 tags.

Shrivastav et al. (Shrivastav et al. 2006) presented a CRF based statistical tagger for Hindi. They used 24 different features (lexical features and spelling features) to generate the model parameters. They experimented on a corpus of around 12,000 tokens and annotated with a tagset of size 23. The reported accuracy was 88.95% with a 4-fold cross validation.

Smriti et al. (Smriti et al. 2006) in their work, describes a technique for morphology-based POS tagging in a limited resource scenario. The system uses a decision tree based learning algorithm (CN2). They used stemmer, morphological analyzer and a verb group analyzer to assign the morphotactic tags to all the words, which identify the *Ambiguity Scheme* and *Unknown Words*. Further, a

manually annotated corpus was used to generate *If-Then* rules to assign the correct POS tags for each ambiguity scheme and unknown words. A tagset of 23 tags were used for the experiment. An accuracy of 93.5% was reported with a 4-fold cross validation on modestly-sized corpora (around 16,000 words). Another reasonably good accuracy POS tagger for Hindi has been developed using Maximum Entropy Markov Model (Dalal et al. 2007). The system uses linguistic suffix and POS categories of a word along with other contextual features. They use the same tagset as in Smriti et al. 2006 and an annotated corpus for training the system. The average per word tagging accuracy of 94.4% and sentence accuracy of 35.2% were reported with a 4-fold cross validation.

In 2006, two machine learning contests were organized on part-of-speech tagging and chunking for Indian Languages for providing a platform for researchers to work on a common problem. Both the contests were conducted for three different Indian languages: Hindi, Bengali and Telugu. All the languages used a common tagset of 27 tags. The results of the contests give an overall picture of the Indian language POS tagging. The first contest was conducted by NLP Association of India (NLP AI) and IIIT-Hyderabad in the summer of 2006. A summary of the approaches and the POS tagging accuracies by the participants are given in Table 1.

In the NLP AI-2006 contest, each participating team worked on POS tagging for a single language of their choice. It was thus not easy to compare the different approaches. Keeping this in mind, the Shallow Parsing for South Asian Languages (SPSAL) contest was held for a multilingual POS tagging and chunking, where the participants developed a common approach for a group of languages. The contest was conducted as a workshop in the IJCAI 2007. Table 2 lists the approaches and the POS tagging accuracy achieved by the teams for Hindi, Bengali and Telugu.

Team	Language	Affiliation	Learning Algo	POS Tagging Accuracy (%)		
				Prec.	Recall	$F_{\beta=1}$
Mla	Bengali	IIT-Kgp	HMM	84.32	84.36	84.34
iitb1	Hindi	IIT-B	ME	82.22	82.22	82.22
Indians	Telugu	IIIT-Hyd	CRF, HMM, ME	81.59	81.59	81.59
litmcsa	Hindi	IIT-M	HMM and CRF	80.72	80.72	80.72
Tilda	Hindi	IIIT-Hyd	CRF	80.46	80.46	80.46
ju_cse_beng	Bengali	JU,Kolkata	HMM	79.12	79.15	79.13
Msrindia	Hindi	Microsoft	HMM	76.34	76.34	76.34

Table 1: Summary of the approaches and the POS tagging accuracy in the NLP AI machine learning contest

Team	Affiliation	Learning Algo	POS Tagging Accuracy (%)		
			Bengali	Hindi	Telugu
Aukbc	Anna University	HMM+rules	72.17	76.34	53.17
HASH	IIT-Kharagpur	HMM(TnT)	74.58	78.35	75.27
litmcsa	John Hopkins University	HMM(TnT)	69.07	73.90	72.38
Indians	IIIT-Hyderabad	CRF+TBL	76.08	78.66	77.37
JU_CSE_BEN G	Jadavpur University	Hybrid HMM	73.17	76.87	67.69
Mla	IIT-Kharagpur	ME + MA	77.61	75.69	74.47
Speech_iit	IIIT-Hyderabad	Decision Tree	60.08	69.35	77.20
Tilda	IIIT-Hyderabad	CRF	76.00	62.35	77.16

Table 2: Summary of the approaches and the POS tagging accuracy in the SPSAL machine learning contest

Although the teams mostly used Hidden Markov Model, Maximum Entropy and Conditional Random Field based models, but different additional resources (e.g. *un-annotated corpus*, *a lexicon with basic POS tags*,

morphological analyzer, named entity recognizer) were used during learning. This might be the reason for achieving different accuracies (tested on a single reference set) for the same learning algorithm using the same training corpora.

2.6. Acknowledgement

Some parts of the information appearing in the survey have been borrowed from previously reported good introductions and papers about POS tagging, the most important ones of which are (Brill, 1995; Dermatas and Kokkinakis, 1995; Marquez and Pedro, 1999).

Chapter 3

Foundational Considerations

In this chapter we discuss several important issues related to the POS tagging problem, which can greatly influence the performance of a tagger. Two main aspects of measuring the performance of a tagger are *the process of evaluation* and *comparison of taggers*. Tagset is the most important issue which can affect the tagging accuracy.

Another important issue of POS tagging is collecting and annotating corpora. Most of the statistical techniques rely on some amount of annotated data to learn the underlying language model. The sizes of the corpus and amount of corpus ambiguity have a direct influence on the performance of a tagger. Finally, there are several other issues e.g. how to handle unknown words, smoothing techniques which contribute to the performance of a tagger.

In the following sections, we discuss three important issues related to POS tagging. The first section discusses the process of corpora collection. In Section 2 we present the tagset which is used for our experiment and give a general overview of the effect of tagset on the performance of a tagger. Finally, in section 3 we present the corpus that has been used for the experiments.

3.1. Corpora Collection

The compilation of raw text corpora is no longer a big problem, since nowadays most of the documents are written in a machine readable format and are available on the web. Collecting raw corpora is a little more difficult problem in Bengali (might be true for other Indian languages also) compared to English and other European languages. This is due to the fact that many different encoding standards are being used. Also, the number of Bengali documents are available in the web is comparatively quite limited.

Raw corpora do not have much linguistic information. Corpora acquire higher linguistic value when they are annotated, that is, some amount of linguistic information (part-of-speech tags, semantic labels, syntactic analysis, named entity etc.) is embedded into it.

Although, many corpora (both raw and annotated) are available for English and other European languages but, we had no tagged data for Bengali to start the POS tagging task. The raw corpus developed at CIIL was available to us. The CILL corpus was developed as a part of the EMILLE¹ project at Central institute Indian Languages, Mysore. We used a portion of the CIIL corpus to develop the annotated data for the experiments. Also, some amount of raw data of the CILL corpora was used for semi-supervised learning.

3.2. The Tagset

With respect to the tagset, the main feature that concerns us is its granularity, which is directly related to the size of the tagset. If the tagset is too coarse, the tagging accuracy will be much higher, since only the important distinctions are considered, and the classification may be easier both by human manual annotators as well as the machine. But, some important information may be missed out due to the coarse grained tagset. On the other hand, a too fine-grained tagset may enrich the supplied information but the performance of the automatic

¹ <http://www.lancs.ac.uk/fass/projects/corpus/emille/>

POS tagger may decrease. A much richer model is required to be designed to capture the encoded information when using a fine grained tagset and hence, it is more difficult to learn.

Even if we use a very fine grained tagset, some fine distinction in POS tagging can not be captured only looking at purely syntactic or contextual information, and sometimes pragmatic level.

Some studies have already been done on the size of the tagset and its influence on tagging accuracy. Sanchez and Nieto (Sanchez and Nieto, 1995) in their work proposed a 479 tag tagset for using the Xerox tagger on Spanish, they latter reduced it to 174 tags as the earlier proposal was considered to be too fine grained for a probabilistic tagger.

On the contrary, Elworthy (Elworthy et al., 1994) states that the sizes of the tagset do not greatly affect the behaviour of the re-estimation algorithms. Dermatus and Kokkinakis (Dermatus and Kokkinakis, 1995), in their work, presented different POS taggers on different languages (Dutch, English, French, German, Greek, Italian and Spanish), each with two different tagsets. Finally, the work in (Teufel et al., 1996) present a methodology for comparing taggers which takes into account the effect of tagset on the evaluation of taggers.

So, when we are about to design a tagset for the POS disambiguation task, some issues needs to be considered. Such issues include – the type of applications (some application may required more complex information whereas only category information may sufficient for some tasks), tagging techniques to be used (statistical, rule based which can adopt large tagsets very well, supervised/unsupervised learning). Further, a large amount of annotated corpus is usually required for statistical POS taggers. A too fine grained tagset might be difficult to use by human annotators during the development of a large annotated corpus. Hence, the availability of resources needs to be considered during the design of a tagset.

During the design of the tagset for Bengali, our main aim was to build a small but clean and completely tagged corpus for Bengali. Other than conventional usages, the resources will be used for machine translation (*hf.* MT) in Indian languages. The tagset for Bengali has been designed considering the traditional grammar and lexical diversity. Unlike Penn Tree bank tagset, we don't use separate tags for the different inflections of a word category.

We have used Penn tagset as a reference point for our tag set design. The Penn Tree bank tagging guidelines for English (Santorini, 1990) proposed a set of 36 tags, which is considered to be one of the standard tagsets for English. However, the number and types of tags required for POS tagging vary from language to language. There is no consensus on the number of tags and it can vary from a small set of 10 tags to as much as 1000 tags. The size of the tagset also depends on the morphological characteristics of the language. Highly inflectional languages may require larger number of tags. In an experiment with Czech (Hladka and Ribarvo, 1998), Haldka and Ribarov showed that the size of the tagset is inversely related to the accuracy of the tagger. However, a tagset which has very few tags cannot be of much use to top level modules like the parser, even if it is very accurate. Thus there is a trade off. In (Ribarvo, 2000; Hladka and Ribarvo, 1998), the authors concluded that for Czech the ideal tagset size should be between 30 and 100. In the context of Indian languages, we did not know of many works on tagset design when we started the work. The LTRC group has developed a tagged corpus called *AnnCora* (Bharati et al., 2001) for Hindi. However, the tagging conventions are different from standard POS tagging. *AnnCora* uses both semantic (e.g. *kAraka* or case relation) and syntactic tags. It is understood that the determination of semantic relations is possible only after parsing a sentence. Therefore, they use a syntactico-semantic parsing method – the *Paninian* approach. They have around 20 relations (semantic tags) and 15 node level tags or syntactic tags. Subsequently, a common tagset has been designed for POS tagging and chunking for a large group of the Indian languages. The tagset consist of 26 lexical tags. The tagset was designed based

on the lexical category of a word. However, some amount of semantic information may needs to be considered during the annotation especially, in the case of labelling main verb (VM) and auxiliary verb (VAUX) for Bengali. Table 3 describes the different lexical categories and used in our experiments. A detailed description of individual tags with examples has been provided in Appendix A.

Tag	Description	Tag	Description	Tag	Description
ADV	Adverb	NEG	Negative particle	RPP	Personal relative pronoun
AVB	Adverbial particle/verbal particle	NN	Default noun/common noun	RPS	Spatial relative pronoun
CND	Conditional	NP	Proper noun	RPT	Temporal relative pronoun
CNJ	Conjunction	NUM	Number	SEN	Sentinel
DTA	Absolute determiner	NV	Verbal noun	SHD	Semantic shades incurring particle
DTR	Relative Determiner	PC	Cardinal pronoun	SYM	Symbol
ETC	Continuation Marke/Ellipsis Marker	PO	Ordinal pronoun	TO	Clitic
FW	Foreign word	PP	Personal pronoun	VF	Finite verb
INT	Interjection	PPI	Inflectional post position	VIS	Imperative/subjunctive verbs
JF	Following Adjectives	PPP	Possessive post position	VM	Modal verb
JJ	Noun-qualifying adjectives	PQ	Question marker	VN	Non-finite verb
JQC	Cardinal qualifying adjectives	PS	Spatial pronoun	VNG	Verb Negative
JQH	Hedged expression	PT	Temporal pronoun		
JQQ	Quantifier	QUA	Qualifier		

Table 3: The tagset for Bengali with 40-tags

The tagset used for our experiment is purely syntactic because we consider POS tagging an independent form parsing; rather the first step before parsing can be done only after the completion of tagging. Some ambiguity that cannot be resolved at the POS tagging level will be propagated to the higher level. We are

following the tagging convention as specified by the Penn-tree bank project. According to this convention tags are all in capital letters and of length two to three. The tag follows the word in question separated by a ‘\’ (back slash) immediately after the word. There are no blank spaces in between. After the tag there should be at least one blank (white space) before the next character, which can be either a word or a sentinel. The following sentence illustrates the convention (it is in the ITRANS notation (Chopde, 2001)).

itimadhye\ADV Aguna\NN nebhAnora\NV lokao\NN ese\VN gela\VF .\SEN
/mean time/ /fire/ [/put off/] /men/ /come/ /have/
In the mean time firemen arrived

We are using a tagset of 40 grammatical tags. The tagset used here is purely syntactic.

3.3. Corpora and Corpus Ambiguity

In this section we describe the corpora that have been used for all the experiments in this thesis. We also describe some properties of the corpora which have a direct influence on the POS tagging accuracy as well as the comparison of taggers.

The hardness of the POS tagging is due to the ambiguity in language as described in section 1.1. The ambiguity varies from language to language and also from corpus to corpus. Although it has been pointed out that most of the words in a language vocabulary (*types*) are unambiguous, a large percentage of the words in a corpus (*tokens*) are ambiguous. This is due to the fact that the occurrences of the high frequency words (most common words) are ambiguous. DeRose (DeRose, 1988) pointed out that 11.5% types (shown in Table 4) and 40% tokens are ambiguous in the Brown corpus for English. A similar study has been conducted for Bengali to find out the degree of ambiguity in both types and tokens in the corpus. We had no such large corpora to find out the degree of

ambiguity like Brown corpus of English. Instead, we use a Morphological Analyzer (MA) for Bengali to find out the possible tags of a given word. Please note that the MA used for Bengali operates on the same cardinality of the tagset (described in the previous section). We used the whole CIIL corpora to find out the degree of ambiguity for Bengali. It has been observed that 10% of the types are ambiguous, which is lesser than the Brown corpus. However, 42% of the tokens in the CIIL corpus are ambiguous which is higher than the English Brown corpus. Table 5 gives the tag ambiguity for Bengali CIIL corpus. This implies that perhaps the POS disambiguation task for Bengali will be more difficult compared to English.

Per Word Tags	No. of Words
1 tag	35,340
2 tags	3,760
3 tags	264
4 tags	61
5 tags	12
6 tags	2
7 tags	1

Table 4: Tag ambiguity of word types in Brown corpus (DeRose , 1988)

Per Word Tags	No. of Words
1 tag	41,719
2 tags	3,149
3 tags	630
4 tags	504
5 tags	256
6 tags	33

Table 5: Tag ambiguity of word types in Bengali CIIL corpus

Another important issue about the Indian languages is morphological richness. Morphological richness can also be considered to be an important factor for POS tagging accuracy and comparison of taggers. Bengali is a highly agglutinative language. So, the vocabulary (unique words) grows at a higher rate as we increase the size of the corpus. Figure 4 plots the vocabulary growth for Bengali and Hindi along with the increment of the size of the corpus (CIIL corpus). As a matter of fact, different surface forms (*token*) appear for a particular lexical item (*type*), which essentially may not increment the number of observing a token. This may affect the counting base stochastic algorithm (e.g. HMM, ME etc.). Thus, it might be the case that the POS tagging task in Bangla is difficult

compared to Hindi under the same experimental setup (amount of training data and learning algorithm).

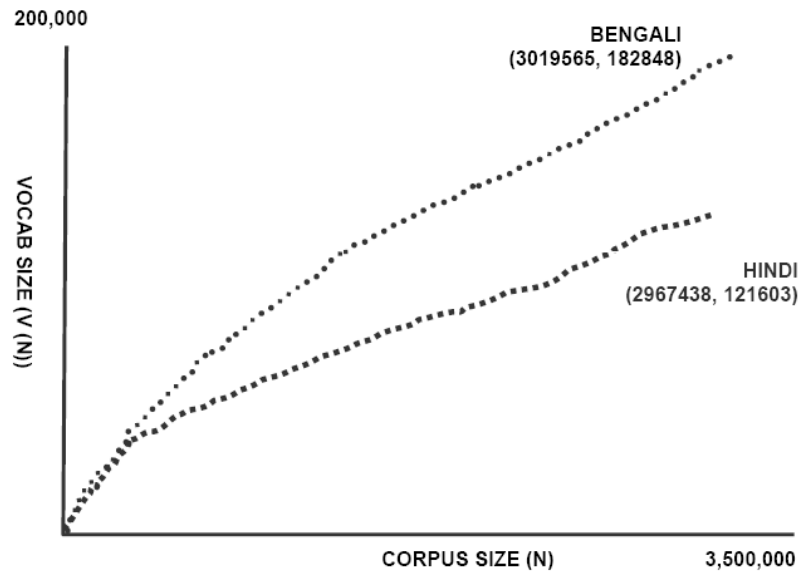


Figure 4: Vocabulary growth of Bengali and Hindi

3.3.1. Data Used for the Experiments

The training data includes manually annotated 3625 sentences (approximately 40,000 words) for all the models. A fixed set of 11,000 unlabeled sentences (approximately 100,000 words) taken from the CIIL corpus is used to re-estimate the model parameter during semi-supervised HMM learning.

All the models have been tested on a set of randomly drawn 400 sentences (5000 words) disjoint from the training corpus. It has been noted that 14% words in the open testing text are unknown with respect to the training set, which is also a little higher compared to the European languages (Dermatas and Kokkinakis, 1995).

The *corpus ambiguity* is defined as the mean number of possible tags for each word of the corpus. It has been observed that the corpus ambiguity in the training text is 1.77 which is much larger compared to the European languages

(Dermatas and Kokkinakis, 1995). Table 6 shows the comparison of corpus ambiguity for 5 different languages.

Language	Dutch	German	English	French	Bengali
Corpus Ambiguity	1.11	1.3	1.34	1.69	1.77
Accuracy	96%	97%	96.5%	94.5%	?
Unknown Words	13%	9%	11%	5%	14%

Table 6: Corpus ambiguity, Tagging accuracy and percentage of unknown word (open testing text) for different language corpora used for POS tagging

Dermatas has shown in his paper (Dermatas and Kokkinakis, 1995), that the tagging accuracy of English is relatively higher compared to French though French has smaller number of unknown words in the open testing text. This may be one of the reasons of relatively lesser accuracy of the Bengali tagging task.

Chapter 4

Tagging with Hidden Markov Model

In this chapter we describe a Hidden Markov Model (HMM) based stochastic algorithm for POS tagging. HMM is the most successfully used simple language model (*n-gam*) for POS tagging that uses very little amount of knowledge about the language, apart from simple contextual information. Since only a small labeled training set is available to us for Bengali POS tagging, a simple HMM based approach does not yield very good results. In our particular work, we have used a morphological analyzer to improve the performance of the tagger. Further, we have made use of semi-supervised learning by augmenting the small labeled training set with a larger unlabeled training set.

The organization of the chapter is as follows: Section 1 describes some basic definitions and notation of the HMM model. Section 2 devoted to our particular approach to Bengali POS tagging using HMM. Section 3 describes the different experiment conducted for the task. Section 4 presents the experimental results and assessment of error types and Section 5 provides the conclusion.

4.1. Hidden Markov Model

A Hidden Markov Model (HMM) is a statistical construct that can be used to solve classification problems that have an inherent state sequence representation.

The model includes an interconnected set of *states* which are connected by a set of *transition probabilities*. Transition probabilities indicate the probability of traveling between two given states. A process starts at a particular state and moves to a new state as governed by the transition probabilities in discrete time intervals. As the process enters into a state one of a set of *output symbol* (also known as *observation*) is emitted by the process. The symbol emitted, is dependent on the probability distribution of the particular state. The output of the HMM is a sequence of output symbols. In an HMM, the exact state sequence corresponding to a particular observation sequence is unknown (i.e. *hidden*).

4.1.1. Basic Definitions and Notation

According to Rabiner (Rabiner, 1989), five elements are required to be defined in an HMM. Figure 5 represents the five tuple of an HMM.

