Simple Type-Level Unsupervised POS Tagging

Yoong Keok Lee **Aria Haghighi** **Regina Barzilay**
Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology
{yklee, aria42, regina}@csail.mit.edu

Abstract

Part-of-speech (POS) tag distributions are known to exhibit *sparsity* — a word is likely to take a single predominant tag in a corpus. Recent research has demonstrated that incorporating this sparsity constraint improves tagging accuracy. However, in existing systems, this expansion come with a steep increase in model complexity. This paper proposes a simple and effective tagging method that directly models tag sparsity and other distributional properties of valid POS tag assignments. In addition, this formulation results in a dramatic reduction in the number of model parameters thereby, enabling unusually rapid training. Our experiments consistently demonstrate that this model architecture yields substantial performance gains over more complex tagging counterparts. On several languages, we report performance exceeding that of more complex state-of-the art systems.[1]

1 Introduction

Since the early days of statistical NLP, researchers have observed that a part-of-speech tag distribution exhibits "one tag per discourse" *sparsity* — words are likely to select a single predominant tag in a corpus, even when several tags are possible. Simply assigning to each word its most frequent associated tag in a corpus achieves 94.6% accuracy on the WSJ portion of the Penn Treebank. This distributional sparsity of syntactic tags is not unique to English — similar results have been observed across multiple languages. Clearly, explicitly modeling such a powerful constraint on tagging assignment has a potential to significantly improve the accuracy of an unsupervised part-of-speech tagger learned without a tagging dictionary.

In practice, this sparsity constraint is difficult to incorporate in a traditional POS induction system (Mérialdo, 1994; Johnson, 2007; Gao and Johnson, 2008; Graça et al., 2009; Berg-Kirkpatrick et al., 2010). These sequence models-based approaches commonly treat token-level tag assignment as the primary latent variable. By design, they readily capture regularities at the *token-level*. However, these approaches are ill-equipped to directly represent *type-based constraints* such as sparsity. Previous work has attempted to incorporate such constraints into token-level models via heavy-handed modifications to inference procedure and objective function (e.g., posterior regularization and ILP decoding) (Graça et al., 2009; Ravi and Knight, 2009). In most cases, however, these expansions come with a steep increase in model complexity, with respect to training procedure and inference time.

In this work, we take a more direct approach and treat a word type and its allowed POS tags as a primary element of the model. The model starts by generating a tag assignment for each word type in a vocabulary, assuming one tag per word. Then, token-level HMM emission parameters are drawn conditioned on these assignments such that each word is only allowed probability mass on a single assigned tag. In this way we restrict the parameterization of a

[1]The source code for the work presented in this paper is available at http://groups.csail.mit.edu/rbg/code/typetagging/.

Language	Original case
English	94.6
Danish	96.3
Dutch	96.6
German	95.5
Spanish	95.4
Swedish	93.3
Portuguese	95.6

Table 1: Upper bound on tagging accuracy assuming each word type is assigned to majority POS tag. Across all languages, high performance can be attained by selecting a single tag per word type.

token-level HMM to reflect lexicon sparsity. This model admits a simple Gibbs sampling algorithm where the number of latent variables is proportional to the number of word types, rather than the size of a corpus as for a standard HMM sampler (Johnson, 2007).

There are two key benefits of this model architecture. First, it directly encodes linguistic intuitions about POS tag assignments: the model structure reflects the one-tag-per-word property, and a type-level tag prior captures the skew on tag assignments (e.g., there are fewer unique determiners than unique nouns). Second, the reduced number of hidden variables and parameters dramatically speeds up learning and inference.

We evaluate our model on seven languages exhibiting substantial syntactic variation. On several languages, we report performance exceeding that of state-of-the art systems. Our analysis identifies three key factors driving our performance gain: 1) selecting a model structure which directly encodes tag sparsity, 2) a type-level prior on tag assignments, and 3) a straightforward naïve-Bayes approach to incorporate features. The observed performance gains, coupled with the simplicity of model implementation, makes it a compelling alternative to existing more complex counterparts.

