Web-based Bengali News Corpus for Lexicon Development and POS Tagging

Asif Ekbal and Sivaji Bandyopadhyay

***Abstract*—Lexicon development and Part of Speech (POS) tagging are very important for almost all Natural Language Processing (NLP) applications. The rapid development of these resources and tools using machine learning techniques for less computerized languages requires appropriately tagged corpus. We have used a Bengali news corpus, developed from the web archive of a widely read Bengali newspaper. The corpus contains approximately 34 million wordforms. This corpus is used for lexicon development without employing extensive knowledge of the language. We have developed the POS taggers using Hidden Markov Model (HMM) and Support Vector Machine (SVM). The lexicon contains around 128 thousand entries and a manual check yields the accuracy of 79.6%. Initially, the POS taggers have been developed for Bengali and shown the accuracies of 85.56%, and 91.23% for HMM, and SVM, respectively. Based on the Bengali news corpus, we identify various word-level orthographic features to use in the POS taggers. The lexicon and a Named Entity Recognition (NER) system, developed using this corpus, are also used in POS tagging. The POS taggers are then evaluated with Hindi and Telugu data. Evaluation results demonstrates the fact that SVM performs better than HMM for all the three Indian languages.**

***Index Terms*—Web based corpus, lexicon, part of speech (POS) tagging, hidden Markov model(HMM), support vector machine (SVM), Bengali, Hindi, Telugu.**

I. Introduction

The mode of language technology work has changed dramatically since the last few years with the web being used as a data source in wide range of research activities. The web is anarchic, and its use is not in the familiar territory of computational linguistics. The web walked in to the ACL meetings started in 1999. The use of the web as a corpus for teaching and research on language has been proposed a number of times [1], [2], [3], [4]. There has been a special issue of the Computational Linguistics journal on Web as Corpus [5]. Several studies have used different methods to mine web data.

There is a long history of creating a standard for western language resources, such as EAGLES [1], PROLE/SIMPLE [6], ISLE/MILE [7], [8]. On the other hand, instead of having great linguistic and cultural diversities, Asian language resources have received much less attention than their western counterparts. An initiative [9] has started to create a common standard for Asian language resources.

Part of Speech (POS) tagging is the task of labeling each word in a sentence with its appropriate syntactic category called part of speech. Part of speech tagging is a very important preprocessing task for language processing activities. This helps in doing deep parsing of text and in developing Information extraction systems, semantic processing etc. Part of speech tagging for natural language texts are developed using linguistic rules, stochastic models and a combination of both. Stochastic models [10] [11] [12] have been widely used in POS tagging task for simplicity and language independence of the models. Among stochastic models, Hidden Markov Models (HMMs) are quite popular. Development of a stochastic tagger requires large amount of annotated corpus. Stochastic taggers with more than 95% word-level accuracy have been developed for English, German and other European languages, for which large labeled data are available. The problem is difficult for Indian languages (ILs) due to the lack of such annotated large corpus.

Simple HMMs do not work well when small amount of labeled data are used to estimate the model parameters. Incorporating diverse features in an HMM-based tagger is also difficult and complicates the smoothing typically used in such taggers. In contrast, a Maximum Entropy (ME) based method [13] or a Conditional Random (CRF) Field based method [14] or a SVM based system [15] can deal with diverse and overlapping features of the Indian languages. A POS tagger has been proposed in [16] for Hindi, which uses an annotated corpus of 15,562 words collected from the BBC news site, exhaustive morphological analysis backed by high coverage lexicon and a decision tree based learning algorithm (CN2). The accuracy was 93.45% for Hindi with a tagset of 23 POS tags.

International Institute of Information Technology (IIIT), Hyderabad, India initiated a POS tagging contest, NLPAI ML [2] for the Indian languages in 2006. Several teams came up with various approaches and the highest accuracies were 82.22% for Hindi, 84.34% for Bengali and 81.59% for Telugu. As part of the SPSAL Workshop[3] in IJCAI-07, a competition on POS tagging and chunking for south Asian languages was conducted by IIIT, Hyderabad. The best accuracies reported were 78.66% for Hindi [17], 77.61% for Bengali [18] and 77.37% for Telugu [17]. Other works for POS tagging in Bengali can be found in [19] with a ME approach and in [20] with a CRF approach.

Newspaper is a huge source of readily available documents.

Manuscript received May 4, 2008. Manuscript accepted for publication June 12, 2008.

Authors are with the Department of Computer Science and Engineering, Jadavpur University, Kolkata, India 700032, e-mail: asif.ekbal@gmail.com, sivaji_cse_ju@yahoo.com.

[1] http://www.ilc.cnr.it/Eagles96/home.html

[2] http://ltrc.iiitnet/nlpai_ contest06/proceedings.php

[3] http://shiva.iiit.ac.in/SPSAL2007/SPSAL-Proceedings.pdf

In the present work, we have used the corpus that has been developed from the web archive of a very well known and widely read Bengali newspaper. Bengali is the seventh popular language in the world, second in India and the national language in Bangladesh. Various types of news (International, National, State, Sports, Business etc.) are collected in the corpus and so a variety of linguistics features of Bengali are covered. We have developed a lexicon in an unsupervised way using this news corpus without using extensive knowledge of the language. We have developed POS taggers using HMM and SVM. The news corpus has been used to identify several orthographic word-level features to be used in POS tagging, particularly in the SVM model. We have used the lexicon and a NER system [21] as the features in the SVM-based POS tagger. These are also used as the means to handle the unknown words in order to improve the performance in both the models.

The paper is organized as follows. Section II briefly reports about the Bengali news corpus generation from the web. Section III discusses about the use of language resources particularly in lexicon development. Section IV describes the POS tagset used in the present work. Section V reports the development of POS tagger using HMM. Section VI deals with the development of POS tagger using SVM. Unknown word handling techniques are described in Section VII. Evaluation results of the POS tagger for Bengali, Hindi and Telugu are reported in Section VIII. Finally, Section IX concludes the paper.

II. Development of the Tagged Bengali News Corpus from the Web

The development of the Bengali news corpus is a sequence of language resource acquisition using a web crawler, language resource creation that includes HTML file cleaning, code conversion and language resource annotation that involves defining a tagset and subsequent tagging of the news corpus.