1. The number of distinct states (N) in a model. We denote the individual state as $S = \{S_1, S_2, \dots, S_N\}$. In case of Part-of-speech tagging, N is the number of tags in the tagset $\{T\}$ that will be used by the system. Each tag in the tagset corresponds to one state in the HMM.
2. The number of distinct output symbols (M) in the HMM. We denote the individual symbol as $V = \{v_1, v_2, \dots, v_M\}$. For Part-of-Speech tagging, M is the number of words in the lexicon of the system.
3. The state transition probabilities $A = \{a_{ij}\}$. The probability a_{ij} , is the probability of moving state i to j in one transition. In part-of-speech tagging the states correspond to tags, so a_{ij} is the probability that the model will move from tag t_i to t_j (where $t_i, t_j \in \{T\}$). In other words, a_{ij} is the probability that t_j follows t_i (i.e. $P(t_j | t_i)$). This probability is usually estimated from the annotated training corpus during training.

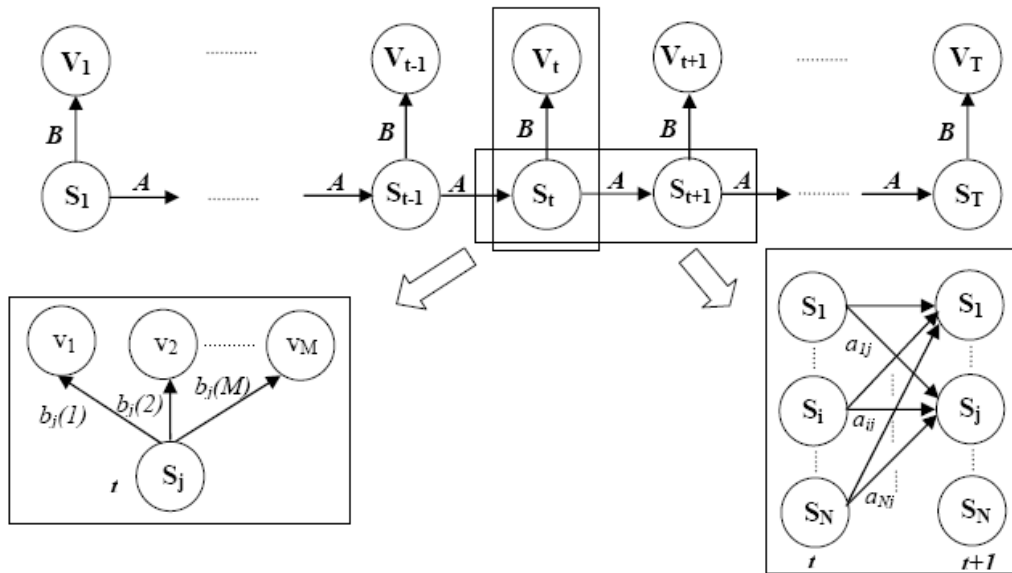


Figure 5: General Representation of an HMM

4. The observation symbol probability $B = \{b_j(k)\}$. The probability $b_j(k)$ denotes the probability that the k -th output symbol will be emitted when the model is in state j . For POS tagging, this is the probability that the word w_k will be emitted when the process is in state t_j (i.e. $P(w_k | t_i)$). This probability can also be estimated from the training corpus.
5. $\pi = \{\pi_i\}$, the initial state distribution. π_i is the probability that the model will start at state i . For POS tagging, this is the probability that the sentence will begin with a particular tag t_i

When using an HMM to perform POS tagging, the aim is to determine the most likely tag (states) sequence that generates the words of a sentences (the sequence of output symbols). In other words, we calculate the sequence of tags (S) given a sentence (W) that maximizes $P(W | S)$. The Viterbi (Viterbi, 1967) algorithm can be used to find out the most likely tag sequence. The algorithm will be discussed in brief in the subsequent sections.

4.2. Our Approach

We have used an HMM for automatic POS tagging of natural language text. As described in chapter 1, we distinguish between three main components in our system. The three components of the HMM based tagger are depicted in Figure 6. First, the system requires some knowledge about the task of POS disambiguation. The knowledge may come from several resources and can be encoded in various representations. We call this representation as *language model*. In particular to HMM, the language model is represented by the model parameters $\mu = (\pi, A, B)$. We aim to estimate the model parameters $\mu = (\pi, A, B)$ of the HMM using corpora. The model parameters of the HMM are estimated based on the labeled data during supervised learning. Unlabelled data are used to re-estimate the model parameters during semi-supervised learning. The model parameters are re-estimated using Baum-Welch algorithm. The taggers will be implemented based on both bigram and trigram HMM models.

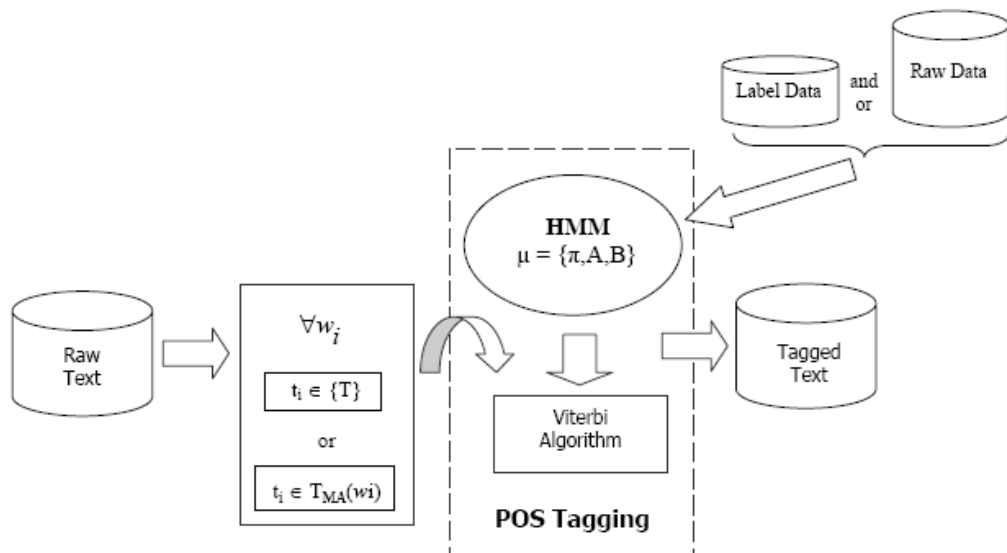


Figure 6: The HMM based POS tagging architecture

Secondly, there is a *disambiguation algorithm*, which decides the best possible tag assignment for every word in a sentence according to the language model. We use Viterbi algorithm for disambiguation. The third component estimates the set of possible tags $\{T\}$, for every word in a sentence. We shall call this as

possible class restriction module. This module consists of a list of lexical units associated with the list of possible tags. In our approach we first assume that every word can be associated with all the tags in the tagset (i.e. *a set of 40 tags in the tagset* $\{T\}$). Further, we assume the POS tag of a word w can take the values from the set $T_{MA}(w)$, where $T_{MA}(w)$ is computed by the Morphological Analyzer. These three components are related and we combine them into a single tagger description. The input to the disambiguation algorithm takes the list of lexical units with the associated list of possible tags. The disambiguation module provides the output tag for each lexical unit using the encoded information from the language model. The following subsections give a detailed design of the above three components in our work.

4.2.1. Models

There are several ways of representing the HMM based model for automatic POS tagging according to the way we acquire knowledge. The HMM models use the following three sources of information.

- (1) *Symbol emission probabilities*, i.e. the probability of a particular tag t_i , given a particular word w_i , $P(w_i | t_i)$.
- (2) *State transition probabilities*, i.e. the probability of a particular tag depending on the previous tags, $P(t_i | t_{i-1}t_{i-2}\dots t_{i-k})$.
- (3) *Probability for the initial state*, i.e. the probability of a particular tag as an initial state of a Markov model

The above parameters can be estimated using only labeled data during supervised learning. We shall call this model **HMM-S**. Further, semi-supervised learning can be performed by augmenting the labeled data with additional unlabelled data. We shall call this model **HMM-SS**.

The state transition probabilities are often estimated based on previous one (first-order or bigram) or two (second-order or trigram) tags. Depending on the order of the symbol transition probability we shall call the Markov process as first-order (**HMM1**) and second-order (**HMM2**) Markov process respectively. We adopt four different Markov models for representing the language model: (1)

Supervised first-order HMM (HMM-S1) (2) *Semi-supervised first-order HMM (HMM-SS1)* (3) *Supervised second-order HMM (HMM-S2)* (4) *Semi-supervised second-order HMM (HMM-SS2)*.

Supervised HMM (HMM-S)

In this model the model parameters are estimated using only labeled training data. In a k -th order Markov model, the state transition probability of a particular tag t_i depends on the previous $k-1$ tags in the sequence, $P(t_i | t_{i-1} t_{i-2} \dots t_{i-k+1})$. In the *supervised first-order HMM* (HMM-S1), the state transition probability of a particular tag t_i depends only on the previous tag t_{i-1} (i.e. $P(t_i | t_{i-1})$). The symbol emission and state transition probabilities are estimated directly from the labeled training data as follows.

$$P(t_i | t_{i-1}) = \frac{C(t_{i-1}, t_i)}{C(t_{i-1})} \text{ and } P(w_i | t_i) = \frac{C(w_i, t_i)}{C(t_i)}, \text{ where } C(\) \text{ denotes the number of}$$

occurrence in the labeled training data. As we are dealing with a small labeled corpora it is often possible that $C(t_{i-1}, t_i)$ and $C(w_i, t_i)$ will become zero. To cope with the above situation, state transition probabilities are smoothed and symbol emission probabilities are estimated for handling unknown words that are not in the labeled corpora (see sub-section 4.2.3).

Like supervised first-order HMM (HMM-S1), the model parameters of the supervised second-order HMM (HMM-S2) are also estimated simply by counting from the labeled training data. Here the state transition probabilities of a particular tag t_i depends on the previous two tags t_{i-1} and t_{i-2} , $P(t_i | t_{i-2}, t_{i-1})$. Experiments have been carried out with TnT tagger (Brants, 2000); a supervised trigram HMM tagger along with suffix tree information for unknown words. When a particular instance of a trigram state transition probability does not occur in the training data the state transition probabilities are smoothed and the symbol emission probabilities for unknown words are computed using the

probability distribution for a particular suffix generated from all words in the labeled corpora (Brants, 2000).

Semi-supervised HMM (HMM-SS)

In semi supervised first-order HMM, we first make use of the labeled training data to train the initial model. Further we make use of semi-supervised learning by augmenting the labeled data with a large amount of unlabelled data. The semi-supervised learning uses Baum-Welch re-estimation (or equivalently the *expectation maximization* (EM)) algorithm by recursively defining two sets of probabilities, the forward probabilities and the backward probabilities. First we determine the initial choice for model parameters A , B and π from the labeled data. After choosing the above starting values, we iteratively use Baum-Welch algorithm to compute the new values of model parameters until convergence.

Baum Welch, or forward backward algorithm, recursively define two sets of probabilities. The forward probabilities,

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^N \alpha_t(i) a_{ij} \right] b_j(w_{t+1}) \quad 1 \leq t \leq T, \text{ (where } \alpha_1(i) = \pi_i b_i(w_1) \text{ for all } i),$$

and the backward probabilities,

$$\beta_t(i) = \sum_{j=1}^N a_{ij} b_j(w_{t+1}) \beta_{t+1}(j) \quad T-1 \leq t \leq 1, \text{ (where } \beta_T(j) = 1 \text{ for all } j).$$

The forward probability $\alpha_t(i)$ is the joint probability of the sequence up to time t , $\{w_1, w_2, \dots, w_t\}$ and the Markov process is in state i at time t . Similarly, the backward probability $\beta_t(j)$ is the probability of seeing sequence $\{w_{t+1}, w_{t+2}, \dots, w_T\}$ and the Markov process is in state j at time t . It follows the probability of the entire sequence is

$$P = \sum_{i=1}^N \sum_{j=1}^N \alpha_t(i) a_{ij} b_j(w_{t+1}) \beta_{t+1}(j) \quad \text{for ant } t \text{ in the range } 1 \leq t \leq T-1.$$

After the initial choice of the model parameters $\mu = (\pi, A, B)$ from the training data, the expected number of transition γ_{ij} from state i to j conditions on the observation sequence W is computed as follows:

$\gamma_{ij} = \frac{1}{P} \sum_{t=1}^{T-1} \alpha_t(i) a_{ij} b_j(w_{t+1}) \beta_{t+1}(j)$, which is expected number of transition from state i to j .

Hence, the expected transition probability from a particular state i to a particular state j (i.e. \hat{a}_{ij}) are estimated by:

$$\hat{a}_{ij} = \frac{\gamma_{ij}}{\sum_{j=1}^N \gamma_{ij}} = \frac{\sum_{t=1}^{T-1} \alpha_t(i) a_{ij} b_j(w_{t+1}) \beta_{t+1}(j)}{\sum_{t=1}^{T-1} \alpha_t(i) \beta_t(i)} \quad \text{Eq. 2}$$

In particular to POS tagging, the above probability is the ratio of the expected number of transitions from a particular tag t_i to another particular tag t_j and the total expected number of transition from tag t_i to t_j .

Similarly, the emission probability (i.e. $\hat{b}_j(k)$) and initial probability (i.e. $\hat{\pi}_i$) can be estimated as follows:

$$\hat{b}_j(k) = \frac{\sum_{t=1}^T \mathbb{1}_{W_t = w_k} \alpha_t(j) \beta_t(j)}{\sum_{t=1}^T \alpha_t(j) \beta_t(j)} \quad \text{Eq. 3}$$

and

$$\hat{\pi}_i = \frac{1}{P} \alpha_1(i) \beta_1(i) \quad \text{Eq. 4}$$

The Baum Welch algorithm uses EM algorithm. Starting from at the initial model $\mu = (\pi, A, B)$ obtained by the supervised learning using the small annotated data, we repeatedly compute the new values $\{\hat{\mu} = (\hat{\pi}, \hat{A}, \hat{B})\}$ applying the equation 2-4 until convergence. It has been shown that the algorithm will converge, possibly to a non global local maximum.

4.2.2. Disambiguation

The aim of the disambiguation algorithm is to assign the most probable tag sequence $t_1 \dots t_n$, to a observed sequence of words $w_1 \dots w_n$, that is

$$S = \arg \max_{t_1 \dots t_n} P(t_1 \dots t_n | w_1 \dots w_n)$$

The stochastic optimal sequence of tags $t_1 \dots t_n$, are assigned to the word sequence $w_1 \dots w_n$, can be expressed as a function of both lexical $P(w_i | t_i)$ and language model $P(t_i | t_{i-1})$ probabilities using Bayes' Theorem:

$$\begin{aligned} P(t_1 \dots t_n | w_1 \dots w_n) &= \frac{P(w_1 \dots w_n | t_1 \dots t_n) P(t_1 \dots t_n)}{P(w_1 \dots w_n)} \\ &= \frac{\prod_{i=1, n} P(w_i | t_i) P(t_i | t_{i-1})}{P(w_1 \dots w_n)} \end{aligned}$$

Since the probability of the word sequence $P(w_1 \dots w_n)$ is the same for all candidate tag sequences, the most probable tag sequence (S) satisfies:

$$S = \arg \max_{t_1 \dots t_n} \prod_{i=1, n} P(w_i | t_i) P(t_i | t_{i-1}) \quad \text{Eq. 5}$$

We use the Viterbi algorithm to find out the most probable tag sequence for a given word sequence based on equation 5. It is a very effective dynamic programming algorithm which takes $O(TN^2)$ time. The algorithm works as follows:

Let $S = \{s(t)\} \ 1 \leq t \leq T$ is a state sequence (i.e. the tag sequence) that generates $W = \{w(t)\}$ (the word sequence or observation of the HMM). Then the probability that S generates W is,

$$P(S) = \pi_{s(1)} b_{s(1)}(w_1) \prod_{t=2}^T a_{s(t-1)s(t)} b_{s(t)}(w_t)$$

To find the most probable sequence, the process starts with $\phi_1(i) = \pi_i b_i(w_1)$ where $1 \leq i \leq N$, and then performs the following steps:

$$\left. \begin{aligned} \phi_t(j) &= \max_{1 \leq i \leq N} [\phi_{t-1}(i) a_{ij} b_j(w_t)] \\ \text{and} \\ \psi_t(j) &= \arg \max_{1 \leq i \leq N} [\phi_{t-1}(i) a_{ij} b_j(w_t)] \end{aligned} \right\} \begin{array}{l} 2 \leq t \leq T \\ 1 \leq j \leq N \end{array}$$

The most probable sequence at state i in time t is the only consideration for each time t and state i . The probability of the most probable sequence is $\max_{1 \leq i \leq N} [\phi_T(i)]$. The most probable sequence is reconstructed by $s(T) = \arg \max_{1 \leq i \leq N} \phi_T(i)$ and $s(t-1) = \psi_t(s_t)$ for $T \geq t \geq 2$.

We approach the problem of finding most probable tag sequence in three different ways:

- (1) The first model uses a set of 40 tags for each word (w_i) in a test sentence and the most probable tag sequence is determined using a dynamic programming for all the models described in the previous section.
- (2) In order to further improve tagging accuracy, we integrate morphological information with the above models. We assume that the POS tag of a word w can take the values from the set $T_{MA}(w)$, where $T_{MA}(w)$ is computed by the Morphological Analyzer (Maitra, 2004) which we call as the *possible class restriction* module. Note that the size of $T_{MA}(w)$ is much smaller than T . Thus, we have a restricted choice of tags as well as tag sequences for a given sentence. Since the correct tag t for w is always in $T_{MA}(w)$, it is always possible to find out the correct tag sequence for a sentence even after applying the morphological restriction. Due to the much reduced set of possibilities, this model performs better for both the HMM models even

when a small amount of labeled training data is available. We shall call these new models **HMM-S1 + MA**, **HMM-SS1 + MA**, **HMM-S2 + MA** and **HMM-SS2 + MA** respectively.

- (3) It is also possible that due to an incomplete wordlist, all words are not included to the Morphological Analyzer, and the set $T_{MA}(w)$ may be empty. In this situation, the above method fails and in this case, we assume the POS tags of a word w can take values from the set of $T_O(w)$, where $T_O(w)$ denotes the set of open class grammatical categories.

Figure 7 illustrates this disambiguation procedure. The top figure shows the Viterbi search space of the HMM based graphical model. The numbers of states for each observation (i.e. the word) is equals to the size of the tagset.

