Proceedings of the Twelfth Conference of the European Chapter of the Association for
Computational Linguistics (EACL), Athens, Greece, March 2009, pp. 363-371.

Weakly Supervised Part-of-Speech Tagging for Morphologically-Rich, Resource-Scarce Languages

Kazi Saidul Hasan and **Vincent Ng**
Human Language Technology Research Institute
University of Texas at Dallas
Richardson, TX 75083-0688
{saidul,vince}@hlt.utdallas.edu

Abstract

This paper examines unsupervised approaches to part-of-speech (POS) tagging for morphologically-rich, resource-scarce languages, with an emphasis on Goldwater and Griffiths's (2007) fully-Bayesian approach originally developed for English POS tagging. We argue that existing unsupervised POS taggers unrealistically assume as input a perfect POS lexicon, and consequently, we propose a weakly supervised fully-Bayesian approach to POS tagging, which relaxes the unrealistic assumption by automatically acquiring the lexicon from a small amount of POS-tagged data. Since such relaxation comes at the expense of a drop in tagging accuracy, we propose two extensions to the Bayesian framework and demonstrate that they are effective in improving a fully-Bayesian POS tagger for Bengali, our representative morphologically-rich, resource-scarce language.

1 Introduction

Unsupervised POS tagging requires neither manual encoding of tagging heuristics nor the availability of data labeled with POS information. Rather, an unsupervised POS tagger operates by only assuming as input a POS lexicon, which consists of a list of possible POS tags for each word. As we can see from the partial POS lexicon for English in Figure 1, "the" is *unambiguous* with respect to POS tagging, since it can only be a determiner (DT), whereas "sting" is *ambiguous*, since it can be a common noun (NN), a proper noun (NNP) or a verb (VB). In other words, the lexicon imposes constraints on the possible POS tags of each word, and such constraints are then used by an unsupervised tagger to label a new sentence. Conceivably, tagging accuracy decreases with the increase in ambiguity: unambiguous words such as "the" will always be tagged correctly; on the other hand, *unseen* words (or words not present in the POS lexicon) are among the most ambiguous words, since they are not constrained at all and therefore can receive any of the POS tags. Hence, unsupervised POS tagging can present significant challenges to natural language processing researchers, especially when a large fraction of the words are ambiguous. Nevertheless, the development of unsupervised taggers potentially allows POS tagging technologies to be applied to a substantially larger number of natural languages, most of which are resource-scarce and, in particular, have little or no POS-tagged data.

Word	POS tag(s)
...	...
running	NN, JJ
sting	NN, NNP, VB
the	DT
...	...

Figure 1: A partial lexicon for English

The most common approach to unsupervised POS tagging to date has been to train a hidden Markov model (HMM) in an unsupervised manner to maximize the likelihood of an unannotated corpus, using a special instance of the expectation-maximization (EM) algorithm (Dempster et al., 1977) known as Baum-Welch (Baum, 1972). More recently, a fully-Bayesian approach to unsupervised POS tagging has been developed by Goldwater and Griffiths (2007) [henceforth G&G] as a viable alternative to the traditional maximum-likelihood-based HMM approach. While unsupervised POS taggers adopting both approaches have demonstrated promising results, it is important to note that they are typically evaluated by assuming the availability of a *perfect* POS lexicon. This assumption, however, is fairly unrealistic in practice, as a perfect POS lexicon can only be constructed by having a linguist manually label each word in a language with its possible POS tags.[1] In other words, the labor-intensive POS lexicon construction process renders unsupervised POS taggers a lot less unsupervised than they appear. To make these unsupervised taggers practical, one could attempt to automatically construct a POS lexicon, a task commonly known as *POS induction*. However, POS induction is by no means an easy task, and it is not clear how well unsupervised POS taggers work when used in combination with an automatically constructed POS lexicon.

The goals of this paper are three-fold. First, motivated by the successes of unsupervised approaches to English POS tagging, we aim to investigate whether such approaches, especially G&G's fully-Bayesian approach, can deliver similar performance for Bengali, our representative resource-scarce language. Second, to relax the unrealistic assumption of employing a perfect lexicon as in existing unsupervised POS taggers, we propose a *weakly supervised* fully-Bayesian approach to POS tagging, where we automatically construct a POS lexicon from a small amount of POS-tagged data. Hence, unlike a perfect POS lexicon, our automatically constructed lexicon is necessarily *incomplete*, yielding a large number of words that are completely ambiguous. The high ambiguity rate inherent in our weakly supervised approach substantially complicates the POS tagging process. Consequently, our third goal of this paper is to propose two potentially performance-enhancing extensions to G&G's Bayesian POS tagging approach, which exploit morphology and techniques successfully used in supervised POS tagging.

The rest of the paper is organized as follows. Section 2 presents related work on unsupervised approaches to POS tagging. Section 3 gives an introduction to G&G's fully-Bayesian approach to unsupervised POS tagging. In Section 4, we describe our two extensions to G&G's approach. Section 5 presents experimental results on Bengali POS tagging, focusing on evaluating the effectiveness of our two extensions in improving G&G's approach. Finally, we conclude in Section 6.