A web crawler has been developed for acquisition of language resources from the web archive of a leading Bengali newspaper. The web crawler retrieves the web pages in Hyper Text Markup Language (HTML) format from the news archive of a leading Bengali news paper within a range of dates provided as input. The news documents in the archive are stored in a particular fashion. The user has to give the range of dates as starting yy-mm-dd and ending yy-mm-dd format. The crawler generates the Universal Resource Locator (URL) address for the index (first) page of any particular date. The index page contains actual news page links and links to some other pages (e.g., Advertisement, TV schedule, Tender, Comics and Weather etc.) that do not contribute to the corpus generation. The HTML files that contain news documents are identified and the rest of the HTML files are not considered further.

The HTML files that contain news documents are identified by the web crawler and require cleaning to extract the Bengali text to be stored in the corpus along with relevant details. An HTML file consists of a set of tagged data that includes Bengali and English texts. The HTML file is scanned from the beginning to look for tags like <fontFACE = “Bengali Font Name”> . . . </font>, where the “Bengali Font Name“ is the name of one of the Bengali font faces as defined in the news archive. The Bengali texts in the archive are written in dynamic fonts and the Bengali pages are generated on the screen on the fly, i.e., only when the system is online and connected to the web. Moreover, the newspaper archive uses graphemic coding whereas orthographic coding is required for text processing tasks. Hence, Bengali texts, written in dynamic fonts are not suitable for text processing activities. In graphemic coding, a word is coded according to the constituent graphemes. But in orthographic coding the word is coded according to the constituent characters. In graphemic coding conjuncts have separate codes. But in orthographic coding it is coded in terms of the constituent consonants. A code conversion routine has been written to convert the dynamic codes used in the HTML files to represent Bengali text to ISCII codes. A separate code conversion routine has been developed for converting ISCII codes to UTF-8 codes.

TABLE I
News Corpus Tagset

Tag	Definition	Tag	Definition
header	Header of the news document	reporter	Reporter name
title	Headline of the news document	agency	Agency providing news
t1	1st headline of the title	location	The news location
t2	2nd headline of the title	body	Body of the news document
date	Date of the news document	p	Paragraph
bd	Bengali date	table	Information in tabular form
day	Day	tc	Table Column
ed	English date	tr	Table row

The Bengali news corpus developed from the web is annotated using a tagset that includes the *type* and *subtype* of the news, title, date, reporter or agency name, news location and the body of the news. A news corpus, whether in Bengali or in any other language has different parts like title, date, reporter, location, body etc. A news document is stored in the corpus in XML format using the tagset, mentioned in Table I. The *type* and *subtype* of the news item are stored as attributes of the *header*. The news items have been classified on geographic domain (International, National, State, District, Metro) as well as on topic domain (Politics, Sports, Business).

The news corpus contains 108,305 number of news documents with about five years (2001-2005) of news data collection. Some statistics about the tagged news corpus are presented in Table II. Details of corpus development are reported in [22].

III. Lexicon Development from the Corpus

An unsupervised machine learning method has been used for lexicon development from the Bengali news corpus. No extensive knowledge about the language is required except the knowledge of the different inflections that can appear with the different words in Bengali.

In Bengali, there are five different POS namely, noun, pronoun, verb, adjective, and indeclinable (postpositions, conjunctions, and interjections). Noun, verb and adjective belong

TABLE II
CORPUS STATISTICS

Total no. of news documents in the corpus	108,305
Total no. of sentences in the corpus	2,822,737
Average no. of sentences in a document	27
Total no. of wordforms in the corpus	33,836,736
Average no. of wordforms in a document	313
Total no. of distinct wordforms in the corpus	467,858

TABLE III
LEXICON STATISTICS

Iteration	1	2	3	4	5
News Documents	9737	19929	39924	69951	99651
Sentences	0.22	0.49	1.02	1.79	2.55
Wordforms	2.77	5.98	12.53	21.53	30.61
Distinct Wordforms	0.10	0.15	0.23	0.37	0.526
Root words	0.03	0.04	0.065	0.09	0.128

to the open class of POS in Bengali. Initially, all the words (inflected and uninflected) are extracted from the corpus and added to a database. A list of inflections that may appear with noun words is kept and it has 27 entries. In Bengali, verbs can be categorized into 20 different groups according to their spelling patterns and the different inflections that can be attached to them. Original wordform of a verb word often changes when any suffix is attached to it. At present, there are 214 different entries in the verb inflection list. Noun and verb words are tagged by looking at their inflections. Some inflections may be common to both nouns and verbs. In these cases, more than one root word will be generated for a wordform. The POS ambiguity is resolved by checking the number of occurrences of these possible root words along with the POS tags as derived from other wordforms. Pronoun and indeclinable are basically closed class of POS in Bengali and these are added to the lexicon manually. It has been observed that adjectives in Bengali generally occur in four different forms based on the suffixes attached. The first type of adjectives can form comparative and superlative degree by attaching the suffixes -*tara* and -*tamo* to the adjective word. These adjective stems are stored in the lexicon with adjective POS. The second set of suffixes (e.g. -*gato*, -*karo* etc.) identifies the POS of the wordform as adjective if only there is a noun entry of the desuffixed word in the lexicon. The third group of suffixes (e.g. -*janok*, -*sulav* etc.) identifies the POS of the wordform as adjective and the desuffixed word is included in the lexicon with noun POS. The last set of suffixes identifies the POS of the wordform as adjective.

The system retrieves the words from the corpus and creates a database of distinct wordforms. Each distinct wordform in the database is checked for pronoun and indeclinable. If the wordform is neither a pronoun nor an indeclinable, it is analyzed to identify the possible root word along with the POS tag obtained from inflection analysis. Different suffixes are compared with the end of a word. If any match is found then the remaining part of that word from the beginning is stored as a candidate root word for that inflected word along with the appropriate POS information. So, one or more [root word, POS] pairs are obtained after suffix analysis of a wordform. It may happen that wordform itself is a root word, so the [wordform, {all possible POS}] is also added to the previous candidate root word list. Two intermediate databases have been kept. A wordform along with the candidate [root word, POS] pairs is stored in one database. The other database keeps track of the distinct candidate [root word, POS] pairs along with its frequency of occurrence over the entire corpus. After suffix analysis of all distinct wordforms, the [root word, POS] pair that has highest frequency of occurrence over the entire corpus is selected from the candidate [root word, POS] pairs for the wordform. If the frequency of occurrences for two or more [root word, POS] pairs are same, the root word with the maximum number of characters is chosen as the possible root.