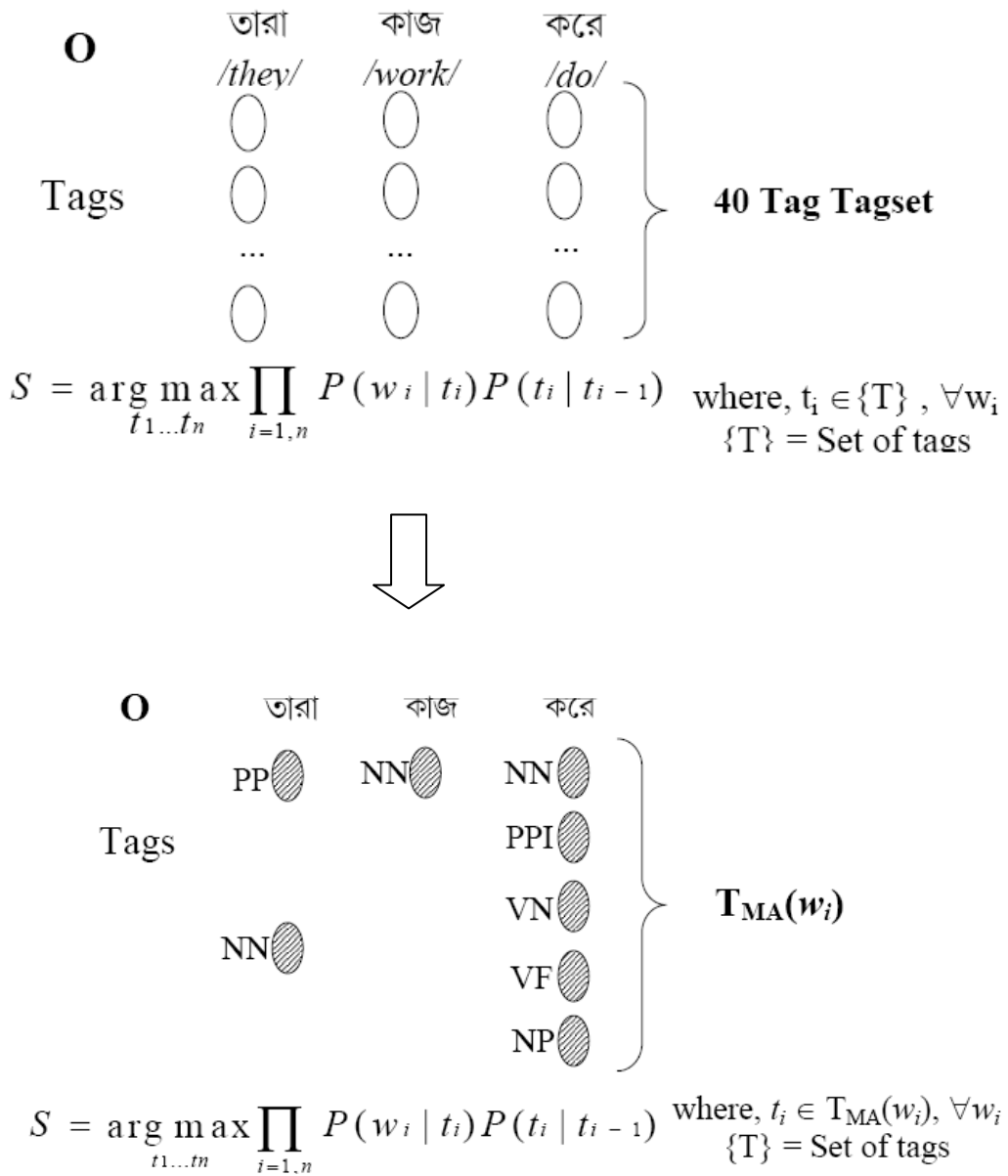


Figure 7: Uses of Morphological Analyzer during decoding

Further, the use of morphology reduces the Viterbi search space as depicted in bottom of the figure. Here the numbers of the states is restricted to the number of possible tags given the morphological analyzer. For example, in the example in figure 7, the word *tArA (/they/)* has only two possible tags (personal pronoun (PP) and common noun (NN)) whereas the word *kare (/do/)* has five possible tags (common noun (NN), post position (PPI), finite verb (VF), non-finite verb (VN) and proper noun (NP)). When a word is unknown to the morphological analyzer, we considered that the possible tags can be any of the open class grammatical category (i.e. all class of noun, verbs, adjective, adverbs and interjections). In summary, we get a much reduced size of viterbi search space which one sequence that represent the actual tag assignment out of all possible assignment.

Our MA has high accuracy and coverage but it still has some missing words and a few errors. For the purpose of these experiments we have made sure that all words of the test set are present in the root dictionary that the MA uses.

While MA helps us to restrict the possible choice of tags for a given word, one can also use suffix information (i.e., the sequence of last few characters of a word) to further improve the models. For HMM models, suffix information has been used during smoothing of emission probabilities. We shall denote the models with suffix information with a ‘+suf’ marker. Thus, we have – **HMM-S1+suf**, **HMM-S1+suf+MA**, **HMM-SS1+suf** etc.

4.2.3. Smoothing

It may be the case that all events are not encountered in the limited training corpus that we have. The probabilities corresponding to these events would be set to zero. However the event may occur during testing. The problem can be solved using different smoothing algorithms. Initially simple add-one smoothing was used to estimate the state transition probabilities that are not in the training corpora. Further, linear interpolation of unigram and bigram has

been implemented for smoothing the state transition probabilities. We smooth the n -gram state transition probability for various n as follows:

$$P_{ii}(t_i | t_{i-1}, t_{i-2}, \dots, t_{i-(n-1)}) = \lambda_1 P(t_i) + \lambda_2 P(t_i | t_{i-1}) + \dots + \lambda_n P(t_i | t_{i-1}, t_{i-2}, \dots, t_{i-(n-1)})$$

The values of $\lambda_1, \lambda_2, \dots, \lambda_n$ are estimated by deleted interpolation (Brants, 2000) and $\sum_{i=1}^n \lambda_i = 1$.

When some new text is processed, few words might be unknown to the tagger. In our model, words are unknown when they are not included in the training text. Initially, we estimated the symbol emission probability by simple add-one smoothing. Further, we use suffix information for handling unknown words which has been found to work well for highly inflected languages (Samuelsson, 1993). The term suffix is a sequence of last few characters of a word, which does not necessarily mean a linguistically meaningful suffix. First we calculate the probability of a particular tag t_i , given the last m letters (l_i) of an n letter word: $P(t_i | l_{n-m+1}, \dots, l_n)$. Based on the above hypothesis we calculate the symbol emission probabilities using Bayes' rule:

$$\begin{aligned} P(\text{Unknown_word} | t_i) &= \frac{P(t_i | \text{Unknown_word})P(\text{Unknown_word})}{P(t_i)} \\ &= \frac{P(t_i | l_{n-m+1}, \dots, l_n)P(\text{Unknown_word})}{P(t_i)} \end{aligned}$$

The probability $P(\text{Unknown_word})$ is approximated in open testing text by measuring the unknown word frequency. Therefore the model parameters are adopted each time an open testing text is being tagged. The probability $P(t_i | l_{n-m+1}, \dots, l_n)$ and the probability $P(t_i)$ are measured in the training text. We conducted different experiments varying the suffix length from 1 to 6 characters. It has been observed empirically that the suffix length of 4 gives better results for all the HMM based models. Based on our observations, the inclusion of suffix essentially captures helps to understand the morphological inflection of the surface form word and in Bangla most morphological

inflections lies in the last 4 characters of the words. Finally, each symbol emission probability of unknown word has been normalized:

$$\sum_1^N P(w_i | t_i) + P(\text{Unknown_word} | t_i) = 1$$

Where, N is the number of known words and $t_i \in \{T\}$.

4.3. Experiments

In the first experiment, we have implemented baseline model to understand the complexity of the POS tagging task. In this model the tag probabilities depend only on the current word:

$$P(t_1 \dots t_n | w_1 \dots w_n) = \prod_{i=1, n} P(t_i | w_i)$$

The effect of this is that the each word in the test data will be assigned the tag which occurred most frequently for that word in the training data.

We have a total of eight models (HMM-S1, HMM-S1+suf, HMM-S1+MA, HMM-S1+suf+MA, HMM-SS1, HMM-SS1+suf, HMM-SS1+MA, HMM-SS1+suf+MA) as described in subsection 4.2.2 under bigram HMM based stochastic tagging schemes. The same training text has been used to estimate the parameters for all the models. The model parameters for supervised HMM based models are estimated from the annotated text corpus. For semi-supervised learning, the HMM learned through supervised training is considered as the initial model. Further, a larger unlabelled training data has been used to re-estimate the model parameters of the semi-supervised HMM. The experiments were conducted with three different sizes (10K, 20K and 40K words) of the training data to understand the relative performance of the models as we keep on increasing the size of the annotated data.

We have also carried out some experiments (HMM-S2) with the freely available ACOPOST² tagger and TnT tagger, which is based on a supervised trigram HMM with suffix tree information for unknown words. These

² <http://acopost.sourceforge.net/>

experiments give us some insight about the performance of the tagging task in comparison with the order of the Markov model in a poor-resource scenario.

4.3.1. Training Data

As described in Chapter 3, the training data consists of 3625 manually annotated sentences (approximately 40,000 words). The same training corpus is used for all the HMM based tagging schemes. The training data has been annotated using a tag set consisting of 40 grammatical tags. A fixed set of 11,000 unlabeled sentences (approximately 100,000 words) taken from CIIL corpus are used to re-estimate the model parameter during semi-supervised learning. It has been observed that the corpus ambiguity (mean number of possible tags for each word) in the training text is 1.77 which is much larger compared to the figure reported for European languages (Dermatas et al., 1995).

4.3.2. Test Data

All the models have been tested on a set of randomly drawn 400 sentences (5000 words) distinct from the training corpus. It has been noted that 14% words in the open testing text are unknown with respect to the training set, which is also a little higher compared to the European languages (Dermatas et al., 1995).

4.4. System Performance

We define the tagging accuracy as the ratio of the correctly tagged words to the total number of words.

$$Accuracy(\%) = \frac{\text{Correctly tagged words by the system}}{\text{Total no. of words in the evaluation set}} \times 100$$

Figure 8 shows the improvement in accuracy of each of the models along with the increase in the size of annotated training data when supervised learning algorithm (HMM-S1, HMM-S1+suf, HMM-S1+MA and HMM-S1+suf+MA) have been used to estimate the model parameters. Similarly, Figure 9 represents the improvement in accuracy of each of the semi-supervised models (HMM-SS1+suf, HMM-SS1+suf, HMM-SS1+MA, HMM-SS1+suf+MA) along with the increase in the size of annotated training data.

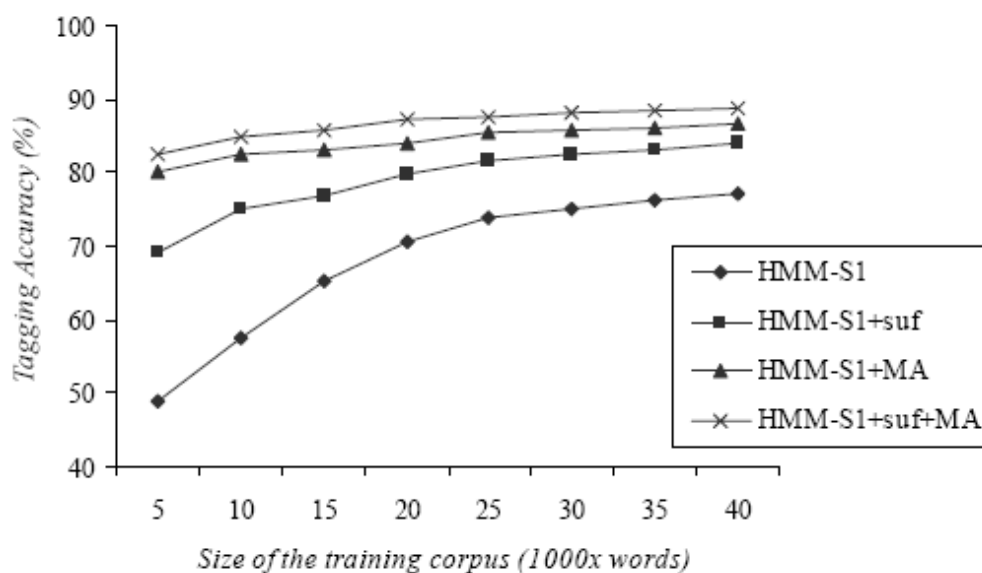


Figure 8: The accuracy growth of different supervised HMM models.

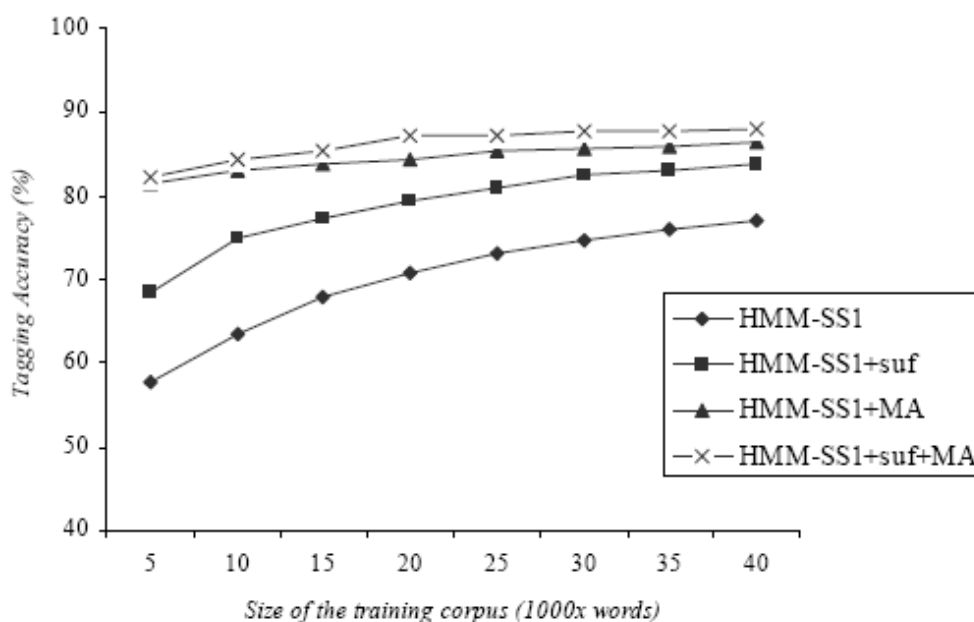


Figure 9: The accuracy growth of different semi-supervised HMM tagging models.

We also measure the known word and unknown word accuracy separately for all the models to understand their performance in a poor resource scenario. It is obvious that the number of unknown words is always high when less amount of annotated data is available. So, the models which better handles the unknown words are considered to be a well fitted model for the POS disambiguation task in a poor resource scenario. Figure 10 shows known and unknown word accuracies under different HMM models.

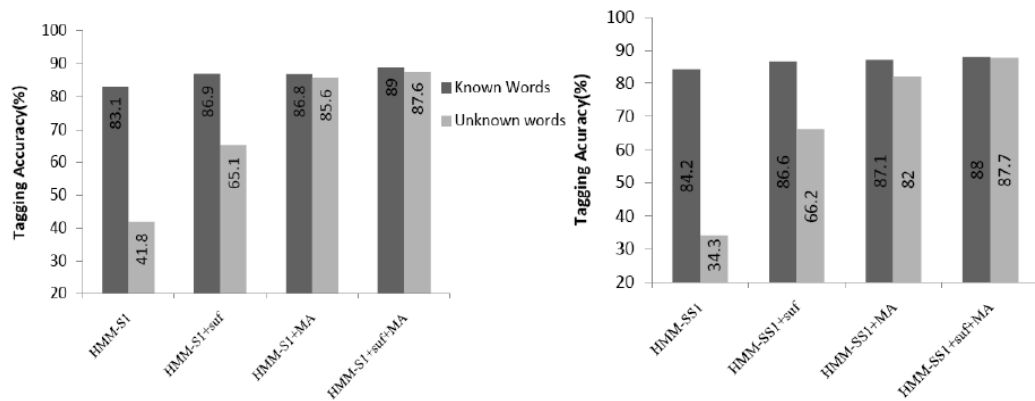


Figure 10: Known and Unknown accuracy under different HMM based models

It is interesting to note that known word accuracy under the supervised HMM models improve (about 4% for both the cases) while suffix or MA (HMM-S1+suf, HMM-S1+MA) is being used in the models over the simple HMM model (HMM-S1). However a combination of both suffix and MA (HMM-S1+suf+MA) gives about 6% improvement compared to the HMM-S1 model. However, the improvement in unknown word accuracy is much higher due to the uses of suffix and/or MA. The uses of suffix (HMM-S1+suf) and MA (HMM-S1+MA) gives an improvement of 25% and 44% respectively compared to the HMM-S1 model. The improvement is much higher (46%) while both suffix and MA are used in the HMM model (HMM-S1+suf+MA). Similar trend has been observed for the semi-supervised HMM models.

Table 7 summarizes the final accuracies achieved by different learning methods with the varying size of the training data. Note that the baseline model

(i.e., the tag probabilities depends only on the current word) has an accuracy of 76.8%.

Method	Accuracy		
	10K	20K	40K
HMM-S1	57.53 (74.19, 29.42)	70.61 (79.42, 40.88)	77.29 (83.07, 41.78)
HMM-S1+suf	75.12 (84.35, 59.54)	79.76 (84.58, 63.53)	83.85 (86.90, 65.13)
HMM-S1+MA	82.39 (85.10, 77.80)	84.06 (84.77, 81.67)	86.64 (86.82, 85.56)
HMM-S1+suf+MA	84.73 (87.81, 79.53)	87.35 (88.29, 84.21)	88.75 (88.95, 87.55)
HMM-SS1	63.40 (80.84, 30.11)	70.67 (82.32, 31.39)	77.16 (84.15, 34.25)
HMM-SS1+suf	75.08 (84.08, 59.88)	79.31 (83.91, 63.81)	83.76 (86.63, 66.21)
HMM-SS1+MA	83.04 (87.09, 76.24)	84.47 (86.00, 79.32)	86.41 (87.12, 82.02)
HMM-SS1+suf+MA	84.41 (87.16, 79.77)	87.16 (87.81, 84.96)	87.95 (88.00, 87.71)

Table 7: Tagging accuracies (%) of different models with 10K, 20K and 40K training data. The accuracies are represented in the form of Overall Accuracy (Known Word Accuracy, Unknown Word Accuracy)

4.4.1. Observations

We find that in both of the HMM based models (**HMM-S1** and **HMM-SS1**), the use of suffix information as well as the use of a morphological analyzer improves the accuracy of POS tagging with respect to the base models. The use of MA gives better results than the use of suffix information. When we use both suffix information as well as MA, the results are even better.

HMM-SS1 does better than **HMM-S1** when very little tagged data is available, for example, when we use 10K training corpus. However, the accuracies of the semi-supervised HMM models are slightly poorer than that of the supervised HMM models for moderate size training data when suffix information is used. We postulate that this discrepancy arises due to the over-fitting of the supervised models in the case of small training data; the problem is alleviated with the increase in the annotated data.

As we have noted already the use of MA and/or suffix information improves the accuracy of the POS tagger. But what is significant to note is that the percentage of improvement is higher when the amount of training data is less. The **HMM-S1+suf** model gives an improvement of around 18%, 9% and 6% over the **HMM-S1** model for 10K, 20K and 40K training data respectively. Similar trends are observed in the case of the semi-supervised HMM and the ME models. The use of morphological restriction (**HMM-S1+MA**) gives an improvement of 25%, 14% and 9% respectively over the **HMM-S1** in case of 10K, 20K and 40K training data. As the improvement due to MA decreases with increasing data, it might be concluded that the use of morphological restriction may not improve the accuracy when a large amount of training data is available. From our empirical observations we found that both suffix and morphological restriction (**HMM-S1+suf+MA**) gives an improvement of 27%, 17% and 12% over the HMM-S model respectively for the three different sizes of training data.

Furthermore, in order to estimate the relative performance of the models, experiments were carried out with two existing taggers: TnT (Brants, 2000) and ACOPOST. The accuracy achieved using TnT are 87.44% and 87.36% respectively with bigram and trigram model for 40K training data. The accuracy with ACOPOST is 86.3%. This reflects that the higher order Markov models do not work well under the current experimental setup. Though the trigram HMM performs better than bigram HMM in literature but, a lower accuracy has been achieved by trigram HMM model for both the models. This can be explained by the fact that higher order Markov models perform well when a large number of

annotated data is available (to find a significant number of instances for state transition probabilities). Only 40, 000 annotated data has been used to conduct the experiment, which may be one reason for the relatively lower accuracy of trigram HMM. The scenario may change if a large amount of corpora can be used to train both bigram and trigram HMM.

4.4.2. Assessment of Error Types

Due to part-of-speech ambiguity, errors are produced by HMM model. Ambiguity mainly affects the assignment of correct part-of-speech to every word in a sentence. For example, the word ‘মহারাজা(*lmaharaja*)’ can be either a *common noun* or an *adjective*; the word ‘কের(*lkare*)’ can be either a *finite-verb* or a *non-finite verb*, even it can be a *post-position* also. It has been observed from the corpora that the word ‘মহারাজা(*lmaharaja*)’ is more likely to be a noun compare to an adjective. Similarly, the word ‘কের(*lkare*)’ is more likely to be a verb compares to post-position. The above observation probably fails to classify all occurrences of ‘মহারাজা(*lmaharaja*)’ as an adjective and ‘কের(*lkare*)’ as post-position. Table 8 shows the top 5 confusion classes of the HMM-S1+suf+MA model. First column gives the actual class with their frequency of occurrence in the test data, second column gives the predicted class corresponds to the actual class, third column gives the percentage of total error and fourth column gives the percentage of the error of for the particular class.