[1]When evaluating an unsupervised POS tagger, researchers typically construct a *pseudo-perfect* POS lexicon by collecting the possible POS tags of a word directly from the corpus on which the tagger is to be evaluated.

2 Related Work

With the notable exception of Synder et al.'s (2008; 2009) recent work on unsupervised multilingual POS tagging, existing approaches to unsupervised POS tagging have been developed and tested primarily on English data. For instance, Merialdo (1994) uses maximum likelihood estimation to train a trigram HMM. Schütze (1995) and Clark (2000) apply syntactic clustering and dimensionality reduction in a knowledge-free setting to obtain meaningful clusters. Haghighi and Klein (2006) develop a prototype-driven approach, which requires just a few prototype examples for each POS tag and exploits these labeled words to constrain the labels of their distributionally similar words. Smith and Eisner (2005) train an unsupervised POS tagger using contrastive estimation, which seeks to move probability mass to a positive example e from its neighbors (i.e., negative examples are created by perturbing e). Wang and Schuurmans (2005) improve an unsupervised HMM-based tagger by constraining the learned structure to maintain appropriate marginal tag probabilities and using word similarities to smooth the lexical parameters.

As mentioned before, Goldwater and Griffiths (2007) have recently proposed an unsupervised fully-Bayesian POS tagging framework that operates by integrating over the possible parameter values instead of fixing a set of parameter values for unsupervised sequence learning. Importantly, this Bayesian approach facilitates the incorporation of sparse priors that result in a more practical distribution of tokens to lexical categories (Johnson, 2007). Similar to Goldwater and Griffiths (2007) and Johnson (2007), Toutanova and Johnson (2007) also use Bayesian inference for POS tagging. However, their work departs from existing Bayesian approaches to POS tagging in that they (1) introduce a new sparse prior on the distribution over tags for each word, (2) extend the Latent Dirichlet Allocation model, and (3) explicitly model ambiguity class. While their tagging model, like Goldwater and Griffiths's, assumes as input an incomplete POS lexicon and a large unlabeled corpus, they consider their approach "semi-supervised" simply because of the human knowledge involved in constructing the POS lexicon.

3 A Fully Bayesian Approach

3.1 Motivation

As mentioned in the introduction, the most common approach to unsupervised POS tagging is to train an HMM on an unannotated corpus using the Baum-Welch algorithm so that the likelihood of the corpus is maximized. To understand what the HMM parameters are, let us revisit how an HMM simultaneously generates an output sequence $\mathbf{w} = (w_0, w_1, ..., w_n)$ and the associated hidden state sequence $\mathbf{t} = (t_0, t_1, ..., t_n)$. In the context of POS tagging, each state of the HMM corresponds to a POS tag, the output sequence $\mathbf{w}$ is the given word sequence, and the hidden state sequence $\mathbf{t}$ is the associated POS tag sequence. To generate $\mathbf{w}$ and $\mathbf{t}$, the HMM begins by guessing a state t_0 and then emitting w_0 from t_0 according to a state-specific output distribution over word tokens. After that, we move to the next state t_1, the choice of which is based on t_0's transition distribution, and emit w_1 according to t_1's output distribution. This generation process repeats until the end of the word sequence is reached. In other words, the parameters of an HMM, θ, are composed of a set of state-specific (1) output distributions (over word tokens) and (2) transition distributions, both of which can be learned using the EM algorithm. Once learning is complete, we can use the resulting set of parameters to find the most likely hidden state sequence given a word sequence using the Viterbi algorithm.

Nevertheless, EM sometimes fails to find good parameter values.[2] The reason is that EM tries to assign roughly the same number of word tokens to each of the hidden states (Johnson, 2007). In practice, however, the distribution of word tokens to POS tags is highly skewed (i.e., some POS categories are more populated with tokens than others). This motivates a fully-Bayesian approach, which, rather than committing to a particular set of parameter values as in an EM-based approach, integrates over all possible values of θ and, most importantly, allows the use of priors to favor the learning of the skewed distributions, through the use of the term $P(\theta|\mathbf{w})$ in the following equation:

$$P(\mathbf{t}|\mathbf{w}) = \int P(\mathbf{t}|\mathbf{w}, \theta) P(\theta|\mathbf{w}) d\theta \qquad (1)$$

The question, then, is: which priors on θ would allow the acquisition of skewed distributions? To answer this question, recall that in POS tagging, θ is composed of a set of tag transition distributions and output distributions. Each such distribution is a multinomial (i.e., each trial produces exactly one of some finite number of possible outcomes). For a multinomial with K outcomes, a K-dimensional Dirichlet distribution, which is conjugate to the multinomial, is a natural choice of prior. For simplicity, we assume that a distribution in θ is drawn from a symmetric Dirichlet with a certain hyperparameter (see Teh et al. (2006) for details).