2 Related Work

Recent work has made significant progress on unsupervised POS tagging (Mérialdo, 1994; Smith and Eisner, 2005; Haghighi and Klein, 2006; Johnson, 2007; Goldwater and Griffiths, 2007; Gao and Johnson, 2008; Ravi and Knight, 2009). Our work is closely related to recent approaches that incorporate the sparsity constraint into the POS induction process. This line of work has been motivated by empirical findings that the standard EM-learned unsupervised HMM does not exhibit sufficient word tag sparsity.

The extent to which this constraint is enforced varies greatly across existing methods. On one end of the spectrum are clustering approaches that assign a single POS tag to each word type (Schutze, 1995; Lamar et al., 2010). These clusters are computed using an SVD variant without relying on transitional structure. While our method also enforces a singe tag per word constraint, it leverages the transition distribution encoded in an HMM, thereby benefiting from a richer representation of context.

Other approaches encode sparsity as a soft constraint. For instance, by altering the emission distribution parameters, Johnson (2007) encourages the model to put most of the probability mass on few tags. This design does not guarantee "structural zeros," but biases towards sparsity. A more forceful approach for encoding sparsity is posterior regularization, which constrains the posterior to have a small number of expected tag assignments (Graça et al., 2009). This approach makes the training objective more complex by adding linear constraints proportional to the number of word types, which is rather prohibitive. A more rigid mechanism for modeling sparsity is proposed by Ravi and Knight (2009), who minimize the size of tagging grammar as measured by the number of transition types. The use of ILP in learning the desired grammar significantly increases the computational complexity of this method.

In contrast to these approaches, our method directly incorporates these constraints into the structure of the model. This design leads to a significant reduction in the computational complexity of training and inference.

Another thread of relevant research has explored the use of features in unsupervised POS induction (Smith and Eisner, 2005; Berg-Kirkpatrick et al., 2010; Hasan and Ng, 2009). These methods demonstrated the benefits of incorporating linguistic features using a log-linear parameterization, but requires elaborate machinery for training. In our

work, we demonstrate that using a simple naïve-Bayes approach also yields substantial performance gains, without the associated training complexity.

3 Generative Story

We consider the unsupervised POS induction problem without the use of a tagging dictionary. A graphical depiction of our model as well as a summary of random variables and parameters can be found in Figure 1. As is standard, we use a fixed constant K for the number of tagging states.

Model Overview The model starts by generating a tag assignment $\boldsymbol{T}$ for each word type in a vocabulary, assuming one tag per word. Conditioned on $\boldsymbol{T}$, features of word types $\boldsymbol{W}$ are drawn. We refer to $(\boldsymbol{T}, \boldsymbol{W})$ as the lexicon of a language and ψ for the parameters for their generation; ψ depends on a single hyperparameter β.

Once the lexicon has been drawn, the model proceeds similarly to the standard token-level HMM: Emission parameters θ are generated conditioned on tag assignments $\boldsymbol{T}$. We also draw transition parameters ϕ. Both parameters depend on a single hyperparameter α. Once HMM parameters (θ, ϕ) are drawn, a token-level tag and word sequence, (t, w), is generated in the standard HMM fashion: a tag sequence t is generated from ϕ. The corresponding token words w are drawn conditioned on t and θ.[2] Our full generative model is given by:

$$
\begin{aligned}
&P(\boldsymbol{T}, \boldsymbol{W}, \theta, \psi, \phi, \boldsymbol{t}, \boldsymbol{w} | \alpha, \beta) = \\
&\quad P(\boldsymbol{T}, \boldsymbol{W}, \psi | \beta) && \text{[Lexicon]} \\
&\quad P(\phi, \theta | \boldsymbol{T}, \alpha, \beta) && \text{[Parameter]} \\
&\quad P(\boldsymbol{w}, \boldsymbol{t} | \phi, \theta) && \text{[Token]}
\end{aligned}
$$

We refer to the components on the right hand side as the lexicon, parameter, and token component respectively. Since the parameter and token components will remain fixed throughout experiments, we briefly describe each.