Actual Class (frequency)	Predicted Class	% of total errors	% of class errors
NP(251)	NN	21.03	43.82
JJ(311)	NN	5.16	8.68
NN(1483)	JJ	4.78	1.68
DTA(100)	PP	2.87	15
NN(1483)	VN	2.29	0.81

Table 8: Five most common types of errors

The most common type of error is the confusion of adjective and common noun – a result of the fact that most of the adjectives can be used as common nouns in Bengali. Almost all the confusions are wrong assignment due to less number of instances in the training corpora, including errors due to long distance phenomena.

4.5. Conclusion

In this chapter we have described an approach for automatic stochastic tagging of natural language text. The models described here are very simple and efficient for automatic tagging even when the amount of available labeled text is small. The models have a much higher accuracy than the naive baseline model. However, the performance of the current system is not as good as that of the best POS-tagger available for English and other European languages. The best performance is achieved for the supervised bigram HMM learning model along with morphological restriction on the possible grammatical categories of a word and suffix information for handling unknown words. In fact, in all the models discussed above the use of MA enhances the performance of the POS tagger significantly. We conclude that the use of morphological features is especially helpful to develop a reasonable POS tagger when tagged resources are limited.

Although HMM performs reasonably well for part-of-speech disambiguation task, it uses only local features (*current word, previous one or two tags*) for POS tagging. Uses of only local features may not work well for a morphologically rich and relatively free order word language – Bengali. Further, we plan to use other data driven statistical approaches, which use unrestricted and rich features in the framework of a probabilistic model. Maximum Entropy model and Conditional Random Fields can make use of feature information. Hence we will like to explore their use for POS tagging of Bengali.

Chapter 5

Tagging with Maximum Entropy Model

In the previous chapter, we have presented different HMM based stochastic language models for POS tagging. Simple HMM models do not work well when small amounts of labeled data are used to estimate the model parameters. Incorporating a diverse set of overlapping features in a HMM-based tagger is difficult and complicates the smoothing typically used for such taggers. In contrast, a Maximum Entropy based methods can deal with diverse, overlapping features. Maximum Entropy is a very flexible method of statistical modeling which handles the sparse data problem. Under this model, a natural combination of several features can be easily incorporated, which can not be done naturally in HMM models.

In this chapter, we present our work on Maximum Entropy based stochastic algorithm for POS tagging in Bengali. We also present the uses of a morphological analyzer to improve the performance of a tagger in the maximum entropy framework. Finally, we present the uses of different features and their effective performance in the Maximum Entropy model.

The organization of the chapter is as follows: Section 1 describes some basic definitions and notation of the Maximum Entropy model. Section 2 is devoted to our particular approach to Bengali POS tagging using Maximum

Entropy model. Section 3 describes the different experiments conducted for this task. Section 4 presents the experimental results and assessment of error types and Section 5 provides the conclusion.

5.1. Maximum Entropy Model

Maximum Entropy (ME) is a very flexible method of statistical modeling. The ME model estimates the probabilities based on the imposed constraints. Such constraints are derived from the training data, maintaining some relationship between *history* and *outcomes*. *Outcomes* are defined as the set of allowable tags. ME model 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 . A *history* in ME is all of the conditioning data which enables to assign probabilities to the set of outcomes. 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:

$$P(t|h_i) = p(t | \text{information derivable from the test corpus at index } i)$$

The computation of $P(t|h)$ 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.

Given a set of features and the training data, the ME estimation process produces a model in which every feature f_i is associated with a parameter λ_i . This allows the computation of the conditional probability as follows:

$$P(t|h) = \frac{\prod_i \lambda_i f_i(h,t)}{Z_\lambda(h)}$$

$$Z_\lambda(h) = \sum_t \prod_i \lambda_i f_i(h,t)$$

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.

We have used the Java-based OpenNLP maximum entropy package³ for the computation of the value of the parameters of λ_i . This allows us to concentrate on selecting the features which best characterize the problem instead of worrying about assigning the relative weights to the features.

5.1.1. Building a Model with ME

A simple method of building a ME model is by using the Generalized Iterative Scaling (GIS) algorithm, which is guaranteed to converge to a solution (Darroch and Ratcliff, 1972). An outline of the algorithm as applied to a conditional model is given below.

Generalized iterative Scaling:

Given a family of index functions f_i and the associated estimation for the value of the functions K_i , each iteration j creates a new estimation of the model parameters λ_i which matches the constraints better than the previous. Each- iteration consists of the following steps:

1. Compute the expectation of all the f_i under the current estimate of the probability function.

$$K_i^{(j)} = \sum_h \tilde{P}(h) \sum_t P_j(t | h) f_i(h, t)$$

2. Compute the actual value of $K_i^{(j)}$ and update the λ_i according to the following formula:

$$\lambda_i^{(j+1)} = \lambda_i^{(j)} \cdot \frac{K_i}{K_i^{(j)}}$$

3. Define the next estimate of the probability function based on the new λ_i

³ <http://maxent.sourceforge.net>

$$P_{j+1}(t|h) = \frac{\prod_i \lambda_i^{(j+1) f_i(h,t)}}{Z_{\alpha} h^{(j+1)}}$$

Continue iterating until convergence or near-convergence.

5.2. Our Particular Approach with ME Model

Construction of a Maximum Entropy modelling system is a process of trial and error. The process mainly involves identifying a set of features which reduces the system error (i.e. the identification of features which has reasonably good contribution in the classification task).

As described in chapter 4 (section 4.2), we also distinguish three main components in our ME based model for the Bengali POS tagging task. The three components of the ME based tagger are depicted in Figure 11.

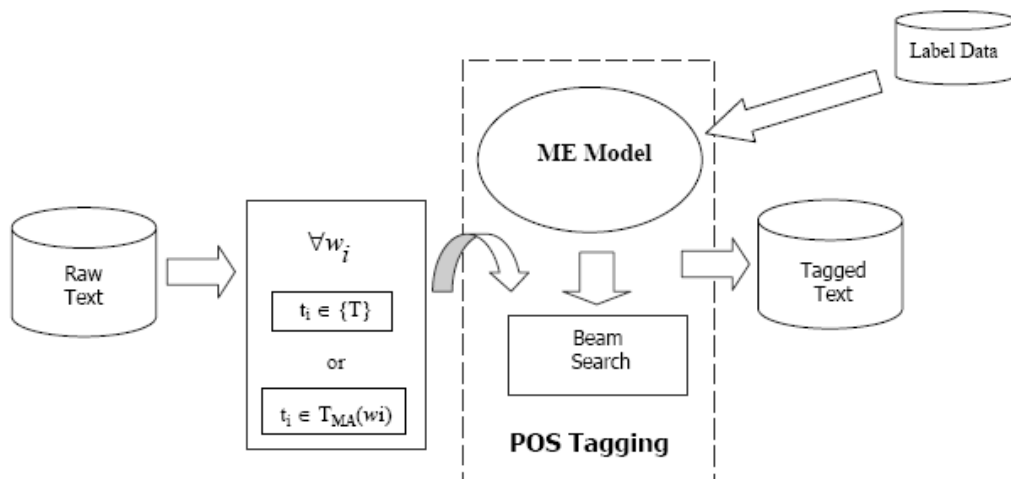


Figure 11: The ME based POS tagging architecture

In particular to ME model, the *language model* component is represented by the model parameters. Similar to HMM there is a *disambiguation algorithm*, which decides the most probable tag sequence for a given word sequence. We use *beam search* algorithm for the disambiguation. The third component

(*possible class restriction module*) is being used in the ME model as it was in the HMM based POS tagging model. In our ME based model, we also assume that every word can be associated with all the tags in the tagset (i.e. *a set of 40 tags in the tagset* $\{T\}$). Further, we assume the POS tag of a word w can take the values from the set $T_{MA}(w)$, where $T_{MA}(w)$ is computed by the Morphological Analyzer. The following subsections give a detailed description of the different components used and our experimental results with the ME model.

5.2.1. Features

The features are binary valued functions which associate a tag with various elements of the context; for example:

$$f_i(h,t) = \begin{cases} 1 & \text{if current_token}(h)=Ami \text{ and } t=PRP \\ 0 & \text{otherwise} \end{cases}$$

Feature selection plays a crucial role in the ME framework. Experiments were carried out identify the most suitable features for the POS tagging task. The main features for the POS tagging task have been identified based on the different possible combinations of available word and tag context. The features also include prefix and suffix for all words. The term prefix/suffix is a sequence of first/last few characters of a word, which may not necessarily be a linguistically meaningful prefix/suffix. The use of prefix and suffix information as features is found to be effective for highly inflected languages. We considered different combinations from the following set of features for identifying the best feature set for the POS tagging task:

$$F = \{w_i, w_{i-1}, w_{i-2}, w_{i+1}, w_{i+2}, t_{i-1}, t_{i-2}, |pre| \leq 4, |suf| \leq 4\}$$

A pictorial representation of the potential features is depicted in the Figure 12. The single solid line represents the whole feature set ‘F’ which consists of both static and dynamic features. The dotted line represents the static features which are predetermined from the input sentences. The dashed line denotes the

set of dynamic features which are estimated in run time. The doubled solid line represents the predicted outcome.

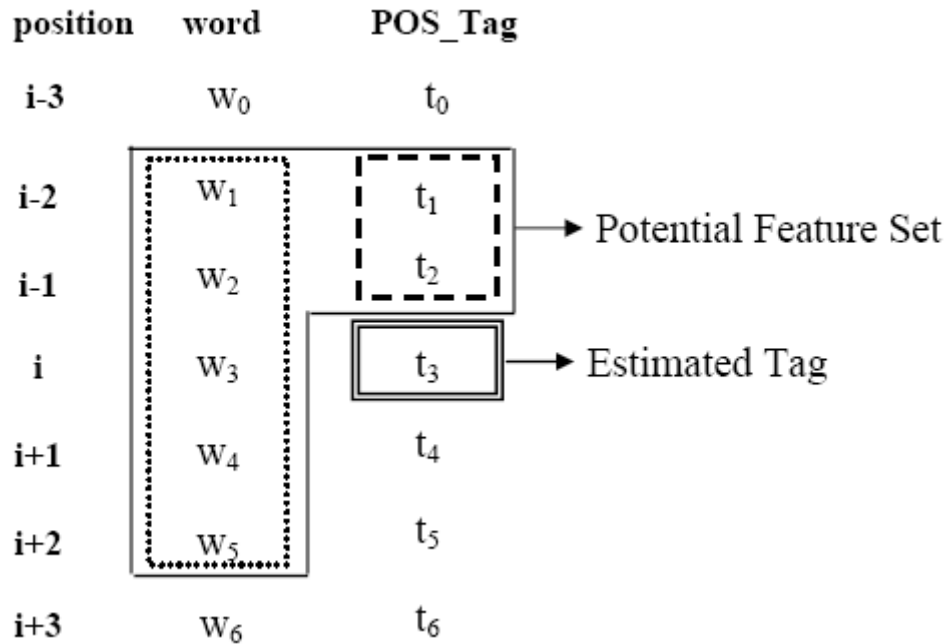


Figure 12: The Potential Feature Set (F) for the ME model

5.2.2. Training the System

As mentioned above, we built many different POS taggers with the ME tool. These models were differentiated from each other by the features which were included in the model. These models use a corpus hand-marked with the correct POS labels.

The system uses Generalized Iterative Scaling (GIS) to build the ME model, which is guaranteed to converge to a solution in this kind of problem (Darroch, 1972). The procedure of training the system is summarized below.

1. Define the training corpus, C

2. Tokenize the training corpus
3. Create a file of candidate features, including lexical features derived from the training corpus
4. Create an *event file* listing every feature which activates every pair $\langle h, t \rangle$ for $h \in C$ and $t \in \{T\}$
5. Compute the ME weightings λ_i for every f_i using the ME toolkit with the event file as input

5.2.3. Decoding

The problem of POS tagging can be formally stated as follows. Given a sequence of words $w_1 \dots w_n$, we want to find the corresponding sequence of tags $t_1 \dots t_n$, drawn from a set of tags T, which satisfies:

$$P(t_1 \dots t_n | w_1 \dots w_n) = \prod_{i=1 \dots n} P(t_i | h_i) \quad \text{Eq. 6}$$

Where, h_i is the context for word w_i . The Beam Search algorithm is used find the most probable sequence given the sentence.

The POS tagger has been implemented based on the ME model (described in sub-section 5.2.1) to find the most probable tag sequence for a given sequence of words. The models use a set of 40 tags (T) for every word in a sentence and the most probable tag sequence is determined using the Beam Search algorithm using equation 6.

Let $W = \{w_1 \dots w_n\}$ be an untagged sentence, and let s_{ij} be the j th highest probability tag sequence up to word w_i . The following is the procedure for the beam search:

1. Generate the probability of each tag from the set $\{T\}$ for w_1 , find top N (size of the beam), set $s_{1j}, 1 \leq j \leq N$, accordingly.
2. Initialize $i = 2$.
 - (a) Initialize $j = 1$
 - (b) Generate the probability of each tag from the set $\{T\}$ for w_i , given $s_{(i-1)j}$ as previous tag context, and append each tag to $s_{(i-1)j}$ to make a new sequence
 - (c) $j = j + 1$, repeat from (b) if $j \leq N$
3. Find N highest probability sequences generated by the above loop, and set $s_{ij}, 1 \leq j \leq N$, accordingly.
4. $i = i + 1$, repeat from (a) if $j \leq N$
5. Return highest probability sequence s_{n1}

Figure 13: The Beam search algorithm used in the ME based POS tagging model

As we have done with the HMM model, we integrate morphological information with the ME model in order to further improve the tagging accuracy. We assume that the POS-tag of a word w can take values from the set $T_{MA}(w)$, where $T_{MA}(w)$ is computed by the Morphological Analyzer. Note that the size of $T_{MA}(w)$ is much smaller than T . Thus, we have a restricted choice of tags as well as tag sequences for a given sentence. Since the correct tag t for w is always in $T_{MA}(w)$ (assuming that the morphological analyzer is complete), it is always

possible to find out the correct tag sequence for a sentence even after applying the morphological restriction. Due to a much reduced set of possibilities, this model performs better even when only a small labeled training corpus is available. We shall call these new models **ME + MA_R**.

It is also possible that all words are not included to the Morphological Analyzer, then the set $T_{MA}(w)$ return NULL. In this case, we assume the POS tags of a word w can take values from the set of $T_O(w)$, where $T_O(w)$ denotes the set of open class grammatical categories (all classes of Noun, Verb, Adjective, Adverb and Interjection).

Figure 13 represents the decoding procedure of the ME based POS tagging model. The figure at the top, represents the search space of the beam search algorithm. We use a beam of size 3 for our experiments and at every level we explore only the top 3 probable tags. Following this procedure, we find the most probable tag sequence using the Beam search algorithm as described in Figure 14.

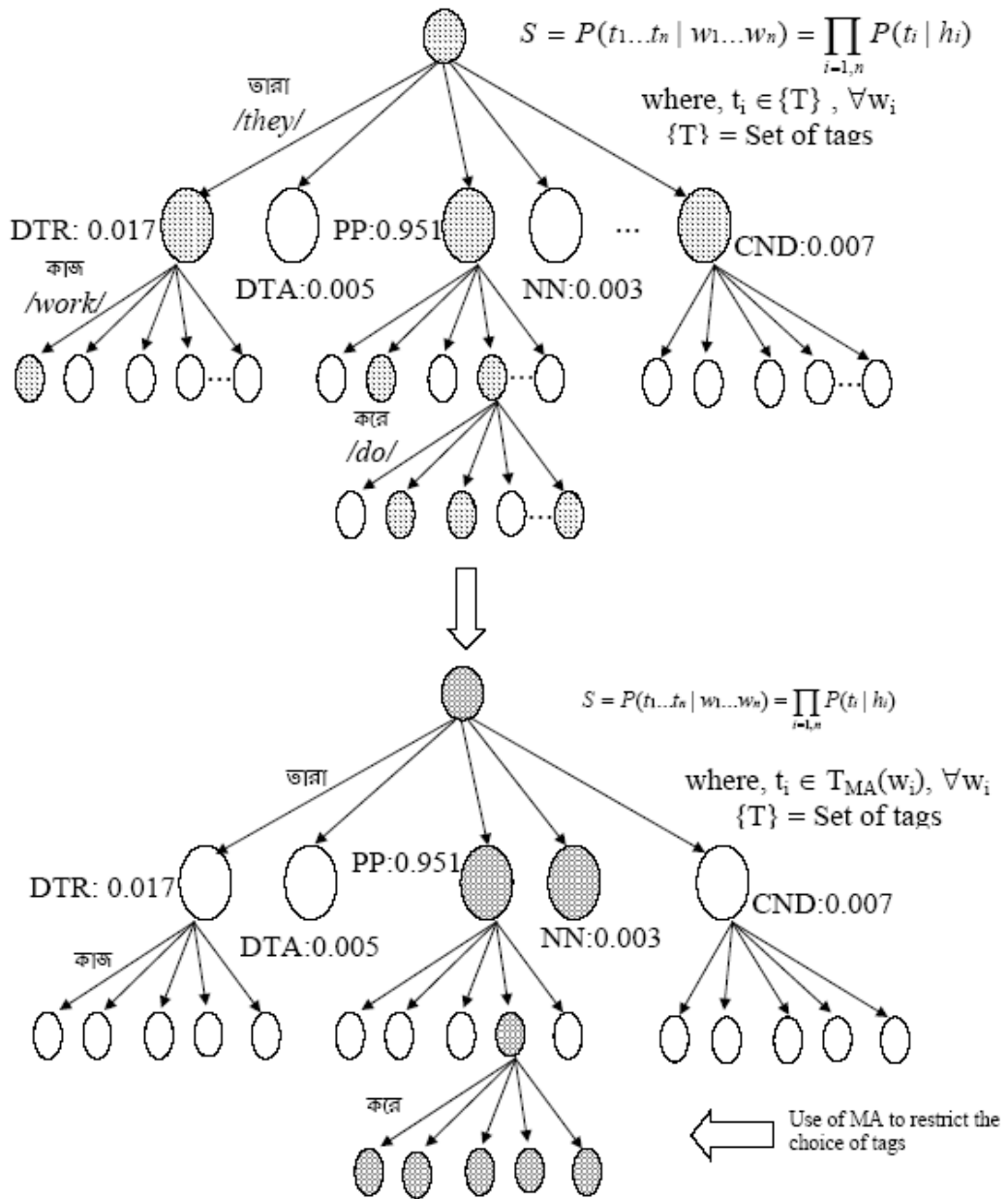


Figure 14: Decoding the most probable tag sequence in ME based POS tagging model

As discussed earlier, we use the Morphological Analyzer to restrict the choice of tags for each word in a sentence. Unlike the simple ME model, we generate the probability of each tags from the set $T_{MA}(w)$ for every word w , as shown in step 1 and 2(b) in Figure 15. The bottom portion of the figure represents the search space of the decoding procedure. For example, the word $\text{তারা}\{tArA(/they/)\}$ has only two possible tags NN (common noun) and PP (personal pronoun) with probability values on 0.003 and 0.951 respectively. So, we explore only these two nodes for the next input sequence. But, earlier we had a fixed beam of size 3, and the three high probable tags from the set of 40 tags are PP:0.951, DTR:0.017 and CND:0.007. Thus, we explore two nodes (DTR and CND) for the word $tArA$ which are not grammatically correct tags for the word. Figure 15 represents the modified search procedure during the disambiguation of the ME based POS tagging model.

Let $W = \{w_1 \dots w_n\}$ is an untagged sentence, and let s_{ij} be the j th highest probability tag sequence up to word w_i . The following is the procedure for beam search:

1. Generate the probability of each tag from the set $T_{MA}(w_1)$ for w_1 , find top N (size of the beam), set $s_{1j}, 1 \leq j \leq N$, accordingly.
2. Initialize $i = 2$.
 - (a) Initialize $j = 1$
 - (b) Generate the probability of each tag from the set $T_{MA}(w_i)$ for w_i , given $s_{(i-1)j}$ as previous tag context, and append each tag to $s_{(i-1)j}$ to make a new sequence
 - (c) $j = j + 1$, repeat from (b) if $j \leq N$
3. Find N highest probability sequences generated by above loop, and set $s_{ij}, 1 \leq j \leq N$, accordingly.
4. $i = i + 1$, repeat from (a) if $j \leq N$
5. Return highest probability sequence s_{n1}

Figure 15: Search procedure using MA in the ME based POS tagging model

While MA helps us to restrict the possible choice of tags for a given word, one can also use suffix information (i.e., the sequence of last few characters of a word) to further improve the models. For ME models, suffix information has been used as features. We shall denote the models with suffix information with a ‘+suf’ marker. Thus, we have four different models– **ME**, **ME+suf**, **ME+MA_R** and **ME+suf+MA_R**.