The corpus has been used in the unsupervised lexicon development. Table III shows the results using the corpus. Except news documents, the number of sentences, wordforms, distinct wordforms and root words are mentioned in millions. The lexicon has been checked manually for correctness and it has been observed that the accuracy is approximately 79.6%. The list of rootwords are automatically corrected to a large degree by using the named entity recognizer for Bengali [21] to identify the named entities in the corpus in order to exclude them from the lexicon. The number of root words increases as more and more news documents are considered in the lexicon development.

IV. POS TAGSET USED IN THE WORK

We have used a POS tagset of 26 POS tags, defined for the Indian languages. All the tags used in this tagset (IIIT, Hyderabad, India tag set) are broadly classified into three categories. The first category contains 10 tags that have been adopted with minor changes from the Penn tagset. The second category that contains 8 tags is a modification of similar tags in the Penn tagset. They have been designed to cater to some phenomena that are specific to Indian languages. The third category consists of 8 tags and has been designed exclusively for Indian languages.

- Group 1: NN-Noun, NNP-Proper noun, PRP-Pronoun, VAUX-Verb auxillary, JJ-Adjective, RB-Adverb, RP-Particle, CC-Conjunction, UH-Interjection, SYM-Special symbol.
- Group 2: PREP-Postposition, QF-Quantifiers, QFNUM-Quantifiers number, VFM-Verb finite main, VJJ-Verb non-finite adjectival, VRB-Verb non-finite adverbial, VNN-Verb non-finite nominal, QW-Question words.
- Group 3: NLOC-Noun location, INTF-Intensifier, NEG-Negative, NNC-Compound nouns, NNPC-Compound proper nouns, NVB-Noun in kriyamula, JVB-Adjective in kriyamula, RBVB-Adverb in kriyamula.

V. POS TAGGING USING HIDDEN MARKOV MODEL

A POS tagger based on Hidden Markov Model (HMM) [23] assigns the best sequence of tags to an entire sentence. Generally, the most probable tag sequence is assigned to each sentence following the Viterbi algorithm [24]. The task of POS tagging is to find the sequence of POS tags $T = t_1, t_2, t_3, \ldots t_n$ that is optimal for a word sequence $W =$

$w_1, w_2, w_3 \ldots w_n$. The tagging problem becomes equivalent to searching for $argmax_T P(T) * P(W|T)$, by the application of Bayes' law.

We have used trigram model, i.e., the probability of a tag depends on two previous tags, and then we have,
$P(T) = P(t_1|\$) \times P(t_2|\$, t_1) \times P(t_3|t_1, t_2) \times P(t_4|t_2, t_3) \times \ldots \times P(t_n|t_{n-2}, t-n-1)$, where, an additional tag '\$' (dummy tag) has been introduced to represent the beginning of a sentence.

Due to sparse data problem, the linear interpolation method has been used to smooth the trigram probabilities as follows:
$P'(t_n|t_{n-2}, t_{n-1}) = \lambda_1 P(t_n) + \lambda_2 P(t_n|t_{n-1}) + \lambda_3 P(t_n|t_{n-2}, t_{n-1})$ such that the λs sum to 1.
The values of λs have been calculated by the method given in [12].

To make the Markov model more powerful, ***additional context dependent features*** have been introduced to the emission probability in this work that specifies the probability of the current word depends on the tag of the previous word and the tag to be assigned to the current word. Now, we calculate $P(W|T)$ by the following equation:

$$P(W|T) \approx P(w_1|\$, t_1) \times P(w_2|t_1, t_2) \times \ldots \times P(w_n|t_{n-1}, t_n)$$

So, the emission probability can be calculated as

$$P(w_i|t_{i-1}, t_i) = \frac{freq(t_{i-1}, t_i, w_i)}{freq(t_{i-1}, t_i)}$$

Here also the smoothing technique is applied rather than using the emission probability directly. The emission probability is calculated as:
$P'(w_i|t_{i-1}, t_i) = \theta_1 P(w_i|t_i) + \theta_2 P(w_i|t_{i-1}, t_i)$, where θ_1, θ_2 are two constants such that all θs sum to 1.

The values of θs should be different for different words. But the calculation of θs for every word takes a considerable time and hence θs are calculated for the entire training corpus. In general, the values of θs can be calculated by the same method that was adopted in calculating λs.

VI. POS Tagging using Support Vector Machine

We have developed a POS tagger using Support Vector Machine (SVM). We identify the features from the news corpus to use in the SVM model. Performance of the POS tagger is improved significantly by adopting the various techniques for handling the unknown words. These include word suffixes, identified by observing the various wordforms of the Bengali news corpus. We have also used the lexicon and a NER system [21], developed with the help of news corpus.

A. Support Vector Machine

Support Vector Machines (SVMs), first introduced by Vapnik [25] [26], are relatively new machine learning approaches for solving two-class pattern recognition problems. SVMs are well-known for their good generalization performance, and have been applied to many pattern recognition problems. In the field of Natural Language Processing(NLP), SVMs are applied to text categorization, and are reported to have achieved high accuracy without falling into over-fitting even though with a large number of words taken as the features [27] [28]. Suppose, we have a set of training data for a two-class problem: $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$, where $\mathbf{x}_i \in R^D$ is a feature vector of the i-th sample in the training data and $y \in \{+1, -1\}$ is the class to which $\mathbf{x}_i$ belongs. In their basic form, a SVM learns a linear hyperplane that separates the set of positive examples from the set of negative examples with *maximal margin* (the margin is defined as the distance of the hyperplane to the nearest of the positive and negative examples). In basic SVM framework, we try to separate the positive and negative examples by the hyperplane written as:

$$(\mathbf{w}.\mathbf{x}) + b = 0 \qquad \mathbf{w} \in \mathbf{R}^n, b \in \mathbf{R}.$$