Parameter Component As in the standard Bayesian HMM (Goldwater and Griffiths, 2007), all distributions are independently drawn from symmetric Dirichlet distributions:

$$P(\phi, \theta | \boldsymbol{T}, \alpha, \beta) = \prod_{t=1}^{K} \left(P(\phi_t | \alpha) P(\theta_t | \boldsymbol{T}, \alpha) \right)$$

The transition distribution ϕ_t for each tag t is drawn according to $\text{DIRICHLET}(\alpha, K)$, where α is the shared transition and emission distribution hyperparameter. In total there are $O(K^2)$ parameters associated with the transition parameters.

In contrast to the Bayesian HMM, θ_t is not drawn from a distribution which has support for each of the n word types. Instead, we condition on the type-level tag assignments $\boldsymbol{T}$. Specifically, let $S_t = \{i | T_i = t\}$ denote the indices of the word types which have been assigned tag t according to the tag assignments $\boldsymbol{T}$. Then θ_t is drawn from $\text{DIRICHLET}(\alpha, S_t)$, a symmetric Dirichlet which only places mass on word types indicated by S_t. This ensures that each word will only be assigned a single tag at inference time (see Section 4).

Note that while the standard HMM, has $O(Kn)$ emission parameters, our model has $O(n)$ effective parameters.[3]

Token Component Once HMM parameters (ϕ, θ) have been drawn, the HMM generates a token-level corpus $\boldsymbol{w}$ in the standard way:

$$P(\boldsymbol{w}, \boldsymbol{t} | \phi, \theta) = \prod_{(w,t) \in (\boldsymbol{w}, \boldsymbol{t})} \left(\prod_j P(t_j | \phi_{t_{j-1}}) P(w_j | t_j, \theta_{t_j}) \right)$$

Note that in our model, conditioned on $\boldsymbol{T}$, there is precisely one $\boldsymbol{t}$ which has non-zero probability for the token component, since for each word, exactly one θ_t has support.

3.1 Lexicon Component

We present several variations for the lexical component $P(\boldsymbol{T}, \boldsymbol{W} | \psi)$, each adding more complex parameterizations.

Uniform Tag Prior (1TW) Our initial lexicon component will be uniform over possible tag assignments as well as word types. Its only purpose is

[2] Note that t and w denote tag and word sequences respectively, rather than individual tokens or tags.

[3] This follows since each θ_t has $S_t - 1$ parameters and $\sum_t S_t = n$.