In a Maximum Entropy based POS tagging model, the MA can be used in two different ways. First, the MA can be used to restrict the possible choice of tags for each lexical item during disambiguation as described above. However, the MA can also be used as features during creation and disambiguation of the ME based POS tagging system. We use $T_{MA}(w_i)$ (i.e. the possible choice of tags for the word w_i) for each word as a feature vector for the ME model. We shall denote the model with MA as features with a ‘+MA_F’ marker. Thus, we have two more models – **ME+MA_F** and **ME+suf.+MA_F**

5.3. Experiments

We have a total of six (ME, ME+suf, ME+MA_R, ME+MA_F, ME+suf+MA_R, ME+suf +MA_F) models as described in subsection 5.2.3 under the ME based stochastic tagging schemes. The same training text has been used to estimate the parameters for all the models. The experiments were conducted with three different sizes (10K, 20K and 40K words) of the training data to understand the relative performance of the models as we keep on increasing the size of the annotated data.

Forty different experiments were conducted taking several combinations from set ‘F’ to identify the best suited feature set for the POS the tagging task with the ME and ME+suf model. The detailed results on the accuracies of the experiments are given in the Appendix B. From our empirical analysis we find that the combination of a contextual features (current word and previous tag), prefixes and suffixes gives the best performance for the ME model. We conducted different experiments varying the prefix/suffix length from 1 to 6 characters. It has been observed empirically that the prefix and suffix length of 4 gives better results for all the ME based models. Based on our observations, the inclusion of suffix essentially captures helps to understand the morphological inflection of the surface form word and in Bangla most morphological inflections lies in the last 4 characters of the words. On the other hand, a prefix of length

four might capture the linguistic prefix (if attached) or the root of the word. It is interesting to note that the inclusion of prefix and suffix for all words gives better result instead of using only for rare words. This can be explained by the fact that due to small amount of annotated data, a significant number of instances are not found for most of the words in the language vocabulary. Table 9 lists the final feature set used in our simple Maximum Entropy based POS tagging model.

Condition	Features
Static features for all words	Current word(w_i) $ \text{prefix} \leq 4$ $ \text{suffix} \leq 4$
Dynamic Features for all words	POS tag of previous word (t_{i-1})

Table 9: Feature used in the simple ME based POS tagging

Further, we use MA (either during disambiguation (+MA_R) or as a feature (+MA_F)) with best features identified from the 40 different experiments.

5.3.1. Data Used for the Experiments

The ME models for the Bengali POS tagging has been trained with the same data as used in the HMM model in the previous chapter (subsections 4.3.1 and 4.3.2). The ME model is trained only using the annotated text corpus (approximately 40,000 words). All the models have been tested on a set of randomly drawn 400 sentences (5000 words) distinct from the training corpus as used for the testing in HMM based POS tagging in the previous chapter.

5.4. System Performance

Like HMM based POS tagging system, the tagging accuracy of the ME based POS tagging models have been evaluated as the ratio of the correctly tagged words to the total number of words.

$$Accuracy(\%) = \frac{\text{Correctly tagged words by the system}}{\text{Total no. of words in the evaluation set}} \times 100$$

Figure 16 shows the improvement of the overall accuracy along with the increment of the annotated training data using the features described in table 9. It is interesting to note that the rate of improvement of the overall accuracy using simple ME model is much higher compare to the other three models (ME+suf, ME+MA_R and ME+suf+MA_R), as we keep on increasing the amount of annotated training data. From the above observation, it is significant to note that the use of a morphological analyzer is helpful when less amount of annotated data is available for the POS disambiguation task.

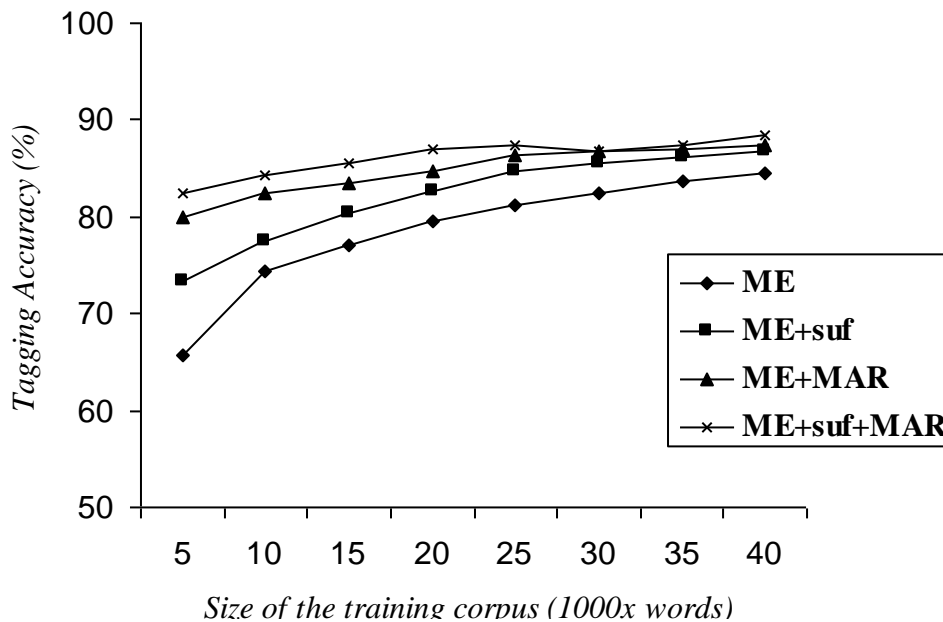


Figure 16: The overall accuracy growth of different ME based tagging model

We also measure the known word and unknown word accuracy separately for all the models to understand their performance in a poor resource scenario. It is obvious that the number of unknown words is always high when less amount of annotated data is available. So, a model which better handles the unknown words are considered to be a well fitted model for the POS disambiguation task in a poor resource scenario.

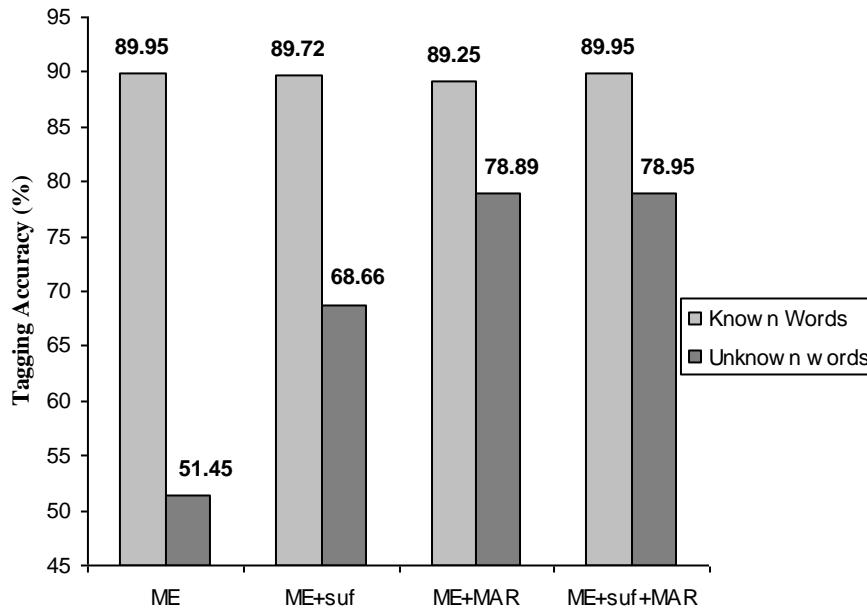


Figure 17: The known and unknown word accuracy under different ME based model

It is interesting to note that the known word accuracy under the above three model are almost same when a reasonable amount of annotated data is available. But, it is clear from the figure 17 that the unknown word error rate is much lower when a morphological analyzer is used to restrict the probable set of tags for a given word. Nevertheless, the unknown word accuracy gives an improvement of 17%, 24% and 27% in case of ME+suf, ME+MA_R and ME+suf+MA_R models respectively over the simple ME model.

Table 10 summarizes the final accuracies achieved by different ME based POS tagging models with the varying size of the training data (10K, 20K and

40K). Note that the baseline model (i.e., the tag probabilities depends only on the current word) has an accuracy of 76.8%.

Method	Accuracy		
	10K	20K	40K
ME	74.37 (87.98,51.38)	79.50 (89.0,52.36)	84.56 (89.95, 51.46)
ME+suf	77.38	82.63	86.78 (89.72, 68.66)
ME+ MA _R	82.51	84.97	87.38 (89.25, 78.89)
ME+suf+MA _R	84.13	87.07	88.41 (89.95, 78.95)

Table 10: Tagging accuracies (%) of different models with 10K, 20K and 40K training data. The accuracies are represented in the form of Overall Accuracy (Known Word Accuracy, Unknown Word Accuracy)

In order to estimate the effect of using MA as feature in the ME based POS tagging model along with the features listed in table 9, two experiments has been conducted - ME+MA_F and ME+suf.+MA_F. The results of the experiments are shown in Table 11 using the 40K annotated training data.

Method	Accuracy (%)
ME + MA _F	86.86
ME+suf+MA _F	88.08

Table 11: Tagging Accuracy with morphology as a feature in ME based POS tagging model

5.4.1. Observations

The above experiments lead us to the following observations.

The use of suffix information plays an important role, especially when the amount of training data is less. It is interesting to note that the **ME+suf** model gives an improvement of around 3%, 3% and 2% over the simple **ME** model for 10K, 20K and 40K training data respectively. The trends were observed in the case of the supervised and semi-supervised HMM models in the previous chapter.

Another significant observation is that the use of morphological restriction (**ME+MA_R**) gives an improvement of 8%, 5% and 3% respectively over the **ME** in case of 10K, 20K and 40K training data. This essentially signifies that the use of morphological restriction works well in the case of small training data. As the improvement due to MA decreases with increasing data, it might be concluded that the use of morphological restriction may not improve the accuracy when a large amount of training data is available.

The above two observations motivated us to use both suffix and MA together for all the models. From our empirical observations we found that both suffix and morphological restriction gives an improvement of 10%, 8% and 4% over the ME model respectively for the three different sizes of training data.

In order to compare the ME models with the Hidden Markov Models, it has been observed that the ME models perform significantly better when the size of the training data is less and suffix information is not considered. However, the ME models achieve comparable accuracy with HMM models when suffix information and/or morphological restriction is used.

Furthermore, in order to estimate the relative performance of the models, experiments were carried out using MA (**ME+MA_F** and **ME+suf+MA_F**) as

feature in the ME model. The respective accuracies achieved by the above models are 86.68% and 88.08% for 40K word training data. The accuracy of the model is quite comparable with the accuracy achieved by the ME model when morphology is used as restriction on the choice of the possible POS tags.

5.4.2. Assessment of Error Types

Table 12 shows the top five confusion classes for ME+suf+MA_R model. Here also we observe a similar trend as in the HMM models. The most common types of errors are the confusion between proper noun and common noun and the confusion between adjective and common noun. This results from the fact that most of the proper nouns can be used as common nouns and most of the adjectives can be used as common nouns in Bengali.

Actual Class (frequency)	Predicted Class	% of total errors	% of class errors
NP(251)	NN	26.34	56.57
JJ(311)	NN	8.16	14.14
VF(445)	VN	4.61	5.62
NN(1483)	JJ	4.61	1.68
DTA(100)	PP	2.42	14.0

Table 12: Five most common types of errors with the ME model

Almost all the confusions are wrong assignment due to less number of instances in the training corpora, including errors due to long distance phenomena.

5.5. Conclusion

In this chapter we have described a Maximum Entropy based approach for automatic POS tagging of Bengali text. The models described here are very simple and are effective for automatic tagging even when the amount of available labeled text is small. The best performance is achieved for the ME model along

with suffix information and morphological restriction on the possible grammatical categories of a word.

Although simple ME based tagger performs reasonably better compare to the simple HMM (HMM-S1), we think none of the tagger is better than other in absolute terms when morphological restriction is applied on the set of tags. Due to the probabilistic formulation, HMM obtains the most likely sequence of tags using the linear sequence of observations, that is, HMM tries to maximize the most likely tag sequence globally for a given sequence of words. Instead, the ME based models locally maximize the conditional probability of a word being into a particular grammatical class.

It has been reported that ME based models performs slightly more accurate compare to markov models (Kazama, 2001; Zhao, 2004; McCallum, 2000). But in our experiment, ME based models achieve roughly the same accuracy as HMM on the Bengali corpus. The power of the ME model lies in its diverse and overlapping set of features. In our ME experiment we are using only a small number of features (*current word, previous tag, prefix/suffix of length four*). We also conducted experiments with large number of features but, the inclusion of more features worsens us the accuracy. A larger number of features can perhaps help when a larger amount of annotated training data is available so that there are significant amounts of evidence for every feature instance. This might be one of the reasons for relatively lower accuracy for inclusion of rich feature set.

The above observations motivated us to use other data driven statistical approaches, which use unrestricted and rich features in the framework of a probabilistic model. Conditional Random Fields are extremely flexible techniques for the above linguistic modeling, which uses arbitrary chain sequence of the Markov process and it can also incorporate large number of features.

Chapter 6

Tagging with Conditional Random Fields

In the previous chapter, we have described different Maximum Entropy based POS tagging models for Bengali. It has been observed that the Maximum Entropy model does better than the HMM model for small training data. But with higher amount of training data the performance of the HMM and ME models are comparable. Maximum Entropy based models are a form of discriminative model, which maximize the conditional probability distribution of the training example. Maximum Entropy model uses per-state exponential model for estimating the conditional probability of the next state given the current state. In contrast, Conditional Random Field (CRF) has a single exponential model for the joint probability of the entire sequence of states given the observation sequence. Like Maximum Entropy model, a CRF based method can also deal with diverse and overlapping features. A CRF is a very flexible method which deals with the sparse data problem well. Under this model, a natural combination of diverse set of features can be easily incorporated, which cannot be done naturally in HMM.

In this chapter, we present our work on CRF based POS tagging in Bengali. We present the uses of different features and their effective performance in the CRF based model. We have used the same features as we used in the Maximum Entropy framework to understand the relative performance of the two

different conditional probability models. Finally, we also present the uses of morphological analyzer to improve the performance of a tagger in the Conditional Random field framework.

The organization of the chapter is as follows: Section 1 provides some basic definition and notation of the CRF model. Section 2 describes our particular experimental setup to Bengali POS tagging using Conditional Random Fields. Section 3 describes the different experiments conducted for the task. Section 4 presents the experimental results and Section 5 provides the conclusion.

6.1. Conditional Random Fields

One of the most common methods for performing POS sequence labeling task is that of employing Hidden Markov Models (HMMs) to identify the most likely POS tag sequence for the words in a given sentence. HMMs are generative models, which maximize the joint probability distribution $p(X, Y)$ where X and Y are random variable respectively representing the observation sequence (i.e. the word sequence in a sentence) and the corresponding label sequence (i.e. the POS tag sequence for the word of a sentence). Due to the joint probability distribution of the generative models, the observation at any given instant of time, may only directly depend on the state or label at that time. This assumption may work for a simple data set. However for the problem of the POS labeling task, the observation sequence may depend on multiple interacting features and long distance dependencies.

One way to satisfy the above criteria is to use a model that defines conditional probability $p(Y/x)$ over label sequences given a particular observation sequence x , rather than a joint probability distribution over both label and observation sequence. Conditional models are used to label an unknown observation sequence, by selecting the label sequence that maximizes the conditional probability.

Conditional Random Fields (CRFs) (Lafferty et. al., 2001) are a probabilistic framework for labeling sequential data based on the conditional approach described above. A CRF is an undirected graphical model that defines a single exponential model over label sequence given the particular observation sequence. The primary advantage of the CRF over the HMMs is the conditional nature, resulting in the relaxation of the independence assumption required by HMMs. CRF also avoid the label bias problem (Lafferty et. al., 2001) of the Maximum Entropy model and on other directed graphical models. Thus CRFs outperforms HMM and ME models on a number of sequence labeling tasks (Lafferty et. al., 2001; Pinto and McCallum, 2003; Sha and Pereira, 2003).

6.1.1. Undirected Graphical Models

A CRF can be viewed as an undirected graphical model or Markov random field, globally conditioned on X , the random variable representing the observation sequence. Formally, $G = (V, E)$ is an undirected graph such that there is a node $v \in V$ corresponding to each of the random variables representing an element Y_v of Y . If each random variable Y_v obeys the Markov property with respect to G , then (Y, X) is a conditional random field. However, when we model the POS sequence labeling problem, the simplest and the most common graph structure encountered is that in which the nodes corresponding to elements of Y (i.e. the POS tag labels) form a simple first order chain as illustrated in figure 19.

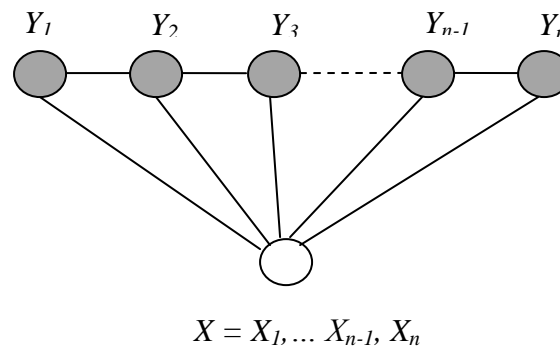


Figure 18: Graphical structure of a chain-structured CRF for sequences.

6.1.2. Statistical Background

Lafferty et al. define 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

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

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 μ_k are the model parameters to be estimated from the training data.

Like Maximum Entropy model, when defining feature functions in CRF model, we construct a set of real valued features $b(x, i)$ of the observation to express some characteristics of the training data.

Each feature function takes on the value of one of the real valued observation features $b(x, i)$ if the current state or current and previous states take on some particular values. Thus all feature functions are real valued in nature. For example, consider the following feature functions:

$$t_j(y_{i-1}, y_i, x, i) = \begin{cases} b(x, i) & \text{if } y_{i-1} = NN \text{ and } y_i = PP \\ 0 & \text{otherwise} \end{cases}$$

This allows the probability of a label sequence y given an observation sequence x to be written as

$$F_j(y, x) = \sum_{j=1}^n f_j(y_{i-1}, y_i, x, i) \quad \text{Eq. 7}$$

Where

$$F_j(y, x) = \sum_{j=1}^n f_j(y_{i-1}, y_i, x, i)$$

and $f_j(y_{i-1}, y_i, x, i)$ is either a state function $s_k(y_i, x, i)$ or a transition function $t_j(y_{i-1}, y_i, x, i)$. $Z(x)$ is the normalizing factor.

6.1.3. Parameter Estimation

Assuming the training data $\{(x^{(k)}, y^{(k)})\}$ are independently and identically distributed, the product of equation 7 over all training sequence, as a function of the parameter λ , is known as the likelihood, denoted by $p(\{y^{(k)}\}|\{x^{(k)}\}, \lambda)$. Maximum likelihood training chooses parameter values such that the logarithm of the likelihood, known as the log-likelihood, is maximized. For a CRF, the log-likelihood is given by

$$L(\lambda) = \sum_k \left[\log \frac{1}{Z(x^{(k)})} + \sum_j \lambda_j F_j(y^{(k)}, x^{(k)}) \right]$$

This function is concave, guaranteeing convergence to the global maxima.

Differentiating the log-likelihood with respect to the parameter λ_j gives

$$\frac{\partial L(\lambda)}{\partial \lambda_j} = E_{\tilde{p}(Y, X)} [F_j(Y, X)] - \sum_k E_{p(Y|x^{(k)}, \lambda)} [F_j(Y, x^{(k)})]$$

where $\tilde{p}(Y, X)$ is the empirical distribution of training data and $E_p[\cdot]$ denotes the expectation with respect to distribution p . The expectation of each feature with respect to the model distribution is equal to the expected value under the empirical distribution of the training data. It is not possible to analytically estimate the parameter values that maximize the log-likelihood – setting the gradient to zero and solving for λ does not always give a closed form solution. Instead, maximum likelihood parameters must be estimated using an iterative technique such as iterative scaling (Darroch and Ratcliff, 1972; Berger, 1997; Pietra et. al., 1995) or gradient based method (Sha and Pereira, 2003; Wallach, 2002). We use an adaptation of Java based open source CRF package⁴.