SVMs find the "optimal" hyperplane (optimal parameter $\overline{w}, b$) which separates the training data into two classes precisely. The linear separator is defined by two elements: a weight vector $\mathbf{w}$ (with one component for each feature), and a bias b which stands for the distance of the hyperplane to the origin. The classification rule of a SVM is:

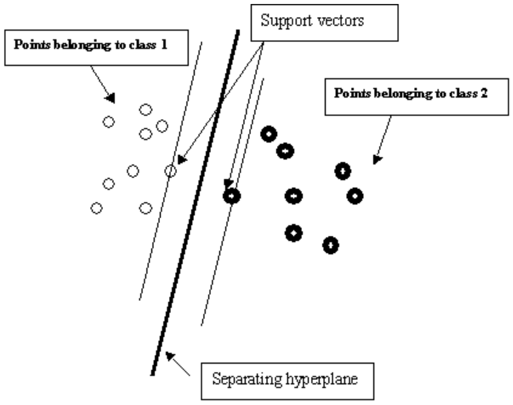

Fig. 1. Example of a 2-dimensional SVM

$$sgn(f(\mathbf{x}, \mathbf{w}, b)) \tag{1}$$

$$f(\mathbf{x}, \mathbf{w}, b) =< \mathbf{w}.\mathbf{x} > +b \tag{2}$$

being $\mathbf{x}$ the example to be classified. In the linearly separable case, learning the maximal margin hyperplane $(\mathbf{w}, b)$ can be stated as a convex quadratic optimization problem with a unique solution: *minimize* $||\mathbf{w}||$, *subject to the constraints* (one for each training example):

$$y_i(< \mathbf{w}.x_i > +b) \geq 1 \tag{3}$$

See an example of a 2-dimensional SVM in Figure 1.

The SVM model has an equivalent dual formulation, characterized by a weight vector α and a bias b. In this case, α contains one weight for each training vector, indicating the importance of this vector in the solution. Vectors with non null weights are called support vectors. The dual classification rule is:

$$f(\mathbf{x}, \alpha, b) = \sum_{i=1}^{N} y_i \alpha_i < \mathbf{x}_i.\mathbf{x} > +b \tag{4}$$

The α vector can be calculated also as a quadratic optimization problem. Given the optimal α^* vector of the dual quadratic optimization problem, the weight vector $\mathbf{w}^*$ that realizes the maximal margin hyperplane is calculated as:

$$\mathbf{w}^* = \sum_{i=1}^{N} y_i \alpha_i^* \mathbf{x}_i \qquad (5)$$

The b^* has also a simple expression in terms of $\mathbf{w}^*$ and the training examples $(\mathbf{x}_i, y_i)_{i=1}^N$.

The advantage of the dual formulation is that efficient learning of non-linear SVM separators, by introducing *kernel functions*. Technically, a *kernel function* calculates a dot product between two vectors that have been (non linearly) mapped into a high dimensional feature space. Since there is no need to perform this mapping explicitly, the training is still feasible although the dimension of the real feature space can be very high or even infinite.

By simply substituting every dot product of $\mathbf{x}_i$ and $\mathbf{x}_j$ in dual form with any *kernel function* $K(\mathbf{x}_i, \mathbf{x}_j)$, SVMs can handle non-linear hypotheses. Among the many kinds of *kernel functions* available, we will focus on the d-th polynomial kernel:

$$K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i.\mathbf{x}_j + 1)^d$$

Use of d-th polynomial kernel function allows us to build an optimal separating hyperplane which takes into account all combination of features up to d.

The SVMs have advantage over conventional statistical learning algorithms, such as Decision Tree, Hidden Markov Models, Maximum Entropy Models from the following two aspects:

1) SVMs have high generalization performance independent of dimension of feature vectors. Conventional algorithms require careful feature selection, which is usually optimized heuristically, to avoid overfitting. So, it can more effectively handle the diverse, overlapping and morphologically complex Indian languages.
2) SVMs can carry out their learning with all combinations of given features without increasing computational complexity by introducing the *Kernel function*. Conventional algorithms cannot handle these combinations efficiently, thus, we usually select “important” combinations heuristically with taking the trade-off between accuracy and computational complexity into consideration.

We have developed our system using SVM [27] [25], which perform classification by constructing an N-dimensional hyperplane that optimally separates data into two categories. Our general POS tagging system includes two main phases: training and classification. The training process was carried out by YamCha[4] toolkit, an SVM based tool for detecting classes in documents and formulating the POS tagging task as a sequential labeling problem. We have used TinySVM-0.07 [5] classifier that seems to be the best optimized among publicly available SVM toolkits. Here, the pairwise multi-class decision method and *second degree polynomial kernel function* have been used. In pairwise classification, we constructed K(K-1)/2 classifiers (here, K=26, no. of POS tags) considering all pairs of classes, and the final decision is given by their weighted voting.

B. Features for POS Tagging

Following are the details of the set of features that have been applied for POS tagging in Bengali.

• Context word feature: Preceding and following words of a particular word are used as features.

• Word suffix:Word suffix information is helpful to identify POS class. One way to use this feature is to consider a fixed length (say, *n*) word suffix of the current and/or the surrounding word(s). If the length of the corresponding word is less than or equal to *n-1* then the feature values are not defined and denoted by ND. The feature value is also not defined (ND) if the token itself is a punctuation symbol or contains any special symbol or digit. The second and the more helpful approach is to modify the feature as binary valued. Variable length suffixes of a word can be matched with the predefined lists of useful suffixes for different classes. This second type of suffixes include the noun, verb and adjective inflections. We have used both type of suffixes as the features.

• Word prefix: Prefix information of a word is also helpful. A fixed length (say, *n*) prefix of the current and/or the surrounding word(s) can be considered as features. This feature value is not defined (ND) if the length of the corresponding word is less than or equal to *n-1* or the word is a punctuation symbol or the word contains any special symbol or digit.

• Part of Speech (POS) Information: POS information of the previous word(s) might be used as a feature. This is the only dynamic feature in the experiment.