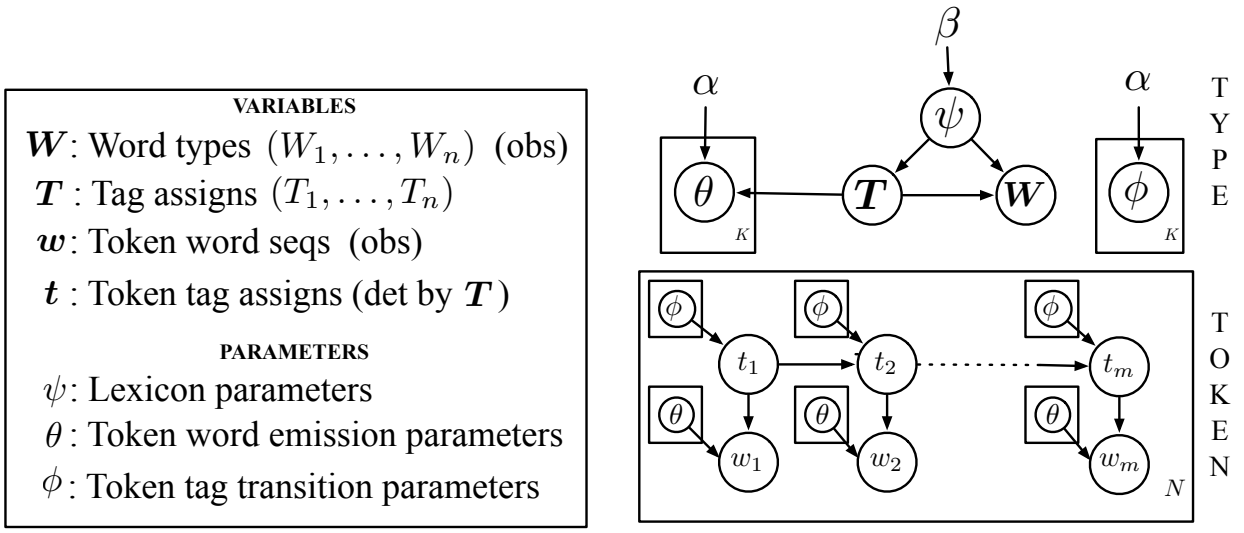

Figure 1: Graphical depiction of our model and summary of latent variables and parameters. The type-level tag assignments $\boldsymbol{T}$ generate features associated with word types $\boldsymbol{W}$. The tag assignments constrain the HMM emission parameters θ. The tokens $\boldsymbol{w}$ are generated by token-level tags $\boldsymbol{t}$ from an HMM parameterized by the lexicon structure. The hyperparameters α and β represent the concentration parameters of the token- and type-level components of the model respectively. They are set to fixed constants.

to explore how well we can induce POS tags using only the one-tag-per-word constraint. Specifically, the lexicon is generated as:

$$\begin{aligned} P(\boldsymbol{T}, \boldsymbol{W}|\psi) =& P(\boldsymbol{T})P(\boldsymbol{W}|\boldsymbol{T}) \\ =& \prod_{i=1}^{n} P(T_i)P(W_i|T_i) = \left(\frac{1}{Kn}\right)^n \end{aligned}$$

This model is equivalent to the standard HMM except that it enforces the one-word-per-tag constraint.

Learned Tag Prior (PRIOR) We next assume there exists a single prior distribution ψ over tag assignments drawn from $\textsc{Dirichlet}(\beta, K)$. This alters generation of $\boldsymbol{T}$ as follows:

$$P(\boldsymbol{T}|\psi) = \prod_{i=1}^{n} P(T_i|\psi)$$

Note that this distribution captures the frequency of a tag across word types, as opposed to tokens. The $P(T|\psi)$ distribution, in English for instance, should have very low mass for the DT (determiner) tag, since determiners are a very small portion of the vocabulary. In contrast, NNP (proper nouns) form a large portion of vocabulary. Note that these observations are not modeled by the standard HMM, which instead can model token-level frequency.

Word Type Features (FEATS): Past unsupervised POS work have derived benefits from features on word types, such as suffix and capitalization features (Hasan and Ng, 2009; Berg-Kirkpatrick et al., 2010). Past work however, has typically associated these features with token occurrences, typically in an HMM. In our model, we associate these features at the type-level in the lexicon. Here, we consider suffix features, capitalization features, punctuation, and digit features. While possible to utilize the feature-based log-linear approach described in Berg-Kirkpatrick et al. (2010), we adopt a simpler naïve Bayes strategy, where all features are emitted independently. Specifically, we assume each word type W consists of feature-value pairs (f, v). For each feature type f and tag t, a multinomial ψ_{tf} is drawn from a symmetric Dirichlet distribution with concentration parameter β. The $P(\boldsymbol{W}|\boldsymbol{T}, \psi)$ term in the lexicon component now decomposes as:

$$\begin{aligned} P(\boldsymbol{W}|\boldsymbol{T}, \psi) =& \prod_{i=1}^{n} P(W_i|T_i, \psi) \\ =& \prod_{i=1}^{n} \left(\prod_{(f,v)\in W_i} P(v|\psi_{T_i f}) \right) \end{aligned}$$

4 Learning and Inference

For inference, we are interested in the posterior probability over the latent variables in our model. During training, we treat as observed the language word types $\boldsymbol{W}$ as well as the token-level corpus $\boldsymbol{w}$. We utilize Gibbs sampling to approximate our collapsed model posterior:

$$P(\boldsymbol{T},\boldsymbol{t}|\boldsymbol{W},\boldsymbol{w},\alpha,\beta) \propto P(\boldsymbol{T},\boldsymbol{t},\boldsymbol{W},\boldsymbol{w}|\alpha,\beta)$$
$$= \int P(\boldsymbol{T},\boldsymbol{t},\boldsymbol{W},\boldsymbol{w},\psi,\theta,\phi,\boldsymbol{w}|\alpha,\beta)d\psi d\theta d\phi$$