⁴ <http://crfpp.sourceforge.net/>

6.2. Experimental Setup

We started by setting up a system with the features used in the Maximum Entropy framework in the previous chapter. In the second step, we used the morphological analyzer during creation and disambiguation of the system.

6.2.1. Features

The features are binary valued functions which associate a tag with various elements of the context; as described in the previous section.

Feature selection plays a crucial role in the CRF framework. Experiments were carried out to find out the most suitable features for the POS tagging task in Maximum Entropy framework as described in the previous chapter. The main features for the POS tagging task have been identified based on the different possible combination of available word and tag context. The features also include prefix and suffix for all words. The term prefix/suffix is a sequence of first/last few characters of a word, which does not mean a linguistically meaningful prefix/suffix. The use of prefix and suffix information works well for highly inflected languages. We considered different combination from the following set for inspecting the best feature set for POS tagging task:

$$F = \{w_i, w_{i-1}, w_{i-2}, w_{i+1}, w_{i+2}, t_{i-1}, t_{i-2}, |pre| \leq 4, |suf| \leq 4\}$$

From the empirical observation we found in the Maximum Entropy based POS tagging model a very simple feature of current word, previous tag and prefix/suffix gives the best result in the current experimental setup. Further the use of MA improves the accuracy of the system. In CRF based POS tagging for Bengali, we use the features that were found to be best suited for the Maximum Entropy model in the previous chapter.

6.2.2. Experiments

Three taggers have been implemented based on the CRF model. The first tagger (we shall call it **CRF**) makes use of the simple contextual features, whereas the second tagger (we shall call it **CRF+suf**) uses prefix suffix features along with the simple contextual features. In order to further improve the tagging accuracy, we integrate morphological information with the model. We use $T_{MA}(w_i)$ (i.e. the possible choice of tags for the word w_i) for each word as a feature vector for the CRF model. We shall denote the model with MA as features with a ‘+MA_F’ marker. Thus, we have models – **CRF+suf.+MA_F**

We have total of three (CRF, CRF+suf, CRF+suf +MA_F) models under the CRF based stochastic tagging scheme. The same training corpus has been used to estimate the parameters for all the models. The experiments were conducted with three different sizes (10K, 20K and 40K words) of the training data to understand the relative performance of the models as we keep on increasing the size of the annotated data.

6.2.3. Data Used for the Experiments

The CRF models for the Bengali POS tagging has been trained with the same data as used in the HMM model in the chapter 4 (subsections 4.3.1 and 4.3.2). The CRF model is trained only using the annotated text corpus (approximately 40,000 words). All the models have been tested on a set of randomly drawn 400 sentences (5000 words) separated from the training corpus as used for the testing in HMM based POS tagging.

6.3. System Performance

Like HMM and ME based POS tagging system, the tagging accuracy of the CRF based POS tagging models have been evaluated as the ratio of the correctly tagged words to the total number of words.

$$Accuracy(\%) = \frac{\text{Correctly tagged words by the system}}{\text{Total no. of words in the evaluation set}} \times 100$$

Figure 20 shows the improvement of the overall accuracy along with the increment of the annotated training data using the features described in section 6.2.2. It is interesting to note that the rate of improvement of the overall accuracy using simple CRF model is much higher compare to the other two models (CRF+suf, and CRF+suf+MA_F), as we keep on increasing the amount of annotated training data. From the above observation, it is significant that the uses of a morphological analyzer work well when fewer amounts of annotated data is available for the POS disambiguation task.

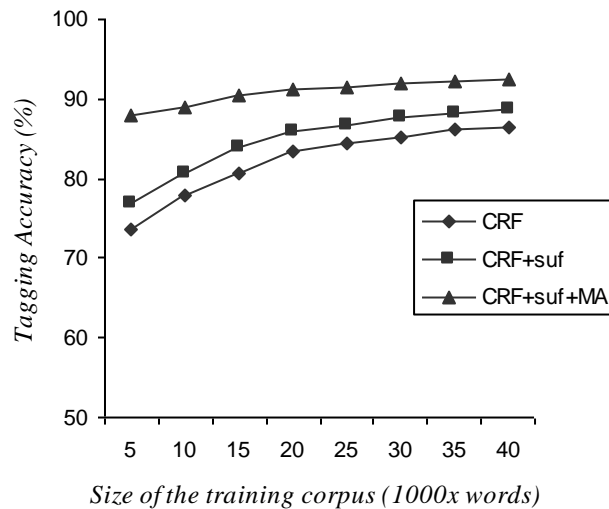


Figure 19: The overall accuracy growth of different CRF based POS tagging model

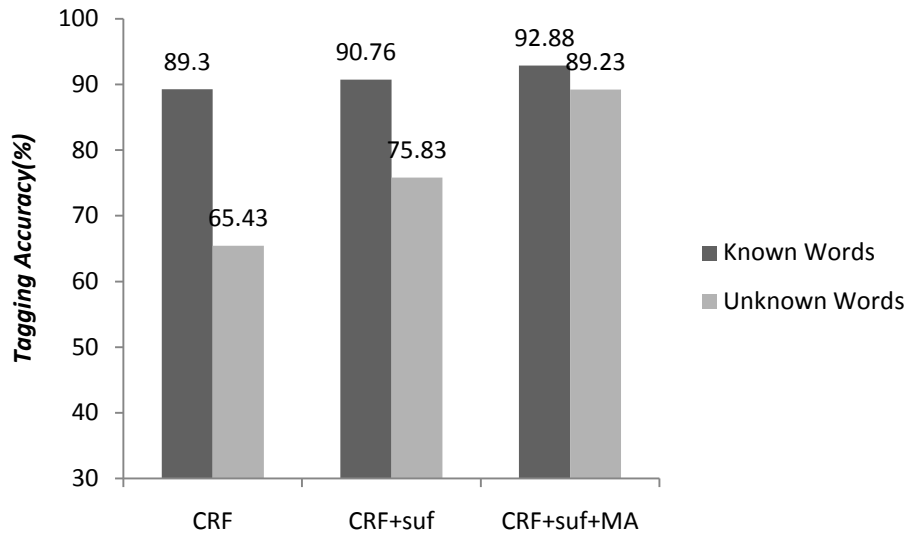


Figure 20: Known and unknown word accuracies with the CRF based models

Figure 20 shows known and unknown word accuracies with different CRF based model. Although a little improvement has been observed for known words but a huge improvement can be found for unknown words with use of suffix and/or MA. The uses of suffix (CRF+suf) give an improvement of 10% over the simple CRF model while the use of suffix and MA together (CRF+suf+MA) improves 24% over the simple CRF model. Due to higher percentage of unknown words in the test set, the unknown word accuracy has a major role to obtain reasonable overall accuracy of the taggers.

Table 13 summarizes the final accuracies achieved by different CRF based POS tagging models with the varying size of the training data (10K, 20K and 40K).

Method	Accuracy		
	10K	20K	40K
CRF	77.9	83.43	86.53
CRF+suf	80.57	85.96	88.61
CRF+suf+MA _F	87.87	90.57	92.37

Table 13: Tagging accuracies (%) of different models with 10K, 20K and 40K training data. The accuracies are represented in the form of Overall Accuracy.

6.3.1. Observations

The experiments with CRF model for Bengali POS tagging performs similar to the ME based POS tagging models. The above experiments lead us to the following observations.

The use of suffix information plays an important role, especially when the amount of training data is less. It is interesting to note that the **CRF+suf** model gives an improvement of around 3%, 2.5% and 2% over the **CRF** model for 10K, 20K and 40K training data respectively. The trends were observed in the case of the supervised and semi-supervised HMM models and ME based POS tagging models in the previous chapters.

Furthermore, the use of morphology gives a significant improvement over the simple CRF and CRF+suf models. The use of morphological restriction (**CRF+suf+MA_F**) gives an improvement of 10%, 7% and 6% respectively over the **CRF** in case of 10K, 20K and 40K training data. This essentially signifies that the use of morphological restriction works well in the case of small training data. As the improvement due to MA decreases with increasing data, it might be concluded that the use of morphological restriction may not improve the accuracy when a large amount of training data is available.

In order to compare the CRF models with the Markov Models and ME based models, it has been observed that the CRF models perform significantly better when the size of the training data is less and suffix information is not considered.

6.4. Conclusion

In this chapter we have described a Conditional Random Field based approach for automatic POS tagging of Bengali text. The models described here are simple and quite effective for automatic tagging even when the amount of available labeled text is small. The best performance is achieved for the CRF model when

using suffix information and morphologically possible grammatical categories as features of a word.

CRF based taggers perform reasonably better compared to the HMM and ME based tagger. Due to the probabilistic formulation, HMM obtains the most likely sequence of tags using the linear sequence of observations, that is, HMM tries to maximize the most likely tag sequence globally for a given sequence of words. On the other hand, the ME based models locally maximize the conditional probability of a word being into a particular grammatical class. In contrast, Conditional Random Field (CRF) has a single exponential model for the joint probability of the entire sequence of states given the observation sequence.

The power of the CRF model lies in its diverse and overlapping set of features. Instead, HMM uses local features (*current word, previous one or two tags*) for POS tagging. In our CRF experiment we are using only a small number of features (*current word, previous tag, prefix/suffix of length four*). This may be one of the reasons of relatively lesser accuracy of the Bengali tagging task. We also conducted experiments with large number of features but, the inclusion of a large features worse us the accuracy. Large number of features works well when a large amount of annotated training data is available to find a significant amount of every feature instance. This might be one of the reasons for relatively lower accuracy in inclusion of rich feature set.

Chapter 7

Conclusion

In this work we have exposed the research carried out on applying statistical and machine learning based algorithm to the POS tagging problem. We have worked on the language Bengali for POS disambiguation problem. We have used machine learning approaches to develop a part of speech tagger for Bengali. However no tagged corpus was available to us for use in this task. We had to start with creating tagged resources for Bengali. Manual part of speech tagging is quite a time consuming and difficult process. So we have worked with methods so that small amount of tagged resources can be used to effectively carry on the part of speech tagging task We have developed around 50,000 word annotated corpora for Bengali that has been used for the experiments.

In particular, we have used Hidden Markov Model to acquire statistical knowledge about part-of-speech ambiguities for the use in disambiguation algorithm. The HMM models described in this thesis are very simple and efficient for automatic tagging even when the amount of available labeled text is small. The models have a much higher accuracy than the naive baseline model. However, the performance of the current system is not as good as that of the best POS-tagger available for English and other European languages. The best performance is achieved for the supervised bigram HMM learning model along

with morphological restriction on the possible grammatical categories of a word and suffix information for handling unknown words. In fact, the use of MA in any of the models enhances the performance of the POS tagger significantly. We conclude that the use of morphological features is especially helpful to develop a reasonable POS tagger when tagged resources are limited.

Although HMM performs reasonably well for part-of-speech disambiguation task but, it uses local features (*current word, previous one or two tags*) for POS tagging. Uses of only local features may not work well for a morphologically rich and relatively free order word language – Bengali. Further, we plan to use other data driven statistical approaches, which use unrestricted and reach features in the framework of a probabilistic model. Maximum Entropy model and Conditional Random Fields are extremely flexible techniques for the above linguistic modelling.

In Chapter 5, we have described a Maximum Entropy based approach for automatic POS tagging natural language test for Bengali. Although simple ME based tagger performs reasonably better compare to the simple HMM (HMM-S1), we think none of the tagger is better than other in absolute terms when morphological restriction is applied on the set of tags. Due to the probabilistic formulation, HMM obtains the most likely sequence of tags using the linear sequence of observations, that is, HMM tries to maximize the most likely tag sequence globally for a given sequence of words. Instead, the ME based models locally maximize the conditional probability of a word being into a particular grammatical class.

It has been reported that ME based models performs slightly more accurately compare to markov models (Kazama, 2001; Zhao, 2004; McCallum, 2000). But in our experiment, ME based models achieve roughly the same accuracy as HMM on the Bengali corpus. The power of the ME model lies in its diverse and overlapping set of features. Instead, HMM uses local features (*current word, previous one or two tags*) for POS tagging. In our ME experiment

we are using only a small number of features (*current word, previous tag, prefix/suffix of length four*). This may be one of the reasons of relatively lesser accuracy of the Bengali tagging task. We also conducted experiments with large number of features but, the inclusion of a large features worse us the accuracy. Large number of features works well when a large amount of annotated training data is available to find a significant amount of every feature instance. This might be one of the reasons for relatively lower accuracy in inclusion of rich feature set.

The above observations motivated us to use other data driven statistical approaches, which use unrestricted and reach features in the framework of a probabilistic model. Conditional Random Fields are extremely flexible techniques for the above linguistic modeling, which uses arbitrary chain sequence of the Markov process and it can also incorporate large number of features.

In chapter 6, we have described our work on Bengali POS tagging using Conditional Random fields. We have used the same potential features of the Maximum Entropy model in the CRF framework to understand the relative performance of the models. CRF based taggers perform reasonably better compare to the HMM and ME based tagger. Due to the probabilistic formulation, HMM obtains the most likely sequence of tags using the linear sequence of observations, that is, HMM tries to maximize the most likely tag sequence globally for a given sequence of words. Instead, the ME based models locally maximize the conditional probability of a word being into a particular grammatical class. In contrast, Conditional Random Field (CRF) has a single exponential model for the joint probability of the entire sequence of states given the observation sequence.

The power of the CRF model lies in its diverse and overlapping set of features. Instead, HMM uses local features (*current word, previous one or two tags*) for POS tagging. In our CRF experiment we are using only a small number of features (*current word, previous tag, prefix/suffix of length four*). This may be one of the reasons of relatively lesser accuracy of the Bengali tagging task. We

also conducted experiments with large number of features but, the inclusion of a large features worse us the accuracy. Large number of features works well when a large amount of annotated training data is available to find a significant amount of every feature instance. This might be one of the reasons for relatively lower accuracy in inclusion of rich feature set

In summary, all the models described in this dissertation are very simple and efficient for automatic tagging of natural language text even when the amount of available annotated text is very small. The models have a much higher accuracy than the naive baseline model. However, the performance of the current system is not as good as that of the contemporary POS-tagger available for English and other European languages. The best performance is achieved for the supervised learning model along with suffix information and morphological restriction on the possible grammatical categories of a word. In fact, the use of MA in any of the models discussed above enhances the performance of the POS tagger significantly. The performance can be improved by increasing the size of the training data.

7.1. Contributions

The main contribution of the thesis can be categorized in the followings

- The application of machine learning techniques in the Bengali POS tagging is the basic objective of the work. We have applied three widely used machine learning techniques for the Bengali POS tagging problem i.e. Hidden Markov Model, Maximum Entropy based model and Conditional Random Fields. We have acquired very simple models based on the above machine learning techniques with a satisfactory test of the acquired models.
- First, we have used very simple HMM model for the Bengali POS tagging task. Since only a small only a small training set is available, a simple HMM based approach does not yield very good results. Further,

we have made use of semi-supervised learning by augmenting the small labeled training set provided with a larger unlabeled training set. Finally. We have used morphological analyzer to improve the performance of the tagger. The work is described in chapter 4.

- Machine learning techniques for accruing discriminative models have been applied for Bengali POS tagging task. We have used Maximum Entropy based model for the task. First, we have conducted different experiments to identify the best suited feature set from the potential feature set for the POS tagging task. In this work, we have also used a morphological analyzer to improve the performance of the tagger. The work is presented in chapter 5.
- In chapter 6, we present the uses of different features and their effective performance in the CRF based model. We have used the same features as we used in the Maximum Entropy framework to understand the relative performance of the two different conditional probability models. Further, we have used morphological analyzer to improve the performance of a tagger in the Conditional Random field framework.

The following two items describe other relevant contribution of the present thesis

- Chapter 2 provides a detailed survey of POS tagging and a broad coverage of compilation of references on different work in Indian Language for the POS tagging task.
- From a practical perspective, we would like to emphasize that a resources comprising of 50,000 POS annotated corpora has been developed as a result of the work. We have also presented a tagset for Bengali that has been developed as a part of the work. Chapter 3 points to the resource and related issues.

7.2. Future Works

Further work is still to be done in several directions. Some of this corresponds to development of resources, while others refer to specific details of implementation and tuning. Some of these can be taken up as immediate goals and others can be considered as a long term goals.

7.2.1. Immediate Goals

Regarding HMM based Bengali POS tagging algorithm, there are some possible extensions that have not been taken into consideration, and we think that they should be studied, i.e. better adjustment of symbol emission probability. In our experiment (as described in chapter 4), we have use smoothing only for the unobserved words. Methods for estimating these probabilities have already been described (e.g. the use of word ending suffix). Nevertheless, this method may fail due to a small amount of training text. To address the above problem, the probability of the unknown word tags can be approximated by the less probable word tags i.e. tags of the word occurring only once or twice. The problem can also be studied by considering the valid linguistic suffix of the word instead of considering the last few characters of a word.

For the Maximum Entropy model, the use of suffix information improves the tagging accuracy. The effect of using linguistic affixes can be studied instead of using last few character sequence as the suffix of a word. This can also be studied in case of CRF based POS tagging algorithm. Further, features can be investigated in both the ME and CRF framework to improve the tagging accuracy. However, the uses of very generic features (*i.e. previous word, next word etc.*) increase the number of feature function greatly. It might be possible that all the members of the generic features do not contribute significantly for the POS disambiguation task. Thus, the effect of inclusion of more specific features (*i.e. is previous word belongs to a particular set, is next word is from a particular set*) instead of the generic features can be studied in both ME and CRF based framework of POS tagging.

7.2.2. Long Term Goals

The taggers based on the above statistical models (HMM, ME and CRF) compensate some errors due to less amount of labeled data or presence of errors in the labeled training data. We investigate the confusion between each pair of grammatical tags by the outcome of the HMM, ME and CRF models compare to the actual tags for a sentence. Further, the context frame rules can be used to cope with the high error propagating confusion classes.

All in all, the development of a machine learning based good accuracy POS tagger requires a large amount of training data. The future work also includes the development of a large amount of annotated data which can be further used for training the system. The present tagger can be used for the initial annotation and the errors can be manually checked which otherwise a very difficult task to annotate large amount of corpus.

We also plan to explore some other machine learning algorithms (e.g. Support Vector Algorithm and Neural Networks) to understand their relative performance of POS Tagging task under the current experimental setup. HMM based models do not work well when the amount of annotated data is less. This might be due to the effect of transition probability over emission probability in the sequence identification. Support Vector Algorithm, Neural Networks or Decision Tree based algorithms might overcome the above situation.

List of Publications

Publications from the Thesis

1. Sandipan Dandapat, Sudeshna Sarkar and Anupam Basu. Automatic Part-of-Speech Tagging for Bengali: An approach for Morphologically Rich Languages in a Poor Resource Scenario. In *Proceedings of the Association of Computational Linguistics (ACL 2007)*, Prague, Czech Republic, pp. 221-224.
2. Sandipan Dandapat Part-of-Speech Tagging and Chunking with Maximum Entropy Model. In *Proceedings of the SPSAL Workshop, IJCAI.2007*.
3. Sandipan Dandapat and Sudeshna Sarkar. Part of Speech Tagging for Bengali with Hidden Markov Model. In *Proceedings of the NLP AI Machine Learning Contest*, http://ltrc.iiit.ac.in/nlpai_contest06. June 2006.
4. Sandipan Dandapat, Sudeshna Sarkar and Anupam Basu. A hybrid part-of-speech tagging and it's application to Bengali. In *Proceedings of the International Conference on Computational Intelligence (ICCI 2004)*, Istanbul, TURKEY, pp. 169-172.