• Named Entity Information: The named entity (NE) information of the current and/or the surrounding word(s) plays an important role in the overall accuracy of the POS tagger. In order to use this feature, a CRF-based NER system [21] has been used. The NER system uses the NE classes namely, *Person name*, *Location name*, *Organization name* and *Miscellaneous name*. Date, time, percentages, numbers and monetary expressions belong to the *Miscellaneous name* category. The NER system was developed using a portion of the Bengali news corpus. This NER system has demonstrated 90.7% f-score value during 10-fold cross validation test with a training corpus of 150K wordforms.

The NE information can be used in two different ways. The first one is to use the NE tag(s) of the current and/or the surrounding word(s) as the features of SVM. The second way is to use this NE information at the time of testing. In order to do this, the test set is passed through the NER system. Outputs of the NER system are given more priorities than the outputs of the POS tagger for the unknown words in the test set. The NE tags are then replaced appropriately by the POS tags (NNPC: Compound proper noun, NNP: Proper noun and QFNUM: Quantifier number).

• Lexicon Feature: The lexicon has been used to improve the performance of the POS tagger. One way is to use this lexicon as the features of the SVM model. To apply this, five different features are defined for the open class of words as follows:

[4] http://chasen-org/ taku/software/yamcha/

[5] http://cl.aist-nara.ac.jp/ taku-ku/software/TinySVM

1) If the current word is found to appear in the lexicon with the ‘noun’ POS, then the feature ‘Lexicon’ is set to 1.
2) If the current word is found to appear in the lexicon with the ‘verb’ POS, then the feature ‘Lexicon’ is set to 2.
3) If the current word is found to appear in the lexicon with the ‘adjective’ POS, then the feature ‘Lexicon’ is set to 3.
4) If the current word is found to appear in the lexicon with the ‘pronoun’ POS, then the feature ‘Lexicon’ is set to 4.
5) If the current word is found to appear in the lexicon with the ‘indeclinable’ POS, then the feature ‘Lexicon’ is set to 5.

The second or the alternative way is to use this lexicon during testing. For an unknown word, the POS information extracted from the lexicon is given more priority than the POS information assigned to that word by the SVM model. An appropriate mapping has been defined from these five basic POS tags to the 26 POS tags. This is also used for handling the unknown words in the HMM model.

•Made up of digits: For a token if all the characters are digits then the feature “Digit” is set to 1; otherwise, it is set to 0. It helps to identify QFNUM (Quantifier number) tag.

•Contains symbol: If the current token contains special symbol (e.g., %, $ etc.) then the feature “ContainsSymbol” is set to 1; otherwise, it is set to 0. This helps to recognize SYM (Symbols) and QFNUM (Quantifier number) tags.

•Length of a word: Length of a word might be used as an effective feature of POS tagging. If the length of the current token is more than three then the feature ’LengthWord’ is set to 1; otherwise, it is set to 0. The motivation of using this feature is to distinguish proper nouns from the other words. We have observed that very short words are rarely proper nouns.

•Frequent word list: A list of most frequently occurring words in the training corpus has been prepared. The words that occur more than 10 times in the entire training corpus are considered to be the frequent words. The feature ‘FrequentWord’ is set to 1 for those words that are in this list; otherwise, it is set to 0.

•Function words: A list of function words has been prepared manually. This list has 743 number of entries. The feature ‘FunctionWord’ is set to 1 for those words that are in this list; otherwise, the feature is set to 0.

•Inflection Lists: Various inflection lists were created manually by analyzing the various classes of words in the Bengali news corpus during lexicon development. A simple approach of using these inflection lists is to check whether the current word contains any inflection of these lists and to take decision accordingly. A feature ’Inflection’ is defined in the following way:

1) If the current word contains any noun inflection then the feature ‘Inflection’ is set to 1.
2) If the current word contains any verb inflection then the value of ’Inflection’ is set to 2.
3) If the current word contains any adjective inflection, then the feature ‘Inflection’ is set to 3.
4) The value of the feature is set to 0 if the current word does not contain any noun, adjective or verb inflection.

VII. Unknown Word Handling Techniques for POS Tagging using HMM and SVM

Handling of unknown word is an important issue in POS tagging. For words, which were not seen in the training set, $P(t_i|w_i)$ is estimated based on the features of the unknown words, such as whether the word contains any particular suffix. The list of suffixes include mostly the noun, verb and adjective inflections. This list has 435 suffixes. The probability distribution of a particular suffix with respect to any specific POS tag is calculated from all words in the training set that share the same suffix.

In addition to the unknown word suffixes, the CRF-based NER system [21] and the lexicon have been used to tackle the unknown word problems. Details of the procedure is given below:

1) Step 1: Find the unknown words in the test set.
2) Step 2: The system assigns the POS tags, obtained from the lexicon, to those unknown words that are found in the lexicon. For noun, verb and adjective words of the lexicon, the system assigns the NN (Common noun), VFM (Verb finite main) and the JJ (Adjective) POS tags, respectively.
   Else
3) Step 3: The system considers the NE tags for those unknown words that are not found in the lexicon
   a) Step 2.1: The system replaces the NE tags by the appropriate POS tags (NNPC [Compound proper noun] and NNP [Proper noun]).

   Else
4) Step 4: The remaining words are tagged using the unknown word features accordingly.

VIII. Evaluation of Results of the POS Taggers

The HMM-based and SVM-based POS taggers are evaluated with the same data sets. Initially, the POS taggers are evaluated with Bengali by including the unknown word handling techniques, discussed earlier. We then evaluate the POS taggers with Hindi and Telugu data. The SVM-based system uses only the language independent features that are applicable to both Hindi and Telugu. Also, we have not used any unknown word handling techniques for Hindi and Telugu.

A. Data Sets

The POS tagger has been trained on a corpus of 72,341 tokens tagged with the 26 POS tags, defined for the Indian languages. This 26-POS tagged training corpus was obtained from the NLPAI ML Contest-2006[6] and SPSAL-2007[7] contest data. The NLPAI ML 2006 contest data was tagged with the 27 different POS tags and had 46,923 tokens. This POS tagged data was converted into the 26-POS [8]tagged data by defining an appropriate mapping. The SPSAL-2007 contest data was

[6] http://ltrc.iiitnet/nlpai_ contest06/data2
[7] http://shiva.iiit.ac.in/SPSAL2007
[8] http://shiva.iiit.ac.in/SPSAL2007/iiit_ tagset_ guidelines.pdf

TABLE IV
TRAINING, DEVELOPMENT AND TEST SET STATISTICS

Language	TRNT	NTD	TST	UTST	UTST (%)
Bengali1	72,341	15,000	35,000	8,890	25.4
Hindi	21,470	5,125	5,681	1,132	19.93
Telugu	27,513	6,129	5,193	2,375	45.74

tagged with 26 POS tags and had 25,418 tokens. Out of 72,341 tokens, around 15K tokens are selected as the development set and the rest has been used as the training set. The systems are tested with a gold standard test set of 35K tokens. We collect the data sets of Hindi and Telugu from the SPSAL-2007 contest. Gold standard test sets are used to report the evaluation results.