Note that given tag assignments $\boldsymbol{T}$, there is only one setting of token-level tags $\boldsymbol{t}$ which has mass in the above posterior. Specifically, for the ith word type, the set of token-level tags associated with token occurrences of this word, denoted $\boldsymbol{t}^{(i)}$, must all take the value T_i to have non-zero mass. Thus in the context of Gibbs sampling, if we want to block sample T_i with $\boldsymbol{t}^{(i)}$, we only need sample values for T_i and consider this setting of $\boldsymbol{t}^{(i)}$.

The equation for sampling a single type-level assignment T_i is given by,

$$P(T_i,\boldsymbol{t}^{(i)}|\boldsymbol{T}_{-i},\boldsymbol{W},\boldsymbol{t}^{(-i)},\boldsymbol{w},\alpha,\beta) =$$
$$P(T_i|\boldsymbol{W},\boldsymbol{T}_{-i},\beta)P(\boldsymbol{t}^{(i)}|T_i,\boldsymbol{t}^{(-i)},\boldsymbol{w},\alpha)$$

where $\boldsymbol{T}_{-i}$ denotes all type-level tag assignment except T_i and $\boldsymbol{t}^{(-i)}$ denotes all token-level tags except $\boldsymbol{t}^{(i)}$. The terms on the right-hand-side denote the type-level and token-level probability terms respectively. The type-level posterior term can be computed according to,

$$P(T_i|\boldsymbol{W},\boldsymbol{T}_{-i},\beta) \propto$$
$$P(T_i|\boldsymbol{T}_{-i},\beta) \prod_{(f,v)\in W_i} P(v|T_i,f,\boldsymbol{W}_{-i},\boldsymbol{T}_{-i},\beta)$$

All of the probabilities on the right-hand-side are Dirichlet, distributions which can be computed analytically given counts.

The token-level term is similar to the standard HMM sampling equations found in Johnson (2007). The relevant variables are the set of token-level tags that appear before and after each instance of the ith word type; we denote these context pairs with the set $\{(t^b,t^a)\}$ and they are contained in $\boldsymbol{t}^{(-i)}$. We use w to represent the ith word type emitted by the HMM:

$$P(\boldsymbol{t}^{(i)}|T_i,\boldsymbol{t}^{(-i)},\boldsymbol{w},\alpha) \propto$$
$$\prod_{(t^b,t^a)} P(w|T_i,\boldsymbol{t}^{(-i)},\boldsymbol{w}^{(-i)},\alpha)$$
$$P(T_i|t^b,\boldsymbol{t}^{(-i)},\alpha)P(t^a|T_i,\boldsymbol{t}^{(-i)},\alpha)$$

All terms are Dirichlet distributions and parameters can be analytically computed from counts in $\boldsymbol{t}^{(-i)}$ and $\boldsymbol{w}^{(-i)}$ (Johnson, 2007).

Note that each round of sampling T_i variables takes time proportional to the size of the corpus, as with the standard token-level HMM. A crucial difference is that the number of parameters is greatly reduced as is the number of variables that are sampled during each iteration. In contrast to results reported in Johnson (2007), we found that the performance of our Gibbs sampler on the basic 1TW model stabilized very quickly after about 10 full iterations of sampling (see Figure 2 for a depiction).

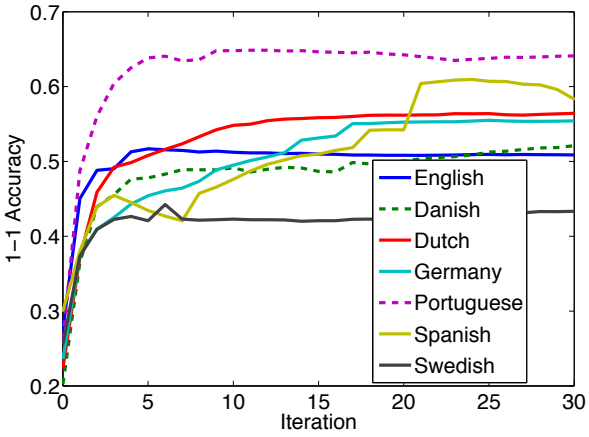

Figure 2: Graph of the one-to-one accuracy of our full model (+FEATS) under the best hyperparameter setting by iteration (see Section 5). Performance typically stabilizes across languages after only a few number of iterations.