Other Related Publications

1. Sujan Kumar Saha, Sanjay Chatterjee, Sandipan Dandapat, Sudeshna Sarkar and Pabitra Mitra. A Hybrid Approach for Named Entity Recognition in Indian Languages. *Accepted in Proceedings of the IJCNLP-08 Workshop on Named Entity Recognition for South and South East Asian Languages (NERSSEAL 08)* Hyderabad, India. (*accepted*) <http://ltrc.iiit.ac.in/ner-ssea-08/>

2. Tirthankar Dasgupta, Sandipan Dandapat, Anupam Basu. A Prototype Machine Translation System from Text-to-Indian Sign Language. *Accepted in Proceedings of the IJCNLP-08 Workshop NLP for Less Privileged Languages (NLPLPL 08)* Hyderabad, India. (accepted) <http://ltrc.iiit.ac.in/nlp-lpl-08/>
3. Debasis Mandal, Sandipan Dandapat, Mayank Gupta, Pratyush Banerjee, Sudeshna Sarkar. Bengali and Hindi to English Cross-language Text Retrieval under Limited Resources. In *Proceedings of 8th Workshop of the Cross-Language Evaluation Forum*, Budapest, Hungary, 19-21 Sept 2007, LNCS, Springer.
4. Sandipan Dandapat, Pabitra Mitra and Sudeshna Sarkar. Statistical Investigation of Bengali Noun-Verb (N-V) Collocations as Multi-word Expressions. In *Proceedings of the Symposium on Modeling and Shallow Parsing of Indian Languages (MSPIL 2006)*, IIT Bombay, INDIA, pp. 230-233, April 2006.

References

- Abney S., 1991 Parsing by chunks. In *Principle-based Parsing*. Kluwer Academic Publishers.
- Abney S. 1997. Part-of-speech tagging and partial parsing. In Ken Church, Steve Young, and Gerrit Bloothoof, editors, *Corpus-Based Methods in Language and Speech*. Kluwer, Dordrecht.
- Arulmozhi P., Rao R. K. and Sobha L., 2006. A Hybrid POS Tagger for a Relatively Free Word Order Language. In *Proceedings of the Modeling and Shallow Parsing of Indian Language (MSPIL)*, Bombay. 79-85.
- Asahara M. and Matsumoto Y., 2000. Extended models and tools for high-performance part-of-speech tagger. In *Proceedings of the 18th conference on Computational linguistics*, Saarbrücken, Germany. 21-27.
- Baum L. E., 1972. An inequality and associated maximization technique in statistical estimation for probabilistic functions of a Markov process. *Inequalities*, 3:1-8, 1972.
- Berger A. L., 1997. The improved iterative scaling algorithm: A gentle introduction, 1997.
- Bharati A., Chaitanya V. and Sangal R., 1995. Natural Language Processing: A Paninian Perspective. Prentice Hall India.
- Bharati A., Sharma D. M., Ramakrishnamacharyulu K. V., Sangal R., 2001. Guidelines for AnnCorra: An Introduction, *Technical Report TR-LTRC-14*, Language Technologies Research Centre, IIIT Hyderabad.
- <http://www.iiit.net/research/ltrc/Publications/Techreports/toc.html>
- Biemann C., 2007. Unsupervised Natural Language Processing using Graph Models. In *Proceedings of the NAACL-HLT Doctoral Consortium*, Rochester. 37-40.

- Black E., Jelinek F., Lafferty J., Mercer R. and Roukos S. 1992. Decision tree models applied to the labeling of text with parts-of-speech. In *Proceedings of the DARPA workshop on Speech and Natural Language*, Harriman, New York.
- Brants T., 2000. TnT – A statistical part-of-speech tagger. In *Proceedings of the 6th Applied NLP Conference*. 224-231.
- Brill E., 1992. A simple rule-based part-of-speech tagger. In *Proceedings of the 3rd Conference on Applied NLP*. 152-155.
- Brill E., 1995a. Transformation-based error-driven learning and Natural Language Processing: A case study in part-of-speech tagging. *Computational Linguistics*, 21(4): 543-565.
- Brill E. 1995b. Unsupervised learning of disambiguation rules for part of speech tagging. In *Proceedings of 3rd Workshop on Very Large Corpora Workshop, Massachusetts*.
- Brown P., Della Pietra V., de Souza P., Lai J. and Mercer R., 1992. Class-based n-gram Models of Natural Language. *Computational Linguistic*, 18(4):467-480.
- Cardie, C. 1993a. A case-based approach to knowledge acquisition for domain-specific sentence analysis. In *Proceedings of AAAI93*, 798-803.
- Chanod J. and Tapanainen P. 1995. Tagging French: comparing a statistical and a constraint-based method. In *Proceedings of the seventh conference on European chapter of the Association for Computational Linguistics*, Dublin, Ireland.
- Chopde A., 2001. ITRANS Version 5.30, <http://www.aczone.com/itrans/>
- Church K. W. 1988. A stochastic parts program and noun phrase parser for unrestricted text. *Proceedings of the second conference on Applied Natural Language Processing*. Austin, Texas, 136-143.
- Cutting D., Kupiec J, Pederson J. and Sibun P., 1992. A practical part-of-speech tagger. In *Proceedings of the 3rd Conference on Applied NLP*. 133-140.

-
- Daelemans W., Zavrel J., Berck P., and Gillis S.. 1996. Mbt: A memory-based part of speech tagger-generator. In *Proceedings of the Workshop on Very Large Corpora*, Copenhagen, Denmark. 14-27.
- Dalal, K. Nagaraj, U. Swant, S. Shelke and P. Bhattacharyya. 2007. Building Feature Rich POS Tagger for Morphologically Rich Languages: Experience in Hindi. In *Proceedings of ICON*, India.
- Dandapat S., Sarkar S. and Basu A., 2004. A hybrid part-of-speech tagging and it's application to Bengali. In *Proceedings of the International Conference on Computational Intelligence*, pp. 169-172
- Dandapat S. and Sarkar S., 2006. Part of Speech Tagging for Bengali with Hidden Markov Model. In *Proceedings of the NLP AI Machine Learning Contest*, http://ltrc.iiit.ac.in/nlpai_contest06.
- Dandapat S., Sarkar S. and Basu A. 2007. Automatic Part-of-Speech Tagging for Bengali: An approach for Morphologically Rich Languages in a Poor Resource Scenario. In *Proceedings of the Association of Computational Linguistics (ACL 2007)*, Prague, Czech Republic. 221-224.
- Darroch J. and Ratcliff D., 1972. Generalized Iterative Scaling for log-linear models, *Ann. Math. Statistics*, 43, 1470-1480.
- Dasgupta S. and Ng V., 2007. Unsupervised Part-of-Speech Acquisition from Resource-Scare Languages. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, Prague. 218-227.
- Dermatas E. and George K., 1995. Automatic stochastic tagging of natural language texts. *Computational Linguistics*, 21(2): 137-163.
- DeRose S. J., 1988. Grammatical category disambiguation by statistical optimization. *Computational Linguistics*, 14:31-39.
- Eineborg M. and Gambäck B. 1994. Tagging experiment using neural networks. In *Proceeding of the 9th Nordic Conference of Computational Linguistic*, Sweden. 71-81.

-
- Elworthy D., 1994. Does Baum-Welch re-estimation help taggers?. In *Proceedings of the fourth conference on Applied natural language processing*, Stuttgart, Germany
- Feldweg H., 1995. Implementation and evaluation of a German HMM for POS disambiguation. In *Proceedings of EACL-SIGDAT-95 Workshop*, Dublin, Ireland.
- Gimenez J. and Marquez L. 2003. Fast and accurate part-of-speech tagging: The SVM approach revisited. In *Proceedings of RANLP*. 158-165.
- Greene B. B. and Rubin G. M., 1971. Automatic grammatical tagging of English. Technical Report, Department of Linguistics, Brown University.
- Hladka B. and Ribarov K. 1998. Part of Speech Tags for Automatic Tagging and Syntactic Structures, *Issues of Valency and Meaning, Studies in Honor of Jarmila Panevova* edited by: Eva Hajicova, Prague.
- Harris Z., 1962. String analysis of the language structure. Mutton and Co., The Hauge.
- Haruno M. and Matsumoto Y., 1997. Mistake-driven mixture of hierarchical tag context trees. In *Proceedings of the eighth conference on European chapter of the Association for Computational Linguistics*, Madrid, Spain. 230-237.
- Heeman, P. A. and J. F. Allen. 1997. Incorporating POS tagging into language modelling. In *Proceedings of the 5th European Conference on Speech Communication and Technology (Eurospeech)*, Rhodes, Greece.
- Hindle D. 1989. Acquiring disambiguation rules from text. In *Proceedings of the 27th annual meeting on Association for Computational Linguistics*. Vancouver, British Columbia, Canada. 118-125.
- Jinshan M., Yu Z., Ting L. and Sheng L. 2004. A Statistical Dependency Parser of Chinese under Small Training Data.
- Karlsson F., Voutilainen A., Heikkila J. and Anttila A., 1995. Constraint Grammar: A Language-Independent System for Parsing Unrestricted Text. Mouton de Gruyter, Berlin.

-
- Kazama J., Miyao Y., and Tsujii J., 2001. A maximum entropy tagger with unsupervised hidden markov models. *In Proceedings of the 6th NLPRS*, pp. 333-340
- Kim J. H. and Kim G. C. 1996. Fuzzy network model for part-of-speech tagging under small training data. *Natural Language Engineering*, v.2 n.2, p.95-110.
- Klein S. and Simmons R., 1963. A computational approach to grammatical coding of English words. *Journal of the Association for Computing Machinery*, 10: 334-337.
- Kupiec J., 1992. Robust part-of-speech tagging using a Hidden Markov Model. *Computer Speech and Language*, 6.
- Lafferty J., McCallum A. and Pereira F., 2001. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of the Eighteenth International Conference on Machine Learning*. 282-289.
- Lezius W., Rapp R. and Wettler M., 1996 . A Morphology-System and Part-of-Speech Tagger for German. In *Results of the 3rd KONVENS Conference*, Mouton de Gruyter, Berlin, 369-378.
- Ma Q. and Isahara H. 1998. A multi-neuro tagger using variable lengths of contexts. In *Proceedings of the 17th international conference on Computational linguistics*, Montreal, Quebec, Canada. 802-806.
- Magerman D. M. 1995. Statistical decision-tree models for parsing. In *Proceedings of the 33rd annual meeting on Association for Computational Linguistics*, Cambridge, Massachusetts. 276-283.
- Maitra R., 2000. Inflectional Morphological Analyzers for Hindi and Bengali Languages. M. Tech. Thesis, Department of CSE, Indian Institute of Technology Kharagpur.
- Màrquez L. and Rodríguez H. 1998. Part of Speech Tagging Using Decision Trees. *Lecture Notes in AI 1398-C. Nédellec & C. Rouveirol (Eds.)*. *Proceedings of the 10th European Conference on Machine Learning, ECML'98*. Chemnitz, Germany.

-
- Marquez L., Padro L., and Rodriguez H. 1998. Improving tagging accuracy by using voting taggers. In *Proceedings of NLP + IA/TAL + AI*, New Brunswick, Canada. 149-155.
- Màrquez L., Padró L. and Rodríguez H., 2000. A Machine Learning Approach to POS Tagging. *Machine Learning*, v.39 n.1, p.59-91.
- Matsukawa T., Miller S. and Weischedel R., 1993. Example-based correction of word segmentation and part of speech labeling. In *Proceedings of the workshop on Human Language Technology*, Princeton, New Jersey.
- McCallum A., Freitag D., Pereira F. . 2000. Maximum Entropy Markov Model for Information Extraction and Segmentation. In *Proceedings of the Seventeenth International Conference on Machine Learning*. 591-598.
- Mcteer M., Schwartz R. and Weischedel R., 1991. Empirical studies in part-of-speech labeling. *Proceedings Of the 4th DARPA Workshop on Speech and Natural Language*, pp. 331-336.
- Merialdo B., 1994. Tagging English Text with a Probabilistic Model. *Computational Linguistics*, 20(2):155-171.
- Moreno-Torres, I., 1994. A morphological disambiguation tool (MDS). An application to Spanish. *ESPRIT BRA-7315 Aquilex II, Working Paper 24*.
- Mylonakis M., Simaan K. and Hwa R., 2007. Unsupervised Estimation of Noisy Channel Model. In *Proceedings of the 24th International Conference on Machine Learning*, Corvallis, Oregon. 665-672.
- Nakagawa T., Kudoh T. and Matsumoto Y. 2001. Unknown word guessing and part-of-speech tagging using support vector machines. In *Proceedings of the Sixth Natural Language Processing Pacific Rim Symposium*. 325-331.
- Nakamura M., Maruyama K., Kawabata T., Shikano K. 1990. Neural network approach to word category prediction for English texts. In *Proceedings of the 13th Conference on Computational Linguistics (COLING 90)*, Helsinki, Finland, 213-218.

-
- Oflazer K. and Kuruöz I. 1994. Tagging and morphological disambiguation of Turkish text. In *Proceedings of the fourth conference on Applied Natural Language Processing*, Stuttgart, Germany.
- Oflazer, K. and G. Tür. 1996. Combining hand-crafted rules and unsupervised learning in constraint-based morphological disambiguation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 69-81.
- Padró L.. 1998. A Hybrid Environment for Syntax-Semantic Tagging. Phd. Thesis, Dep. Llenguatges i Sistemes Informàtics. Universitat Politècnica de Catalunya, February. <http://www.lsi.upc.cs/~padro>.
- Padro M. and Padro L. 2004. Developing Competetive HMM PoS Tagger using Small Training Corpora.
- Pereira F., Singer Y. and Tishby N., 1995. Beyond word N-grams. In *Proceedings of the Third Workshop on Very Large Corpora*. Somerset, New Jersey, Association for Computational Linguistics. 95-106.
- Pietra S. D., Pietra V. D. and Lafferty J., 1995. Inducing features of random fields. *Technical Report CMU-CS-95-144*, Carnegie Mellon University, 1995.
- Pinto D., McCallum A., Wei X. And Croft W. B., 2003. Table extraction using conditional random fields. *Proceedings of the ACM SIGIR, 2003*.
- Ramshaw L. A. and Marcus M. P. 1995. Text chunking using transformation-based learning. In *Proc. Third Workshop on Very Large Corpora. ACL, 1995*
- Ratnaparkhii A., 1996. A maximum entropy part-of-speech tagger. In *Proceedings of the Eempirical Methods in Natural Language Processing Conference*. 133-142.
- Ray P. R., Harish V., Basu A. and Sarkar S., 2003. Part of Speech Tagging and Local Word Grouping Techniques for Natural Language Processing. In *Proceedings 1st International Conference on Natural Language Processing*.
- Ribarov K. 2000. Rule-based Tagging: Morphological Tagset versus Tagset of Analytical Functions. In *Proceedings of the 2nd International Conference on Language Resources & Evaluation (LREC 2000)*.

- Ristad E. S. and Thomas R.G., 1997. Hierarchical Non-Emitting Markov Models. In *Proc. 35th Ann. Meeting ACL*, Madrid. pp. 381-385.
- Ronald Rosenfeld. 1994, Adaptive Statistical Language Modeling: A Maximum Entropy Approach. Carnegie Mellon University, Ph.D. Thesis
- Roth D. and Zelenko D., 1998. Part of speech tagging using a network of linear separators. In *Proceedings of the 36th annual meeting on Association for Computational Linguistics* Montreal, Quebec, Canada.1136-1142.
- Samuelsson C., Voutilainen A. 1997. Comparing a linguistic and a stochastic tagger. In *Proceedings of the eighth conference on European chapter of the Association for Computational Linguistics (EACL)*, Madrid, Spain. 246-253.
- Sanchez F. and Nieto A. F., 1995. Development of a Spanish Version of the Xerox Tagger. Universidad Autonoma de Madrid
- Santorini B. 1990.Part-of-Speech tagging guidelines for the Penn Treebank project. *Technical report MS-CIS-90-47, Department of Computer and Information Science, University of Pennsylvania.*
<http://www.cis.upenn.edu/~treebank/home.html>
- Saul G. and Pereira F., 1997. Aggregate and mixedorder Markov models for statistical language processing. In *Proceedings of the 2nd International Conference on Empirical Methods in Natural Language Processing*.
- Schmid H. 1994. Probabilistic part-of-speech tagging using decision trees. In *Proceedings of the International Conference on New Methods in Language Processing*, Manchester, UK. 44-49.
- Schmid H. 1995. Improvements in part-of-speech tagging with an application to German. In *Proceedings of the EACL SIGDAT Workshop*, Dublin, Ireland.
- Schütze H. 1993. Part-of-speech induction from scratch. In *Proceedings of the 31st annual meeting on Association for Computational Linguistics*, Columbus, Ohio. 251-258.

-
- Schütze H. and Singer Y., 1994. Part-of-speech tagging using a Variable Memory Markov model. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, Las Cruces, New Mexico. 181-187.
- Sha F. and Pereira F., 2003. Shallow parsing with conditional random fields. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, , Edmonton, Canada.134-141.
- Shrivastav M., Melz R., Singh S., Gupta K. and Bhattacharyya P., 2006. Conditional Random Field Based POS Tagger for Hindi. In *Proceedings of the MSPIL*, Bombay,. 63-68.
- Singh S., Gupta K., Shrivastav M. and Bhattacharyya V. 2006. Morphological Richness Offset Resource Demand – Experience in constructing a POS Tagger for Hindi. In *Proceedings of COLLING/ACL 06*. 779-786.
- Teufel S., Schmid H., Heid U., and Schiller A., 1996. EAGLES validation (WP-4) task on tagset and tagger interaction. Technical report, IMS, Universitat Stuttgart.
- Toutanova K., Klein D., Manning C. and Singer Y., 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, Edmonton, Canada. 173-180
- Tseng H., Jurafsky D. and Manning C., 2005. Morphological features help pos tagging of unknown words across language varieties. In *Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing*.
- Tür G. and Oflazer K., 1998. Tagging English by path voting constraints. In *Proceedings of the 17th international conference on Computational linguistics*, Montreal, Quebec, Canada. 1277-1281.
- Tzoukermann E., Radev D. R., and Gale W.A., 1995. Combining linguistic knowledge and statistical learning in French part of speech tagging. In *Proceedings of the EAACL SIGDAT Worksho*, Dublin, Ireland. 51-59.

- Tzoukermann E., Radev D.R., Gale W.A., 1997. Tagging French without Lexical Probabilities - Combining Linguistic Knowledge and Statistical Learning. In *Natural Language Processing using Very Large Corpora*. Kluwer, (also Cmp-ig/97 10002, 10 oct 1997).
- Viterbi A. J., 1967. Error bounds for convolutional codes and an asymptotically optimal decoding algorithm. *IEEE Transaction on Information Theory*, 13:260-269.
- Wallach H. M., 2002. Efficient training of conditional random fields. Master's thesis, University of Edinburgh, 2002.
- Wauschkuhn O. 1995. The influence of tagging on the results of partial parsing in German corpora. *Proc. Fourth International Workshop on Paraing tehnoloyie(IWPT 95)*, Prague/Karlovy Vary, Czech Republic. 260-270.
- Wilks Y., and Stevenson M. 1997. Combining Independent Knowledge Sources for Word Sense Disambiguation. In *Proceedings of the Third Conference on Recent Advances in Natural Language Processing Conference (RANLP-97)*, Bulgeria. 1-7.
- Zavrel J. and Daelemans W., 2003. Feature-rich memory-based classification for shallow NLP and information extraction. In *Text Mining. Theoretical aspects and applications. Springer LCNS series*.
- Zhao Y., Wang X., Liu B. and Guan Y., 2004. Applying Class Triggers in Chinese POS Tagging Based on Maximum Entropy Model. In *Proceedings of the 3rd International Conference on Machine Learning and Cybernetics* . 26-29.

Appendix A

Lexical Categories (Tags) for Bengali

In this section we describe the tagset (the different lexical categories and subcategories) that has been used for our POS tagging experiments. The tagset was developed as a part of the project *SANKALAN*⁵ which was an initiative to build small but clean and completely tagged corpora for India languages. The tag set for Bengali has been designed considering the traditional grammar and lexical diversity. Some standard texts on Bengali Grammar were consulted and inputs from linguists were also sought. The tagged sentences were checked for suitability of parsing and information conservation. The tag sets were enhanced based on the observations. In this way, iteratively a final tagset was designed. We believe that the tag sets are complete and sufficient for our purpose, but we are open to modifications and would like to receive constructive suggestions. In order to further clarify the concepts several examples have been provided which are in the ITRANS⁶ notation.