The value of a hyperparameter, α, affects the skewness of the resulting distribution, as it assigns different probabilities to different distributions. For instance, when $\alpha < 1$, higher probabilities are assigned to *sparse* multinomials (i.e., multinomials in which only a few entries are non-zero). Intuitively, the tag transition distributions and the output distributions in an HMM-based POS tagger are sparse multinomials. As a result, it is logical to choose a Dirichlet prior with $\alpha < 1$. By integrating over all possible parameter values, the probability that i-th outcome, y_i, takes the value k, given the previous $i-1$ outcomes $\mathbf{y}_{-i} = (y_1, y_2, ..., y_{i-1})$, is

$$\begin{aligned} P(k|\mathbf{y}_{-i}, \alpha) &= \int P(k|\theta) P(\theta|\mathbf{y}_{-i}, \alpha) d\theta && (2) \\ &= \frac{n_k + \alpha}{i - 1 + K\alpha} && (3) \end{aligned}$$

where n_k is the frequency of k in $\mathbf{y}_{-i}$. See MacKay and Peto (1995) for the derivation.

3.2 Model

Our baseline POS tagging model is a standard trigram HMM with tag transition distributions and output distributions, each of which is a sparse multinomial that is learned by applying a symmetric Dirichlet prior:

$$\begin{aligned} t_i \mid t_{i-1}, t_{i-2}, \tau^{(t_{i-1}, t_{i-2})} &\sim \text{Mult}(\tau^{(t_{i-1}, t_{i-2})}) \\ w_i \mid t_i, \omega^{(t_i)} &\sim \text{Mult}(\omega^{(t_i)}) \\ \tau^{(t_{i-1}, t_{i-2})} \mid \alpha &\sim \text{Dirichlet}(\alpha) \\ \omega^{(t_i)} \mid \beta &\sim \text{Dirichlet}(\beta) \end{aligned}$$

where w_i and t_i denote the i-th word and tag. With a tagset of size T (including a special tag used as sentence delimiter), each of the tag transition distributions has T components. For the output symbols, each of the $\omega^{(t_i)}$ has W_{t_i} components, where W_{t_i} denotes the number of word types that can be emitted from the state corresponding to t_i.

[2] When given good parameter initializations, however, EM can find good parameter values for an HMM-based POS tagger. See Goldberg et al. (2008) for details.

From the closed form in Equation 3, given previous outcomes, we can compute the tag transition and output probabilities of the model as follows:

$$P(t_i|\mathbf{t}_{-i},\alpha) = \frac{n_{(t_{i-2},t_{i-1},t_i)} + \alpha}{n_{(t_{i-2},t_{i-1})} + T\alpha} \quad (4)$$

$$P(w_i|t_i,\mathbf{t}_{-i},\mathbf{w}_{-i},\beta) = \frac{n_{(t_i,w_i)} + \beta}{n_{t_i} + W_{t_i}\beta} \quad (5)$$

where $n_{(t_{i-2},t_{i-1},t_i)}$ and $n_{(t_i,w_i)}$ are the frequencies of observing the tag trigram (t_{i-2}, t_{i-1}, t_i) and the tag-word pair (t_i, w_i), respectively. These counts are taken from the $i-1$ tags and words generated previously. The inference procedure described next exploits the property that trigrams (and outputs) are *exchangeable*; that is, the probability of a set of trigrams (and outputs) does not depend on the order in which it was generated.

3.3 Inference Procedure

We perform inference using Gibbs sampling (Geman and Geman, 1984), using the following posterior distribution to generate samples:

$$P(\mathbf{t}|\mathbf{w},\alpha,\beta) \propto P(\mathbf{w}|\mathbf{t},\beta)P(\mathbf{t}|\alpha)$$

Starting with a random assignment of a POS tag to each word (subject to the constraints in the POS lexicon), we resample each POS tag, t_i, according to the conditional distribution shown in Figure 2. Note that the current counts of other trigrams and outputs can be used as "previous" observations due to the property of exchangeability.

Following G&G, we use simulated annealing to find the MAP tag sequence. The temperature decreases by a factor of $\exp(\frac{\log(\frac{\theta_2}{\theta_1})}{N-1})$ after each iteration, where θ_1 is the initial temperature and θ_2 is the temperature after N sampling iterations.

4 Two Extensions

In this section, we present two extensions to G&G's fully-Bayesian framework to unsupervised POS tagging, namely, induced suffix emission and discriminative prediction.