Statistics of the training, development and test set are presented in able IV. Following abbreviations are used in the table:

TRNT: No. of tokens in the training set

TST: No. of tokens in the test set

NTD: No. of tokens in the development set

UTST: No. of unknown tokens in the test set

B. Baseline Model

We define the *baseline* model as the one where the POS tag probabilities depend only on the current word:

$$P(t_1, t_2, \ldots, t_n | w_1, w_2, \ldots, w_n) = \prod_{i=1,\ldots,n} P(t_i, w_i).$$

In this model, each word in the test data will be assigned the POS tag, which occurred most frequently for that word in the training data. The unknown word is assigned the POS tag with the help of lexicon, named entity recognizer [21] and word suffixes for Bengali. For unknown words in Hindi and Telugu, some default POS tags are assigned.

C. Evaluation of Results of the HMM-based Tagger

Initially, the HMM based POS tagger has demonstrated an accuracy of 79.06% for the Bengali test set. The accuracy increases upto 85.56% with the inclusion of the different techniques, adopted for handling the unknown words. The results have been presented in Table V.

The POS tagger is then evaluated with Hindi and Telugu data. Evaluation results are presented in Table VI for the test sets.

It is observed from Table V- Table VI that the POS tagger performs best for the Bengali test set. The key to this higher accuracy, compared to Hindi and Telugu, is the mechanism of handling of unknown words. Unknown word features, NER system and lexicon features are used to deal with the unknown words in the Bengali test data. On the other hand, the system cannot efficiently handle the unknown words problem in Hindi and Telugu. Comparison between the performance of Hindi and Telugu shows that the POS tagger performs better with Hindi. One possible reason is the presence of large number of unknown words in the Telugu test set. Agglutinative nature of the Telugu language might be the another possible behind the fall in accuracy. The presence of the large number of unknown words in the Telugu test set. Agglutinative nature of the Telugu language might be the other possible reason behind the fall in accuracy.

TABLE VI
EXPERIMENTAL RESULTS OF HINDI AND TELUGU IN HMM

Language	Model	Accuracy (in %)
Hindi	Baseline	51.2
Hindi	HMM	73.75
Telugu	Baseline	40.87
Telugu	HMM	64.09

D. Evaluation Results of the SVM-based POS Tagger

We conduct a number of experiments in order to identify the best set of features for POS tagging in the SVM model by testing with the development set. We have also conducted several experiments by considering the various polynomial *kernel functions* and found that the system performs best for the polynomial *kernel function* of degree two. Also, it has been observed that the pairwise multi-class decision strategy performs better than the one-vs-rest strategy. The meanings of the notations, used in the experiments, are defined below:

pw, cw, nw: Previous, current and the next word

pwi, nwi: Previous and the next ith word

pre, suf: Prefix and suffix of the current word

ppre, psuf: Prefix and suffix of the previous word

pp: POS tag of the previous word

ppi: POS tag of the previous ith word

pn, cn, nn: NE tags of the previous, current and the next word

pni: NE tag of the previous ith word

$[i, j]$: Window of words spanning from the ith left position to the jth right position, where $i, j > 0$ indicates the words to the right of the current word, $i, j < 0$ indicates the words to the left of the current word, current word is at 0th position.

Evaluation results of the system for the development set are presented in Tables VII- VIII.

Evaluation results (3rd row) of Table VII show that word window $[-2, +2]$ gives the best result with the context window of size five, i.e., previous two and next two words along with the current word. Results also show the fact that further increase (4th and 5th rows) or decrease (2nd row) in window size reduces the accuracy of the POS tagger. Experimental results (6th and 7th rows) show that the accuracy of the POS tagger can be improved by including the dynamic POS information of the previous word(s). Clearly, it is seen that POS information of the previous two words are more effective and increases the accuracy of the POS tagger to 66.93%. Experimental results (8th-10th rows) show the effectiveness of prefixes and suffixes upto a particular length for the highly inflective Indian languages as like Bengali. The prefixes and suffixes of length upto three characters are more effective. Results (10th row) suggest that inclusion of surrounding word suffixes and/or prefixes reduces the accuracy.

It can be decided from the results (2nd-5th rows) of Table VIII that the named entity (NE) information of the current and/or the surrounding word(s) improves the overall accuracy of the POS tagger. It is also indicative from this results (3rd

TABLE V
EXPERIMENTAL RESULTS OF THE TEST SET FOR BENGALI IN HMM

Model	Accuracy (in %)
Baseline	55.9
HMM	79.06
HMM + Lexicon (Unknown word handling technique)	81.87
HMM +Lexicon (Unknown word handling) + NER (Unknown word handling technique)	83.09
HMM+ Lexicon (Unknown word handling) + NER (Unknown word handling) + Unknown word features	85.56

TABLE VII
RESULTS OF THE DEVELOPMENT SET FOR BENGALI IN SVM

Feature (word, tag)	Accuracy (in %)
pw, cw, nw	63.27
pw2, pw, cw, nw, nw2	64.32
pw3, pw2, pw, cw, nw, nw2, nw3	63.53
pw3, pw2, pw, cw, nw, nw2	64.16
pw2, pw, cw, nw, nw2, pp	66.08
pw2, pw, cw, nw, nw2, pp, pp2	66.93
pw2, pw, cw, nw, nw2, pp, pp2, $\|pre\| \leq 4$, $\|suf\| \leq 4$	70.97
pw2, pw, cw, nw, nw2, pp, pp2, $\|pre\| \leq 3$, $\|suf\| \leq 3$	71.34
pw2, pw, cw, nw, nw2, pp, pp2, $\|pre\| \leq 3$, $\|suf\| \leq 3$, $\|ppre\| \leq 3$, $\|psuf\| \leq 3$	70.23