5 Experiments

We evaluate our approach on seven languages: English, Danish, Dutch, German, Portuguese, Spanish, and Swedish. On each language we investigate the contribution of each component of our model. For all languages we do not make use of a tagging dictionary.

Model	Hyper-	English		Danish		Dutch		German		Portuguese		Spanish		Swedish	
	param.	1-1	m-1	1-1	m-1	1-1	m-1	1-1	m-1	1-1	m-1	1-1	m-1	1-1	m-1
1TW	best	45.2	62.6	37.2	56.2	47.4	53.7	44.2	62.2	49.0	68.4	34.3	54.4	36.0	55.3
	median	45.1	61.7	32.1	53.8	43.9	61.0	39.3	68.4	48.5	68.1	33.6	54.3	34.9	50.2
+PRIOR	best	47.9	65.5	42.3	58.3	51.4	65.9	50.7	62.2	56.2	70.7	42.8	54.8	38.9	58.0
	median	46.5	64.7	40.0	57.3	48.3	60.7	41.7	68.3	52.0	70.9	37.1	55.8	36.8	57.3
+FEATS	best	50.9	66.4	52.1	61.2	56.4	69.0	55.4	70.4	64.1	74.5	58.3	68.9	43.3	61.7
	median	47.8	66.4	43.2	60.7	51.5	67.3	46.2	61.7	56.5	70.1	50.0	57.2	38.5	60.6

Table 3: Multi-lingual Results: We report token-level one-to-one and many-to-one accuracy on a variety of languages under several experimental settings (Section 5). For each language and setting, we report one-to-one (1-1) and many-to-one (m-1) accuracies. For each cell, the first row corresponds to the result using the best hyperparameter choice, where best is defined by the 1-1 metric. The second row represents the performance of the median hyperparameter setting. Model components cascade, so the row corresponding to +FEATS also includes the PRIOR component (see Section 3).

Language	# Tokens	# Word Types	# Tags
English	1173766	49206	45
Danish	94386	18356	25
Dutch	203568	28393	12
German	699605	72325	54
Portuguese	206678	28931	22
Spanish	89334	16458	47
Swedish	191467	20057	41

Table 2: Statistics for various corpora utilized in experiments. See Section 5. The English data comes from the WSJ portion of the Penn Treebank and the other languages from the training set of the CoNLL-X multilingual dependency parsing shared task.

5.1 Data Sets

Following the set-up of Johnson (2007), we use the whole of the Penn Treebank corpus for training and evaluation on English. For other languages, we use the CoNLL-X multilingual dependency parsing shared task corpora (Buchholz and Marsi, 2006) which include gold POS tags (used for evaluation). We train and test on the CoNLL-X training set. Statistics for all data sets are shown in Table 2.

5.2 Setup

Models To assess the marginal utility of each component of the model (see Section 3), we incrementally increase its sophistication. Specifically, we evaluate three variants: The first model (1TW) only encodes the one tag per word constraint and is uniform over type-level tag assignments. The second model (+PRIOR) utilizes the independent prior over type-level tag assignments $P(\boldsymbol{T}|\psi)$. The final model (+FEATS) utilizes the tag prior as well as features (e.g., suffixes and orthographic features), discussed in Section 3, for the $P(\boldsymbol{W}|\boldsymbol{T},\psi)$ component.

Hyperparameters Our model has two Dirichlet concentration hyperparameters: α is the shared hyperparameter for the token-level HMM emission and transition distributions. β is the shared hyperparameter for the tag assignment prior and word feature multinomials. We experiment with four values for each hyperparameter resulting in 16 (α, β) combinations:

α	β
$0.001, 0.01, 0.1, 1.0$	$0.01, 0.1, 1.0, 10$

Iterations In each run, we performed 30 iterations of Gibbs sampling for the type assignment variables $\boldsymbol{W}$.[4] We use the final sample for evaluation.