4.1 Induced Suffix Emission

For morphologically-rich languages like Bengali, a lot of grammatical information (e.g., POS) is expressed via suffixes. In fact, several approaches to unsupervised POS induction for morphologically-rich languages have exploited the observation that some suffixes can only be associated with a small number of POS tags (e.g., Clark (2003), Dasgupta and Ng (2007)). To exploit suffixes in HMM-based POS tagging, one can (1) convert the word-based POS lexicon to a *suffix-based POS lexicon*, which lists the possible POS tags for each suffix; and then (2) have the HMM emit suffixes rather than words, subject to the constraints in the suffix-based POS lexicon. Such a suffix-based HMM, however, may suffer from over-generalization. To prevent over-generalization and at the same time exploit suffixes, we propose as our first extension to G&G's framework a hybrid approach to word/suffix emission: a word is emitted if it is present in the word-based POS lexicon; otherwise, its suffix is emitted. In other words, our approach imposes suffix-based constraints on the tagging of words that are unseen w.r.t. the word-based POS lexicon. Below we show how to induce the suffix of a word and create the suffix-based POS lexicon.

Inducing suffixes To induce suffixes, we rely on Keshava and Pitler's (2006) method. Assume that (1) V is a vocabulary (i.e., a set of distinct words) extracted from a large, unannotated corpus, (2) C_1 and C_2 are two character sequences, and (3) C_1C_2 is the concatenation of C_1 and C_2. If C_1C_2 and C_1 are found in V, we extract C_2 as a suffix.

However, this unsupervised suffix induction method is arguably overly simplistic and hence many of the induced affixes could be spurious. To identify suffixes that are likely to be correct, we employ a simple procedure: we (1) score each suffix by multiplying its *frequency* (i.e., the number of distinct words in V to which each suffix attaches) and its *length*[3], and (2) select only those whose score is above a certain threshold. In our experiments, we set this threshold to 50, and generate our vocabulary from five years of articles taken from the Bengali newspaper *Prothom Alo*. This enables us to induce 975 suffixes.

Constructing a suffix-based POS lexicon Next, we construct a suffix-based POS lexicon. For each word w in the original word-based POS lexicon, we (1) use the induced suffix list obtained in the previous step to identify the longest-matching suffix of w, and then (2) assign all the POS tags associated with w to this suffix.

Incorporating suffix-based output distributions Finally, we extend our trigram model by introduc-

[3]The dependence on frequency and length is motivated by the observation that less frequent and shorter affixes are more likely to be erroneous (see Goldsmith (2001)).

$$P(t_i|\mathbf{t}_{-i},\mathbf{w},\alpha,\beta) \propto \frac{n_{(t_i,w_i)}+\beta}{n_{t_i}+W_{t_i}\beta} \cdot \frac{n_{(t_{i-2},t_{i-1},t_i)}+\alpha}{n_{(t_{i-2},t_{i-1})}+T\alpha} \cdot \frac{n_{(t_{i-1},t_i,t_{i+1})}+I(t_{i-2}=t_{i-1}=t_i=t_{i+1})+\alpha}{n_{(t_{i-1},t_i)}+I(t_{i-2}=t_{i-1}=t_i)+T\alpha}$$
$$\cdot \frac{n_{(t_i,t_{i+1},t_{i+2})}+I(t_{i-2}=t_i=t_{i+2},t_{i-1}=t_{i+1})+I(t_{i-1}=t_i=t_{i+1}=t_{i+2})+\alpha}{n_{(t_i,t_{i+1})}+I(t_{i-2}=t_i,t_{i-1}=t_{i+1})+I(t_{i-1}=t_i=t_{i+1})+T\alpha}$$

Figure 2: The sampling distribution for t_i (taken directly from Goldwater and Griffiths (2007)). All n_x values are computed from the current values of all tags except for t_i. Here, $I(arg)$ is a function that returns 1 if arg is true and 0 otherwise, and $\mathbf{t}_{-i}$ refers to the current values of all tags except for t_i.

ing a state-specific probability distribution over induced suffixes. Specifically, if the current word is present in the word-based POS lexicon, or if we cannot find any suffix for the word using the induced suffix list, then we emit the word. Otherwise, we emit its suffix according to a suffix-based output distribution, which is drawn from a symmetric Dirichlet with hyperparameter γ:

$$\begin{aligned} s_i \mid t_i, \sigma^{(t_i)} &\sim \text{Mult}(\sigma^{(t_i)}) \\ \sigma^{(t_i)} \mid \gamma &\sim \text{Dirichlet}(\gamma) \end{aligned}$$

where s_i denotes the induced suffix of the i-th word. The distribution, $\sigma^{(t_i)}$, has S_{t_i} components, where S_{t_i} denotes the number of induced suffixes that can be emitted from the state corresponding to t_i. We compute the induced suffix emission probabilities of the model as follows:

$$P(s_i|t_i,\mathbf{t}_{-i},\mathbf{s}_{-i},\gamma) = \frac{n_{(t_i,s_i)}+\gamma}{n_{t_i}+S_{t_i}\gamma} \quad (6)$$

where $n_{(t_i,s_i)}$ is the frequency of observing the tag-suffix pair (t_i, s_i).