TABLE VIII
RESULTS OF THE DEVELOPMENT SET FOR BENGALI IN SVM

Feature (word, tag)	Accuracy (in %)
pw2, pw, cw, nw, nw2, pp, pp2, $\|pre\| \leq 3$, $\|suf\| \leq 3$, pn, cn, nn	73.31
pw2, pw, cw, nw, nw2, pp, pp2, $\|pre\| \leq 3$, $\|suf\| \leq 3$, pn, cn	74.03
pw2, pw, cw, nw, nw2, pp, pp2, $\|pre\| \leq 3$, $\|suf\| \leq 3$, cn, nn	73.86
pw2, pw, cw, nw, nw2, pp, pp2, $\|pre\| \leq 3$, $\|suf\| \leq 3$, cn	73.08
pw2, pw, cw, nw, nw2, pp, pp2, $\|pre\| \leq 3$, $\|suf\| \leq 3$, pn, cn, Digit, Symbol, Length, FrequentWord, FunctionWord	77.43
pw2, pw, cw, nw, nw2, pp, pp2, $\|pre\| \leq 3$, $\|suf\| \leq 3$, pn, cn, Digit, Symbol, Length, FrequentWord, FunctionWord, Lexicon	82.82
pw2, pw, cw, nw, nw2, pp, pp2, $\|pre\| \leq 3$, $\|suf\| \leq 3$, pn, cn, Digit, Symbol, Length, FrequentWord, FunctionWord, Lexicon, Inflection	86.08

row) that the NE information of the previous and current words, i.e, within the window $[-1, 0]$ is more effective than the NE information of the windows $[-1, +1]$, $[0, +1]$ or the current word alone. An improvement of 3.4% in the overall accuracy is observed with the use of 'Symbol', 'Length', 'FrequentWord', 'FunctionWord' and 'Digit' features. The use of lexicon as the features of SVM model further improves the accuracy by 5.39% (7th row). Accuracy of the POS tagger rises to 86.08% (8th row), an improvement of 3.26%, by including the noun, verb and adjective inflections.

Evaluation results of the POS tagger by including the various mechanisms for handling the unknown words are presented in Table IX for the development set. The table also shows the result of the *baseline* model. Results demonstrate the effectiveness of the use of various techniques for handling the unknown words. Accuracy of the POS tagger increases by 5.44% with the use of lexicon, named entity recognizer [21] and unknown word features.

A gold standard test set of 35K tokens are used to report the evaluation results of the system. Experimental results of the system along with the *baseline* model are presented in Table X for the test set. The SVM-based POS tagger has demonstrated an accuracy of 85.46% with the various contextual and orthographic word-level features. Finally, the POS tagger has shown the overall accuracy of 91.23%, which is an improvement of 5.77% by using the various techniques for handling the unknown words.

In order to evaluate the POS tagger with Hindi and Telugu, we retrain the SVM model with the following language independent features that are applicable to both the languages.

1) Context words: Preceding two and following two words.
2) Word suffix: Suffixes of length upto three characters of the current word.
3) Word prefix: Prefixes of length upto three characters of the current word.
4) Dynamic POS information: POS tags of the current and previous word.
5) Made up of digits: Check whether current word consists of digits.
6) Contains symbol: Check whether the current word contains any symbol.
7) Frequent words: a feature is set appropriately for the most frequently occurring words in the training set.
8) Length: Check whether the length of the current word is less than three.

Experimental results are presented in Table XI for Hindi and Telugu. Results show that the system performs better for Hindi with an accuracy of 77.08%. Accuracy of the system for Telugu is 68.15%, which is less than 19.93% compared to Hindi. The *baseline* model has demonstrated the accuracies of 53.89%, and 42.12% for Hindi, and Telugu, respectively.

TABLE IX
Results of the development set for Bengali with unknown word handling mechanisms in SVM

Feature (word, tag)	Accuracy (in %)
Baseline	55.9
SVM	86.08
SVM + Lexicon (Unknown word handling technique)	88.27
SVM +Lexicon (Unknown word handling) + NER (Unknown word handling technique)	89.13
SVM+ Lexicon (Unknown word handling) + NER (Unknown word handling) + Unknown word features	91.52

TABLE X
Experimental results of the test set for Bengali in SVM

Feature (word, tag)	Accuracy (in %)
Baseline	54.7
SVM	85.46
SVM + Lexicon (Unknown word handling technique)	88.15
SVM +Lexicon (Unknown word handling) + NER (Unknown word handling technique)	90.04
SVM+ Lexicon (Unknown word handling) + NER (Unknown word handling) + Unknown word features	91.23

TABLE XI
Experimental results of Hindi and Telugu in SVM

Language	Set	Accuracy (in %)
Hindi	Development	78.16
Hindi	Test	77.08
Telugu	Development	68.81
Telugu	Test	68.15

E. Error Analysis

For Bengali gold standard test set, we conducted error analysis for each of the models (HMM and SVM) of the POS tagger with the help of confusion matrix. A close scrutiny of the confusion matrix suggests that some of the probable tagging errors facing the current POS tagger are NNC vs NN, JJ vs NN, JJ vs JVB, VFM vs VAUX and VRB vs NN. A multiword extraction unit for Bengali would have taken care of the NNC vs NN problem. The other ambiguities can be taken care of with the use of linguistic rules.

IX. Conclusion

In this work, we have used a Bengali news corpus, developed from the web-archive of leading Bengali newspaper, for lexicon development and POS tagging. Lexicon has been developed in an unsupervised way and contains approximately 0.128 million entries. Manual check of the lexicon has shown an accuracy of 79.6%. We have developed POS taggers using HMM and SVM. The POS tagger has shown the highest accuracy of 91.23% for Bengali in the SVM model. This is an improvement of 5.67% over the HMM-based POS tagger. Evaluation results of the POS taggers for Hindi and Telugu have also shown better performance in the SVM model. The SVM-based POS tagger has demonstrated the accuracies of 77.08%, and 68.81% for Hindi, and Telugu, respectively. Thus, it can be decided that SVM is more effective than HMM to handle the highly inflective Indian languages.