1. Noun

1.1 Proper Noun (\NP)

Proper nouns are to be tagged as “\NP”. The list includes names of places or geographical entities e.g. *bhArata*{*India*}, *ga~NgA*{*Ganga*}, *dillI*{*Delhi*} etc., names of people, books, languages, organizations, scientific names of animals and plants etc. If a name spans more than a word then all the words must be tagged as proper noun separately and the attributes should be determined globally e.g.

sachchidAnanda\NP *hirAnanda*\NP *vAtsyAyana*\NP
Sachidananda *Hirananda* *Vatsayan*

⁵ http://www.cel.iitkgp.ernet.in/SANKALAN_techReport1.pdf

⁶ ITRANS version 5.30 <http://www.aczoom.com/itrans/>

Note that in Bengali proper nouns are often valid dictionary words. The tagging should be done based on semantics eg.

bhAratIya\NP kR^iShi\NP sa.nsthA\NP eka bhAratIya\JJ sa.nsthA\NN .
/Bharatiya/ /Krishi/ /Sanstha/ /an/ /Indian/ /organization/.
Bharatiya Krishi Sanstha is an Indian organization.

Here, *bhAratIya kR^iShi sa.nsthA* must be tagged as \NP since it is the name of an institute, whereas for the second part *bhAratIya* and *sa.nsthA* must be tagged as adjective (\JJ) and normal noun (\NN) respectively.

1.2 Verbal Noun (\NV)

The -A, -Ano and -aoYA forms of verb come under this category. The tag for the verbal noun category is “\NV”. Since verbal nouns act just as nouns in a sentence they can take inflections e.g.

lokerA Adara pAoYArA\NV janya oraO Adara karArA\NV darkArA .
/others/ /love/ /to get/ /he(also)/ /love/ /do/ /need/
To get others love he also needs to love .

ghuma pA.DAnora\NV gAna gAo.
/Sleep/ /to put/ /song/ /sing/
Sing a lullaby.

Certain variations are possible like *karabAra* instead of *karArA*, and *khAoYAbAra* instead of *khAoYAnora*. Note that POS tagging is purely syntactic.

rAmera oShudha khAoYA\NV shyAma lakSha karechhila .
/Ram/ /medicine/ /take/ /Shyam/ /notice/ /[PAST]/
The taking of medicine by Ram was noticed by Shyam

Therefore in constructs like the *khAoYA* should be tagged as \NV despite of the fact that strictly speaking it is not acting as a noun here.

1.3 Default Noun (\NN)

Any noun other than verbal and proper nouns should be tagged as default nouns \NN. This category also includes adjectives in nominal positions like

priyA khuba bhAla\NN

/Priya/ /very/ /good/

Priya is very good.

2. Number (\NUM)

Any numerical figure or spelt out numbers which are not adjectives or pronouns are tagged as \NUM, where cardinality is the value of that number. This category includes dates, years, time, phone numbers or any other numerical data. For example:

1947\NUM *sAle panera;i*\NUM *agAsTa rAta bAroTA*\NUM *bAjAra sathe sathe bhAratabarSha sbAdhIna haYa .*

At the stike of midnight on 15th Aug, 1947 Indian became independent

ekasha\NUM *egArake*\NUM *bilete apaYA ganya karA haYa .*

/hundred/ /eleven/ /England/ /unlucky/ /considered/

Hundred eleven is considered unlucky in England.

3. Pronoun

3.1. Personal Pronoun (\PP)

Personal pronouns are used in nominal positions meant for human or other animate agents for example *Ami*{I}, *tomAdera*{yours'}, *ApanAra*{your + Honorific}, *oTA*{that} etc. They are tagged as "\PP".

3.2. Cardinal Pronoun (\PC)

When numbers like *eka*{one} or inflected numerical forms like *hAjAra*{thousand}, *prathama*{first}, *dbitiYa*{second} etc. are used in nominal

positions referring to some noun entity, then they are tagged as \PC. Attributes are self explanatory. The following examples illustrate the concept.

eka\PC theke sahasra\PC hala AmAdera lokabala .
/one/ /to/ /thousand/ /has increased/ /our/ /manpower/
our man-power has increased from one to thousand

prathama gA.DiTA AsAra pare parei dbitiYaTara\PC AoYaja shonA gela .
/first/ /car/ /car/ /as soon as/ /second/ /sound/ /heard/
As soon as the first car came the sound of the second was heard

3.3. Ordinal Pronoun (\PO)

Words like *kichhu*, *saba*, etc. when used in nominal positions should be tagged as “\PO”. For example

sabAi\PO melAYa kichhu\PO nA kichhu\PO kinachhei .
/everyone/ /fair/ /something/ /or/ /other/ /bought/
Everyone bought something or the other at the fair

3.4. Question Mark (\PQ)

Words like *kothAYa*, *ki*, *ke* that are used for asking questions are to be tagged as “\PQ”. The attributes are self explanatory. Note that question markers can come in other contexts also (like determiner, adverb). In that case they should not be tagged as question marker. Example:

Apani kAdera\PQ sAthe dekhA karate kothAYa\PQ chalalena ?
Where are you going to meet whom?

3.5. Temporal Pronoun (\PT)

Words like *kAla*, *ekhana*, *Aja* etc. which denote time by reference to some other time (normally the present time) are tagged as “\PT”. Normally they are referred to as temporal adverbs, but as they actually sit in a nominal positions of dates and times, we have decided to mark them as temporal pronouns, even though they might later be grouped as adverbial phrases. Examples:

Ajake\PT kAlakera\PT mata;i bR^iShTi pa.Dachhe.
/today/ /yesterday/ /like/ /rain-ing/

Today [it's] raining just like yesterday

3.6. Spatial Pronoun (\PS)

Words like *ekhAna*, *okhAna*, *kothAo*, *dUra* etc. which denote place by reference are tagged as spatial pronouns “\PS”. The reason is similar to that of temporal pronouns. Example:

kichhu dUra\PS giYe se okhAna\PS theke hA.Nka chhA.Dala .
/some/ /distance/ /go/ /he/ /there/ /from/ /yelled/
After going some distance, he yelled.

4. *ityAdi*(\ETC)

The words like *ityadi* which acts as list continuation marker (etc.) or ellipsis marker (...) are tagged as “\ETC”.

5. Relative Pronoun

5.1. Relative Personal Pronoun (\RPP)

yArA, *yAdera* etc. are relative personal pronouns tagged as “\RPP.Person.Inflection”. For example:

yA.NrA\RPP kAla Asabena, tA.NrA mAchha bhAta khAbena .
/those who/ /tomorrow/ /will come/ /they/ /fish/ /rice/ /will eat/
Those who will come tomorrow will eat rice and fish

5.2. Relative Temporal Pronoun (\RPT)

yakhana is the relative temporal pronoun.

5.3. Relative Spatial Pronoun (\RPS)

yekhAne is the relative spatial pronoun

6. Post Position (\PP)

Post positions like *janya*, *sAthe*, *upara*, *Age*, *pare*, *madhye* are tagged as “\PP”. These are usually role markers often with an *-era* or *-ra* linkage from the previous noun. Example:

*surya oThAra Age\PP pA.Nkera madhye\PP madhye\PP padmaphula phuTe
chhila .*

Before sunrise Lotus flowers bloomed in the mud.

7. Adjective

7.1. Non-quantifying Adjectives (\JJ)

Adjectives like *ba.Da*, *lAla*, *sundara*, etc. come under this class. They are tagged as “\JJ.Degree.Gender”. The degree refers to normal, comparative or superlative cases. Thus, *bR^ihata* is normal; *bR^ihatatara* is comparative whereas *bR^ihatatama* is superlative. Example:

*lAla/JJ , Nilal/JJ , Ara nAnA/JJ ra~Ngera khuba sundara/JJ ramdhanu .
/red/ /, /blue/ /, //and/ /many/ /colors/ /very/ /beautiful/ /rainbow/
A very beautiful rainbow of red, blue and many other colors.*

7.2. Quantifying Adjectives

7.2.1 Cardinal Quantifying Adjective (\JQC)

Numbers when used as adjectives are tagged as “\JQC”. Example:

ekaTA\JQC ja~Ngale duTo\JQC bAgha thAkata.

7.2.2 Hedged Expressions (\JQH)

ekaTA-duTo, *pA.ncha-dasakhAnA*, *eka-Adha* are examples of hedged expressions, where an approximate range of numbers are provided instead of specific numbers. They are tagged as \JQC. Example:

*ekasha\JQH dusha\JQH lokake khAoYAnora janYa darakAra challisha-
pa~nchAsa\JQC kilo chAla.*

7.2.3 Quantifying Adjective (\JQQ)

Adjectives like *aneka*, *alpa*, *kichhu*, etc. which act as quantifiers are tagged as quantifying adjectives (\JQQ).

7.3. Following Adjectives(\JF)

This category consists of words which are either possessive pronouns like *AmAra*, *ApanAra* quantifying adjectives or ordinal pronouns like *saba*, *kichhu* etc. The difference from their normal usage is that they act as adjectives and **follow** the noun/pronoun which they qualify. They are tagged as \JF.Category. The following examples should clarify the concept.

oi ba;iTA AmAra\JF, oTA tomAra\PP naYa .
AmarA sabAi\JF bhArater sakala\DET rAjya dekhe phelechhi .
/We/ /all/ /India/ /all/ /states/ /have seen/
All of us have seen all the states of India.

8. Conjunction (\CNJ)

Ara, *bA*, *tabu*, *kintu*, *naYata* etc. are considered as conjunctions and tagged as \CNJ. The conditional conjunctions are not included here. Mathematical operators like *yoga*, *biyoga*, *guna* etc. when spelt out are also tagged as \CNJ. Example:

rAma Ara\CNJ shyAma khAbAra kheYe pa.Date bA\CNJ khelate chale gela.
/Ram/ /and/ /Shyam/ /food/ /after eating/ /to study/ /or/ /to play/ /went/
After eating food Ram and Shyam went to study or play.

9. Conditional (\CND)

yadi-tabe-nAhale or *yadi-tabe* groups indicating conditional statements (if-then-else) are tagged as \CND. Example:

yadi\CND barShA haYa tAhale\CND phasala bhAla habe nahale\CND saba
naShTa haYe yAbe .
If it rains, the crop will be good otherwise all will be destroyed.

10. Particles

10.1 to(\TO)

to is a sort of emphasis in Bengali. Example:

rAma to\TO yAbe. rAma yAbe to\TO ! nA to\TO.

10.2 Negative (\NEG)

nA is a negative marker are tagged as \NEG.

10.3 Shades (\SHD)

nA & *ye* are used in a rather idiosyncratic fashion in Bengali as participles.

Ami nA\SHD kAla yAba nA\NEG.

11. Interjection (\INJ)

Words like *Are*, *AhA*, *hAya* etc. are tagged as interjections (\INT). *dekha* and its forms are also marked as interjections when used for drawing attention. Example:

dekho\INT, kata phula phuTechhe !
/look/ /,/ /how many/ /flowers/ /have bloomed/
Look, how many flowers have bloomed!

12. Symbol (\SYM)

Symbols are characters which are not used as punctuation marks and neither are alphabets of the language. Examples are \$, @, &, +, % etc. Example:

mArkina yuktarAShtre mAthApichhu AYa prAYa \$\SYM 20000\NUM yeTA
bhAratera mAthApichhu Ayera 1000\NUM %\SYM beshi .
United States' per capita is \$2000 which is 1000% more than Indian per capita.

13. Foreign Word (\FW)

Words which are not of Hindi and neither has been assimilated in the language, i.e. are not dictionary words, are tagged as foreign words (\FW). Note that foreign proper nouns are tagged as \NP and not \FW. Example:

syAmasA~nera\NP mobAila\FW phona khuba bhAlo .
/Samsung's/ /mobile/ /phone/ /very/ /good/
Samsung's mobile phone is very good.
R^iShirA mAnuShake amR^itasya\FW putrAH\FW balechhena .

/sages/ /human/ /Amritasya/ /Putra/ /called/

The sages called humans “Amritasya Putra”

phona has not been marked as a foreign word because it is quite frequently used in Bengali and might be thought as an assimilated foreign word. Thus, it depends on the frequency of use of a particular word and tagger’s own judgment whether to tag a word as \FW or not.

14. Qualifier (\QUA)

Qualifiers qualify adjectives or adverbs. Some common examples are *khuba*, *bhIShaNa*, *ekaTu*, *bhAri* etc. Example:

khuba\QUA ba.Da ekaTA pirAmIDera sAmane bhAri\QUA sundara sphi.nksera murti Achhe .

In front of a huge Pyramid, there is a beautiful statue of a Sphinx.

15. Determiner

15.1. Relative Determiner (\DTR)

kona, *saba* etc. when used before other nouns or pronouns act as relative determiners and should be tagged as \DTR. It is called relative because no specific person/entity is referred to by these determiners. Example:

kona\DTR chora dharA pa.Dale saba\DTR pA.Dapa.DashirA beriYe pa.De tAke pulishera hAte tule debAra janya .

Whenever a thief is caught the entire neighborhood collects to hand him over to the police.

15.2. Absolute Determiner (\DTA)

When specific pronouns are used before other nouns or pronouns to refer to some particular person/entity, then these are tagged as \DTA. These are called absolute pronouns because here the reference is direct and are resolvable from the context. Example:

ai\DTA chheleTA sei\DTA bA.Dite Ara konadina Ashe ni .
 /that/ /boy/ /that/ /house/ /never returned/
That boy never returned to that house.

16. Sentential (\SEN)

Punctuation marks like ‘.’, ‘,’; ‘,’; ‘,’; ‘?’; ‘!’ are tagged as \SEN.

17. Adverb

17.1. Verbal Adverb (\AVB)

The *-te* form of the verb in duplication is used to describe adverbial participles in Bengali. They will be tagged as \AVB. Both the words must be tagged as \AVB. Example:

o gAnA gAite\AVB gAite\AVB nAchachhila .
 /she/ /while dancing/ /was dancing/
She was dancing while singing.

17.2. Avyaya (\ADV)

General adverbs are tagged as “\ADV” Degree refers to normal, superlative or comparative forms. Verb modifiers like *bate* and *baiki* are also placed in this category. The following examples should clarify the concepts.

gA.DiTA druta\ADV.N chAlAo.
 /car/ /very fast/ /drove/
[He] drove the car very fast.

chheleTA drutatara\ADV.C pA chAlAte lAgala.
 /the boy/ /very fast/ /walking/ /started/
The boy startted walking very fast.

Ami karaba baTe\ADV .

/I/ /will do/ /[definitely]/

I will [definitely] do [it]

Ami karaba baiki\ADV

/I/ /will do/ /surely/

I will [surely] do [it]..

18. Verb

18.1. Finite Verb (\VF)

All the verbs in the finite (i.e. assertive or negative form) are to be tagged as “\VF”.

rabindranAthera purbapuruSharA pirAli brAhmaNa chhilena\VF

/Rabindranath's/ /ancestors/ /Pirali/ /Bramhins/ /were/

Rabindranath's ancestors were Pirali Bramhins.

18.2. Non-finite Verb (\VN)

The ‘e’ form of the verb used in non-finite form is to be

18.3. Imperative/Subjunctive Verb (\VIS)

18.4. Negative Verb (\VNG)

verbs like *naYa*{is not}, *nei*{don't have} are tagged as “VNG”

18.5. Modal Verb (\VM)

Appendix B

Results obtained by Maximum Entropy based Bengali POS Tagger

The experiments have been carried out for different feature sets. The base feature sets are as follows:

- (1) Current word(w_i) and previous tag (t_{i-1})
- (2) Current word(w_i) and previous two tags (t_{i-1} and t_{i-2})
- (3) Current word(w_i), previous word (w_{i-1}) and previous tag (t_{i-1})
- (4) Current word(w_i), previous word (w_{i-1}) and previous two tags (t_{i-1} and t_{i-2})
- (5) Current word(w_i), next word (w_{i+1}) and previous two tags (t_{i-1} and t_{i-2})
- (6) Current word(w_i), previous word (w_{i-1}) and previous tag (t_{i-1})
- (7) Current word(w_i), previous two words (w_{i-1} and w_{i-2}) and previous tag (t_{i-1})
- (8) Current word(w_i), previous two words (w_{i-1} and w_{i-2}) and previous two tag (t_{i-1} and t_{i-2})

The following tables depict the detail result of the above eight different cases. A tick mark (\surd) indicates the use of that particular feature along with the basic features. All the accuracies are represented in the form of Overall accuracy (*known word accuracy, unknown word accuracy*). The experiments are conducted with the same train and test data (40,000 words and 5,000 words respectively) as described in the chapter3,4 and 5.

Case 1:

Base Features($\forall w_i$)	Prefix	Suffix	MA	Accuracy (%)
{ w_i, t_{i-1} } and t_i				84.56(89.95, 51.46)
		√		85.76(89.72, 61.44)
			√	87.38(89.25, 78.89)
		√	√	87.98(89.53, 78.42)
	√	√		86.78(89.72, 68.66)
	√	√	√	88.41(89.95, 78.95)

Case 2:

Base Features($\forall w_i$)	Prefix	Suffix	MA	Accuracy (%)
{ w_i, t_{i-1}, t_{i-2} } and t_i				83.64(88.95, 50.99)
		√		84.90(88.10, 65.24)
			√	86.94(89.57, 70.13)
		√	√	87.80(89.95, 74.66)
	√	√		86.30(89.27, 68.04)
	√	√	√	87.95(89.23, 77.12)

Case 3:

Base Features($\forall w_i$)	Prefix	Suffix	MA	Accuracy (%)
{w_i, w_{i-1}, t_{i-1} } and t_i				77.81(83.77,41.16)
		√		84.19(87.62,62.34)
			√	85.70(88.12, 70.13)
		√	√	86.02(88.5, 70.66)
	√	√		86.33(89.25,68.35)
	√	√	√	87.61(89.8, 73.91)

Case 4:

Base Features($\forall w_i$)	Prefix	Suffix	MA	Accuracy (%)
{ $w_i, w_{i-1}, t_{i-1}, t_{i-2}$ } and t_i				79.70(85.97,41.16)
		√		84.58(88.42,60.98)
			√	85.27(88.10,68.03)
		√	√	87.20(89.38, 73.66)
	√	√		86.45(89.32,68.81)
	√	√	√	87.81(89.75, 76.07)

Case 5:

Base Features($\forall w_i$)	Prefix	Suffix	MA	Accuracy (%)
{ $w_i, w_{i-1}, w_{i+1}, t_{i-1}, t_{i-2}$ } and t_i				52.57(55.88,32.26)
		√		81.50(84.47,57.14)
			√	82.06(84.15, 68.25)
		√	√	86.86(89.47, 70.97)
	√	√		86.66(89.62,68.57)
	√	√	√	88.21(89.77,78.65)

Case 6:

Base Features($\forall w_i$)	Prefix	Suffix	MA	Accuracy (%)
{ $w_i, w_{i-1}, w_{i+1}, t_{i-1}$ } and t_i				50.07(53.42,29.49)
		√		82.99(86.65,60.52)
			√	82.36(84.15, 70.75)
		√	√	86.97(88.75, 75.77)
	√	√		86.71(89.60, 68.97)
	√	√	√	88.00(89.87, 76.42)

Case 7:

Base Features($\forall w_i$)	Prefix	Suffix	MA	Accuracy (%)
{ $w_i, w_{i-1}, w_{i-2}, t_{i-1}, t_{i-2}$ } and t_i				65.72(70.98, 33.49)
		√		83.31(86.95, 60.98)
			√	83.31(85.48, 70.16)
		√	√	86.82(88.7, 75.35)
	√	√		86.06(89.12, 67.28)
	√	√	√	86.94(88.62, 76.81)

Case 8:

Base Features($\forall w_i$)	Prefix	Suffix	MA	Accuracy (%)
{ $w_i, w_{i-1}, w_{i-2}, t_{i-1}, t_{i-2}$ } and t_i				71.70(77.38, 36.87)
		√		83.12(86.75, 60.83)
			√	83.53(89.17, 69.17)
		√	√	86.84(88.82, 74.73)
	√	√		85.87(89.07, 66.20)
	√	√	√	87.44(89.37, 75.19)