This extension requires that we slightly modify the inference procedure. Specifically, if the current word is unseen (w.r.t. the word-based POS lexicon) and has a suffix (according to the induced suffix list), then we sample from a distribution that is almost identical to the one shown in Figure 2, except that we replace the first fraction (i.e., the fraction involving the emission counts) with the one shown in Equation (6). Otherwise, we simply sample from the distribution in Figure 2.

4.2 Discriminative Prediction

As mentioned in the introduction, the (word-based) POS lexicons used in existing approaches to unsupervised POS tagging were created somewhat unrealistically by collecting the possible POS tags of a word directly from the corpus on which the tagger is to be evaluated. To make the lexicon formation process more realistic, we propose a *weakly supervised* approach to Bayesian POS tagging, in which we *automatically* create the word-based POS lexicon from a small set of POS-tagged sentences that is disjoint from the test data. Adopting a weakly supervised approach has an additional advantage: the presence of POS-tagged sentences makes it possible to exploit techniques developed for supervised POS tagging, which is the idea behind discriminative prediction, our second extension to G&G's framework.

Given a small set of POS-tagged sentences L, discriminative prediction uses the statistics collected from L to predict the POS of a word in a discriminative fashion whenever possible. More specifically, discriminative prediction relies on two simple ideas typically exploited by supervised POS tagging algorithms: (1) if the target word (i.e., the word whose POS tag is to be predicted) appears in L, we can label the word with its POS tag in L; and (2) if the target word does not appear in L but its context does, we can use its context to predict its POS tag. In bigram and trigram POS taggers, the context of a word is represented using the preceding one or two words. Nevertheless, since L is typically small in a weakly supervised setting, it is common for a target word not to satisfy any of the two conditions above. Hence, if it is not possible to predict a target word in a discriminative fashion (due to the limited size of L), we resort to the sampling equation in Figure 2.

To incorporate the above discriminative decision steps into G&G's fully-Bayesian framework for POS tagging, the algorithm estimates three types of probability distributions from L. First, to capture context, it computes (1) a distribution over the POS tags following a word bigram, (w_{i-2}, w_{i-1}), that appears in L [henceforth $D_1(w_{i-2}, w_{i-1})$] and (2) a distribution over the POS tags following a word unigram, w_{i-1}, that appears in L [henceforth $D_2(w_{i-1})$]. Then, to cap-

Algorithm 1 Algorithm for incorporating discriminative prediction

Input: w_i: current word
w_{i-1}: previous word
w_{i-2}: second previous word
L: a set of POS-tagged sentences
Output: Predicted tag, t_i
1: **if** $w_i \in L$ **then**
2: $t_i \leftarrow$ Tag drawn from the distribution of w_i's candidate tags
3: **else if** $(w_{i-2}, w_{i-1}) \in L$ **then**
4: $t_i \leftarrow$ Tag drawn from the distribution of the POS tags following the word bigram (w_{i-2}, w_{i-1})
5: **else if** $w_{i-1} \in L$ **then**
6: $t_i \leftarrow$ Tag drawn from the distribution of the POS tags following the word unigram w_{i-1}
7: **else**
8: $t_i \leftarrow$ Tag obtained using the sampling equation
9: **end if**

ture the fact that a word can have more than one POS tag, it also estimates a distribution over POS tags for each word w_i that appears in L [henceforth $D_3(w_i)$].

Implemented as a set of if-else clauses, the algorithm uses these three types of distributions to tag a target word, w_i, in a discriminative manner. First, it checks whether w_i appears in L (line 1). If so, it tags w_i according to $D_3(w_i)$. Otherwise, it attempts to label w_i based on its context. Specifically, if (w_{i-2}, w_{i-1}), the word bigram preceding w_i, appears in L (line 3), then w_i is tagged according to $D_1(w_{i-2}, w_{i-1})$. Otherwise, it backs off to a unigram distribution: if w_{i-1}, the word preceding w_i, appears in L (line 5), then w_i is tagged according to $D_2(w_{i-1})$. Finally, if it is not possible to tag the word discriminatively (i.e., if all the above cases fail), it resorts to the sampling equation (lines 7–8). We apply simulated annealing to all four cases in this iterative tagging procedure.

5 Evaluation

5.1 Experimental Setup

Corpus Our evaluation corpus is the one used in the shared task of the IJCNLP-08 Workshop on NER for South and South East Asian Languages.[4] Specifically, we use the portion of the Bengali dataset that is manually POS-tagged. IIIT Hyderabad's POS tagset[5], which consists of 26 tags specifically developed for Indian languages, has been used to annotate the data. The corpus is composed of a training set and a test set with approximately 50K and 30K tokens, respectively. Importantly, all our POS tagging results will be reported using only the test set; the training set will be used for lexicon construction, as we will see shortly.

Tagset We collapse the set of 26 POS tags into 15 tags. Specifically, while we retain the tags corresponding to the major POS categories, we merge some of the infrequent tags designed to capture Indian language specific structure (e.g., reduplication, echo words) into a category called OTHERS.