References

[1] M. Rundell, “The Biggest Corpus of All,” *Humanising Language Teaching*, vol. 2, no. 3, 2000.

[2] W. H. Fletcher, “Concordancing the Web with KWiCFinder,” in *Proceedings of the Third North American Symposium on Corpus Linguistics and Language Teaching*, 23-25 March 2001.

[3] T. Robb, “Google as a Corpus Tool?,” *ETJ Journal*, vol. 4, no. 1, Spring 2003.

[4] W. H. Fletcher, “Making the Web More Use-ful as Source for Linguists Corpora,” *In Ulla Conor and Thomas A. Upton (eds.), Applied Corpus Linguists: A Multidimensional Perspective*, pp. 191–205, 2004.

[5] A. Kilgarriff and G. Grefenstette, “Introduction to the Special Issue on the Web as Corpus,” *Computational Linguistics*, vol. 29, no. 3, pp. 333–347, 2003.

[6] A. Lenci, N. Bel, F. Busa, N. Calzolari, E. Gola, M. Monachini, A. Ogonowsky, I. Peters, W. Peters, N. Ruimy, M. Villegas, and A. Zampolli, “Simple: A General Framework for the Development of Multilingual Lexicons,” *International Journal of Lexicography, Special Issue, Dictionaries, Thesauri and Lexical-Semantic Relations*, vol. XIII, no. 4, pp. 249–263, 2000.

[7] N. Calzolari, F. Bertagna, A. Lenci, and M. Monachini, “Standards and Best Practice for Multilingual Computational Lexicons, mile (the multilingual isle lexical entry),” *ISLE Deliverable D2.2 & 3.2*, 2003.

[8] F. Bertagna, A.Lenci, M. Monachini, and N. Calzolari, “Content interoperability of lexical resources, open issues and ‘mile’ perspectives,” in *Proceedings of the LREC 2004*, pp. 131–134, 2004.

[9] T. Takenobou, V. Sornlertlamvanich, T. Charoenporn, N. Calzolari, M. Monachini, C. Soria, C. Huang, X. YingJu, Y. Hao, L. Prevot, and S. Kiyoaki, “Infrastructure for Standardization of Asian Languages Resources,” in *Proceedings of the COLING/ACL 2006*, pp. 827–834, 2006.

[10] D. Cutting, J. Kupiec, J. Pedersen, and P. Sibun, “A Practical Part-of-Speech Tagger,” in *Proceedings of the Third Conference on Applied Natural Language Processing*, pp. 133–140, 1992.

[11] B. Merialdo, “Tagging English Text with a Probabilistic Model,” *Computational Linguistics*, vol. 20, no. 2, pp. 155–171, 1994.

[12] T. Brants, “TnT: A Statistical Part-of-Speech Tagger,” in *Proceedings of the sixth International Conference on Applied Natural Language Processing ANLP-2000*, pp. 224–231, 2000.

[13] A. Ratnaparkhi, “A maximum entropy part-of -speech tagger,” in *Proc. of EMNLP’96.*, 1996.

[14] J. Laffertey, A. McCallum, and F. Pereira, “Conditional random fields: Probabilistic models for segmenting and labeling sequence data,” in *Proceedings of the 18th International Conference on Machine Learning*, 2001.

[15] T. Kudo and Y. Matsumoto, “Chunking with Support Vector Machines,” in *Proceedings of NAACL*, pp. 192–199, 2001.

[16] S. Singh, K. Gupta, M. Shrivastava, and P. Bhattacharyya, “Morphological richness offsets resource demand-experiences in constructing a pos tagger for hindi,” in *Proceedings of the COLING/ACL 2006*, pp. 779–786, 2006.

[17] P. Avinesh and G. Karthik, “Part Of Speech Tagging and Chunking using Conditional Random Fields and Transformation Based Learning,” in *Proceedings of IJCAI Workshop on Shallow Parsing for South Asian Languages*, pp. 21–24, 2007.

[18] S. Dandapat, “Part Of Specch Tagging and Chunking with Maximum Entropy Model,” in *Proceedings of the IJCAI Workshop on Shallow Parsing for South Asian Languages*, (Hyderabad, India), pp. 29–32, 2007.

[19] A. Ekbal, R. Haque, and S. Bandyopadhyay, “Maximum Entropy based Bengali Part of Speech Tagging,” in *A. Gelbukh (Ed.), Advances in Natural Language Processing and Applications, Research in Computing Science (RCS) Journal*, vol. 33, pp. 67–78.

[20] A. Ekbal, R. Haque, and S. Bandyopadhyay, “Bengali Part of Speech Tagging using Conditional Random Field,” in *Proceedings of the seventh International Symposium on Natural Language Processing, SNLP-2007*, 2007.

[21] A. Ekbal, R. Haque, and S. Bandyopadhyay, “Named Entity Recognition in Bengali: A Conditional Random Field Approach,” in *Proceedings of 3rd International Joint Conference Natural Language Processing (IJCNLP-08)*, pp. 589–594, 2008.

[22] A. Ekbal and S. Bandyopadhyay, “A Web-based Bengali News Corpus for Named Entity Recognition,” *Language Resources and Evaluation Journal*, vol. 40, pp. 10.1007/s10579–008–9064–x, 2008.

[23] D. Jurafsky and J. H. Martin, *Speech and Language Processing*. Prentice-Hall, 2000.

[24] A. J. Viterbi, “Error bounds for convolutional codes and an asymptotically optimum decoding algorithm,” *IEEE Transaction on Information Theory*, vol. 13, no. 2, pp. 260–267, 1967.

[25] V. N. Vapnik, *The Nature of Statistical Learning Theory*. New York, NY, USA: Springer-Verlag New York, Inc., 1995.

[26] C. C and V. N. Vapnik, “Support Vector Networks,” *Machine Learning*, vol. 20, pp. 273–297, 1995.

[27] T. Joachims, *Making Large Scale SVM Learning Practical*, pp. 169–184. Cambridge, MA, USA: MIT Press, 1999.

[28] H. Taira and M. Haruno, “Feature Selection in SVM Text Categorization,” in *Proceedings of AAAI-99*, 1999.