Evaluation Metrics We report three metrics to evaluate tagging performance. As is standard, we report the *greedy one-to-one* (Haghighi and Klein, 2006) and the *many-to-one* token-level accuracy obtained from mapping model states to gold POS tags. We also report word type level accuracy, the fraction of word types assigned their majority tag (where the mapping between model state and tag is determined by greedy one-to-one mapping discussed above).[5]

For each language, we aggregate results in the following way: First, for each hyperparameter setting,

[4] Typically, the performance stabilizes after only 10 iterations.

[5] We choose these two metrics over the Variation Information measure due to the deficiencies discussed in Gao and Johnson (2008).

we perform five runs with different random initialization of sampling state. Hyperparameter settings are sorted according to the median one-to-one metric over runs. We report results for the best and median hyperparameter settings obtained in this way. Specifically, for both settings we report results on the median run for each setting.

Tag set As is standard, for all experiments, we set the number of latent model tag states to the size of the annotated tag set. The original tag set for the CoNLL-X Dutch data set consists of compounded tags that are used to tag multi-word units (MWUs) resulting in a tag set of over 300 tags. We tokenize MWUs and their POS tags; this reduces the tag set size to 12. See Table 2 for the tag set size of other languages. With the exception of the Dutch data set, no other processing is performed on the annotated tags.

6 Results and Analysis

We report token- and type-level accuracy in Table 3 and 6 for all languages and system settings. Our analysis and comparison focuses primarily on the one-to-one accuracy since it is a stricter metric than many-to-one accuracy, but also report many-to-one for completeness.

Comparison with state-of-the-art taggers For comparison we consider two unsupervised taggers: the HMM with log-linear features of Berg-Kirkpatrick et al. (2010) and the posterior regularization HMM of Graça et al. (2009). The system of Berg-Kirkpatrick et al. (2010) reports the best unsupervised results for English. We consider two variants of Berg-Kirkpatrick et al. (2010)'s richest model: optimized via either EM or LBFGS, as their relative performance depends on the language. Our model outperforms theirs on four out of five languages on the best hyperparameter setting and three out of five on the median setting, yielding an average absolute difference across languages of 12.9% and 3.9% for best and median settings respectively compared to their best EM or LBFGS performance. While Berg-Kirkpatrick et al. (2010) consistently outperforms ours on English, we obtain substantial gains across other languages. For instance, on Spanish, the absolute gap on median performance is 10%.

	Top 5	Bottom 5
Gold	NNP NN JJ CD NNS	RBS PDT # ” ,
1TW	**CD** WRB **NNS** VBN **NN**	PRP$ WDT : MD .
+PRIOR	**CD JJ NNS** WP$ **NN**	-RRB- **,** $ ” .
+FEATS	**JJ NNS CD NNP** UH	**,** PRP$ **#** . “

Table 5: Type-level English POS Tag Ranking: We list the top 5 and bottom 5 POS tags in the lexicon and the predictions of our models under the best hyperparameter setting.

Our second point of comparison is with Graça et al. (2009), who also incorporate a sparsity constraint, but does via altering the model objective using posterior regularization. We can only compare with Graça et al. (2009) on Portuguese (Graça et al. (2009) also report results on English, but on the reduced 17 tag set, which is not comparable to ours). Their best model yields 44.5% one-to-one accuracy, compared to our best median 56.5% result. However, our full model takes advantage of word features not present in Graça et al. (2009). Even without features, but still using the tag prior, our median result is 52.0%, still significantly outperforming Graça et al. (2009).