Hyperparameter settings Recall that our tagger consists of three types of distributions — tag transition distributions, word-based output distributions, and suffix-based output distributions — drawn from a symmetric Dirichlet with α, β, and γ as the underlying hyperparameters, respectively. We automatically determine the values of these hyperparameters by (1) randomly initializing them and (2) resampling their values by using a Metropolis-Hastings update (Gilks et al., 1996) at the end of each sampling iteration. Details of this update process can be found in G&G.

Inference Inference is performed by running a Gibbs sampler for 5000 iterations. The initial temperature is set to 2.0, which is gradually lowered to 0.08 over the iterations. Owing to the randomness involved in hyperparameter initialization, all reported results are averaged over three runs.

Lexicon construction methods To better understand the role of a POS lexicon in tagging performance, we evaluate each POS tagging model by employing lexicons constructed by three methods.

The first lexicon construction method, arguably the most unrealistic among the three, follows that of G&G: for each word, w, in the *test* set, we (1) collect from each occurrence of w in the training set *and* the test set its POS tag, and then (2) insert w and all the POS tags collected for w into the POS lexicon. This method is unrealistic because (1) in practice, a human needs to list all possible POS tags for each word in order to construct this lexicon, thus rendering the resulting tagger considerably less unsupervised than it appears; and (2) constructing the lexicon using the dataset on which the tagger is to be evaluated implies that there is no *unseen* word w.r.t. the lexicon, thus unrealistically simplifies the POS tagging task. To make the method more realistic, G&G also create a set of *relaxed* lexicons. Each of these lexicons includes the tags for only the words that appear at least d times in the test corpus, where d ranges

[4] The corpus is available from http://ltrc.iiit.ac.in/ner-ssea-08/index.cgi?topic=5.

[5] http://shiva.iiit.ac.in/SPSAL2007/iiit_tagset_guidelines.pdf

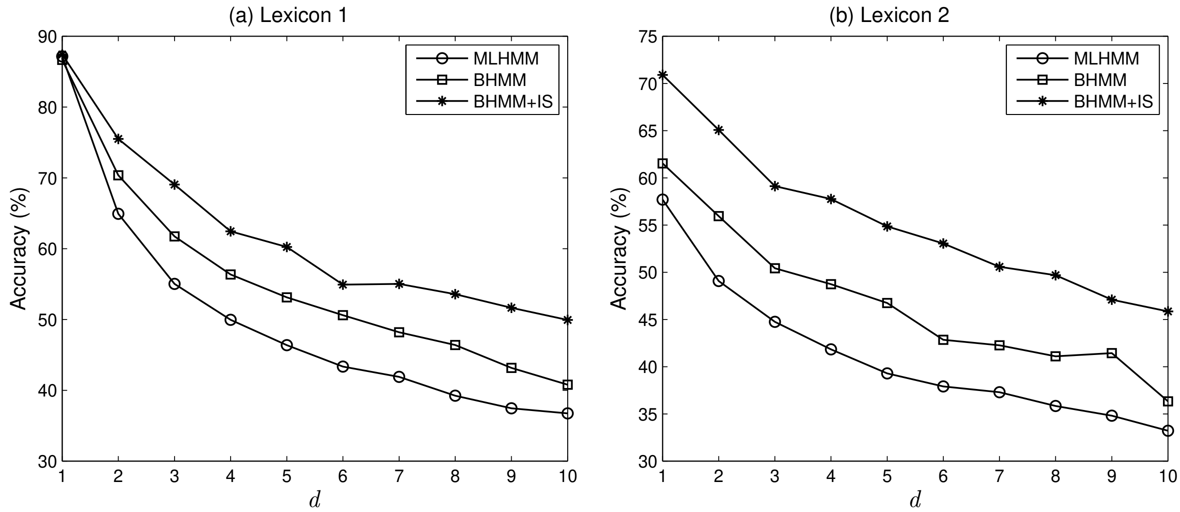

Figure 3: Accuracies of POS tagging models using (a) Lexicon 1 and (b) Lexicon 2

from 1 to 10 in our experiments. Any unseen (i.e., out-of-dictionary) word is ambiguous among the 15 possible tags. Not surprisingly, both ambiguity and the unseen word rate increase with d. For instance, the ambiguous token rate increases from 40.0% with 1.7 tags/token (d=1) to 77.7% with 8.1 tags/token (d=10). Similarly, the unseen word rate increases from 16% (d=2) to 46% (d=10). We will refer to this set of tag dictionaries as *Lexicon 1*.

The second method generates a set of relaxed lexicons, *Lexicon 2*, in essentially the same way as the first method, except that these lexicons include only the words that appear at least d times in the training data. Importantly, the words that appear solely in the test data are not included in any of these relaxed POS lexicons. This makes Lexicon 2 a bit more realistic than Lexicon 1 in terms of the way they are constructed. As a result, in comparison to Lexicon 1, Lexicon 2 has a considerably higher ambiguous token rate and unseen word rate: its ambiguous token rate ranges from 64.3% with 5.3 tags/token (d=1) to 80.5% with 8.6 tags/token (d=10), and its unseen word rate ranges from 25% (d=1) to 50% (d=10).

The third method, arguably the most realistic among the three, is motivated by our proposed weakly supervised approach. In this method, we (1) form ten different datasets from the (labeled) training data of sizes 5K words, 10K words, . . ., 50K words, and then (2) create one POS lexicon from each dataset L by listing, for each word w in L, all the tags associated with w in L. This set of tag dictionaries, which we will refer to as *Lexicon 3*, has an ambiguous token rate that ranges from 57.7% with 5.1 tags/token (50K) to 61.5% with 8.1 tags/token (5K), and an unseen word rate that ranges from 25% (50K) to 50% (5K).

5.2 Results and Discussion

5.2.1 Baseline Systems

We use as our first baseline system G&G's Bayesian POS tagging model, as our goal is to evaluate the effectiveness of our two extensions in improving their model. To further gauge the performance of G&G's model, we employ another baseline commonly used in POS tagging experiments, which is an unsupervised trigram HMM trained by running EM to convergence.

As mentioned previously, we evaluate each tagging model by employing the three POS lexicons described in the previous subsection. Figure 3(a) shows how the tagging accuracy varies with d when Lexicon 1 is used. Perhaps not surprisingly, the trigram HMM (MLHMM) and G&G's Bayesian model (BHMM) achieve almost identical accuracies when d=1 (i.e., the complete lexicon with a zero unseen word rate). As d increases, both ambiguity and the unseen word rate increase; as a result, the tagging accuracy decreases. Also, consistent with G&G's results, BHMM outperforms MLHMM by a large margin (4–7%).

Similar performance trends can be observed when Lexicon 2 is used (see Figure 3(b)). However, both baselines achieve comparatively lower tagging accuracies, as a result of the higher unseen word rate associated with Lexicon 2.

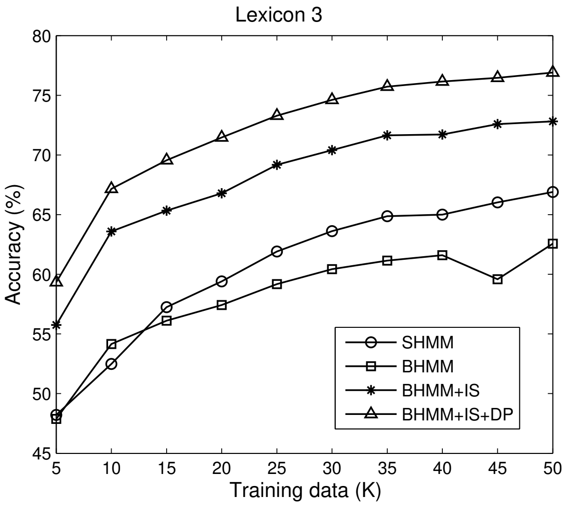

Figure 4: Accuracies of the POS tagging models using Lexicon 3

Results using Lexicon 3 are shown in Figure 4. Owing to the availability of POS-tagged sentences, we replace MLHMM with its *supervised* counterpart that is trained on the available labeled data, yielding the SHMM baseline. The accuracies of SHMM range from 48% to 67%, outperforming BHMM as the amount of labeled data increases.

5.2.2 Adding Induced Suffix Emission

Next, we augment BHMM with our first extension, induced suffix emission, yielding BHMM+IS. For Lexicon 1, BHMM+IS achieves the same accuracy as the two baselines when d=1. The reason is simple: as all the test words are in the POS lexicon, the tagger never emits an induced suffix. More importantly, BHMM+IS beats BHMM and MLHMM by 4–9% and 10–14%, respectively. Similar trends are observed for Lexicon 2, where BHMM+IS outperforms BHMM and MLHMM by a larger margin of 5–10% and 12–16%, respectively. For Lexicon 3, BHMM+IS outperforms SHMM, the stronger baseline, by 6–11%. Overall, these results suggest that induced suffix emission is a strong performance-enhancing extension to G&G's approach.

5.2.3 Adding Discriminative Prediction

Finally, we augment BHMM+IS with discriminative prediction, yielding BHMM+IS+DP. Since this extension requires labeled data, it can only be applied in combination with Lexicon 3. As seen in Figure 4, BHMM+IS+DP outperforms SHMM by 10–14%. Its discriminative nature proves to be strong as it even beats BHMM+IS by 3–4%.

Predicted Tag	Correct Tag	% of Error
NN	NNP	8.4
NN	JJ	6.9
VM	VAUX	5.9

Table 1: Most frequent POS tagging errors for BHMM+IS+DP on the 50K-word training set

5.2.4 Error Analysis

Table 1 lists the most common types of errors made by the best-performing tagging model, BHMM+IS+DP (50K-word labeled data). As we can see, common nouns and proper nouns (row 1) are difficult to distinguish, due in part to the case insensitivity of Bengali. Also, it is difficult to distinguish Bengali common nouns and adjectives (row 2), as they are distributionally similar to each other. The confusion between main verbs [VM] and auxiliary verbs [VAUX] (row 3) arises from the fact that certain Bengali verbs can serve as both a main verb and an auxiliary verb, depending on the role the verb plays in the verb sequence.

6 Conclusions

While Goldwater and Griffiths's fully-Bayesian approach and the traditional maximum-likelihood parameter-based approach to unsupervised POS tagging have offered promising results for English, we argued in this paper that such results were obtained under the unrealistic assumption that a perfect POS lexicon is available, which renders these taggers less unsupervised than they appear. As a result, we investigated a weakly supervised fully-Bayesian approach to POS tagging, which relaxes the unrealistic assumption by automatically acquiring the lexicon from a small amount of POS-tagged data. Since such relaxation comes at the expense of a drop in tagging accuracy, we proposed two performance-enhancing extensions to the Bayesian framework, namely, induced suffix emission and discriminative prediction, which effectively exploit morphology and techniques from supervised POS tagging, respectively.

Acknowledgments

We thank the three anonymous reviewers and Sajib Dasgupta for their comments. We also thank CRBLP, BRAC University, Bangladesh, for providing us with Bengali resources and Taufiq Hasan Al Banna for his MATLAB code. This work was supported in part by NSF Grant IIS-0812261.

References

Leonard E. Baum. 1972. An equality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes. *Inequalities*, 3:1–8.

Alexander Clark. 2000. Inducing syntactic categories by context distribution clustering. In *Proceedings of CoNLL: Short Papers*, pages 91–94.

Alexander Clark. 2003. Combining distributional and morphological information for part-of-speech induction. In *Proceedings of the EACL*, pages 59–66.

Sajib Dasgupta and Vincent Ng. 2007. Unsupervised part-of-speech acquisition for resource-scarce languages. In *Proceedings of EMNLP-CoNLL*, pages 218–227.

Arthur P. Dempster, Nan M. Laird, and Donald B. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39:1–38.

Stuart Geman and Donald Geman. 1984. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721–741.

Walter R. Gilks, Sylvia Richardson, and David J. Spiegelhalter (editors). 1996. *Markov Chain Monte Carlo in Practice*. Chapman & Hall, Suffolk.

Yoav Goldberg, Meni Adler, and Michael Elhadad. 2008. EM can find pretty good HMM POS-taggers (when given a good start). In *Proceedings of ACL-08:HLT*, pages 746–754.

John Goldsmith. 2001. Unsupervised learning of the morphology of a natural language. *Computational Linguistics*, 27(2):153–198.

Sharon Goldwater and Thomas L. Griffiths. 2007. A fully Bayesian approach to unsupervised part-of-speech tagging. In *Proceedings of the ACL*, pages 744–751.

Aria Haghighi and Dan Klein. 2006. Prototype-driven learning for sequence models. In *Proceedings of HLT-NAACL*, pages 320–327.

Mark Johnson. 2007. Why doesn't EM find good HMM POS-taggers? In *Proceedings of EMNLP-CoNLL*, pages 296–305.

Samarth Keshava and Emily Pitler. 2006. A simpler, intuitive approach to morpheme induction. In *PASCAL Challenge Workshop on Unsupervised Segmentation of Words into Morphemes*.

David J. C. MacKay and Linda C. Bauman Peto. 1995. A hierarchical Dirichlet language model. *Natural Language Engineering*, 1:289–307.

Bernard Merialdo. 1994. Tagging English text with a probabilistic model. *Computational Linguistics*, 20(2):155–172.

Hinrich Schütze. 1995. Distributional part-of-speech tagging. In *Proceedings of EACL*, pages 141–148.

Noah A. Smith and Jason Eisner. 2005. Contrastive estimation: Training log-linear models on unlabeled data. In *Proceedings of the ACL*, pages 354–362.

Benjamin Snyder, Tahira Naseem, Jacob Eisenstein, and Regina Barzilay. 2008. Unsupervised multilingual learning for POS tagging. In *Proceedings of EMNLP*, pages 1041–1050.

Benjamin Snyder, Tahira Naseem, Jacob Eisenstein, and Regina Barzilay. 2009. Adding more languages improves unsupervised multilingual tagging. In *Proceedings of NAACL-HLT*.

Yee Whye Teh, Michael Jordan, Matthew Beal, and David Blei. 2006. Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, 101(476):1527–1554.

Kristina Toutanova and Mark Johnson. 2007. A Bayesian LDA-based model for semi-supervised part-of-speech tagging. In *Proceedings of NIPS*.

Qin Iris Wang and Dale Schuurmans. 2005. Improved estimation for unsupervised part-of-speech tagging. In *Proceedings of the 2005 IEEE International Conference on Natural Language Processing and Knowledge Engineering (IEEE NLP-KE)*, pages 219–224.