Ablation Analysis We evaluate the impact of incorporating various linguistic features into our model in Table 3. A novel element of our model is the ability to capture type-level tag frequencies. For this experiment, we compare our model with the uniform tag assignment prior (1TW) with the learned prior (+PRIOR). Across all languages, +PRIOR consistently outperforms 1TW, reducing error on average by 9.1% and 5.9% on best and median settings respectively. Similar behavior is observed when adding features. The difference between the featureless model (+PRIOR) and our full model (+FEATS) is 13.6% and 7.7% average error reduction on best and median settings respectively. Overall, the difference between our most basic model (1TW) and our full model (+FEATS) is 21.2% and 13.1% for the best and median settings respectively. One striking example is the error reduction for Spanish, which reduces error by 36.5% and 24.7% for the best and median settings respectively. We observe similar trends when using another measure – type-level accuracy (defined as the fraction of words correctly assigned their majority tag), according to which

Language	Metric	BK10 EM	BK10 LBFGS	G10	FEATS Best	FEATS Median
English	1-1	48.3	56.0	–	50.9	47.8
	m-1	68.1	75.5	–	66.4	66.4
Danish	1-1	42.3	42.6	–	52.1	43.2
	m-1	66.7	58.0	–	61.2	60.7
Dutch	1-1	53.7	55.1	–	56.4	51.5
	m-1	67.0	64.7	–	69.0	67.3
Portuguese	1-1	50.8	43.2	44.5	64.1	56.5
	m-1	75.3	74.8	69.2	74.5	70.1
Spanish	1-1	–	40.6	–	58.3	50.0
	m-1	–	73.2	–	68.9	57.2

Table 4: Comparison of our method (FEATS) to state-of-the-art methods. Feature-based HMM Model (Berg-Kirkpatrick et al., 2010): The KM model uses a variety of orthographic features and employs the EM or LBFGS optimization algorithm; Posterior regulariation model (Graça et al., 2009): The G10 model uses the posterior regularization approach to ensure tag sparsity constraint.

Language	1TW	+PRIOR	+FEATS
English	21.1	28.8	42.8
Danish	10.1	20.7	45.9
Dutch	23.8	32.3	44.3
German	12.8	35.2	60.6
Portuguese	18.4	29.6	61.5
Spanish	7.3	27.6	49.9
Swedish	8.9	14.2	33.9

Table 6: Type-level Results: Each cell report the type-level accuracy computed against the most frequent tag of each word type. The state-to-tag mapping is obtained from the best hyperparameter setting for 1-1 mapping shown in Table 3.

our full model yields 39.3% average error reduction across languages when compared to the basic configuration (1TW).

Table 5 provides insight into the behavior of different models in terms of the tagging lexicon they generate. The table shows that the lexicon tag frequency predicated by our full model are the closest to the gold standard.

7 Conclusion and Future Work

We have presented a method for unsupervised part-of-speech tagging that considers a word type and its allowed POS tags as a primary element of the model. This departure from the traditional token-based tagging approach allows us to explicitly capture type-level distributional properties of valid POS tag assignments as part of the model. The resulting model is compact, efficiently learnable and linguistically expressive. Our empirical results demonstrate that the type-based tagger rivals state-of-the-art tag-level taggers which employ more sophisticated learning mechanisms to exploit similar constraints.

In this paper, we make a simplifying assumption of one-tag-per-word. This assumption, however, is not inherent to type-based tagging models. A promising direction for future work is to explicitly model a distribution over tags for each word type. We hypothesize that modeling morphological information will greatly constrain the set of possible tags, thereby further refining the representation of the tag lexicon.

Acknowledgments

The authors acknowledge the support of the NSF (CAREER grant IIS-0448168, and grant IIS-0904684). We are especially grateful to Taylor Berg-Kirkpatrick for running additional experiments. We thank members of the MIT NLP group for their suggestions and comments. Any opinions, findings, conclusions, or recommendations expressed in this paper are those of the authors, and do not necessarily reflect the views of the funding organizations.

References

Taylor Berg-Kirkpatrick, Alexandre Bouchard-Côté, John DeNero, and Dan Klein. 2010. Painless un-

supervised learning with features. In *Proceedings of NAACL-HLT*, pages 582–590.

Sabine Buchholz and Erwin Marsi. 2006. Conll-x shared task on multilingual dependency parsing. In *In Proc. of CoNLL*, pages 149–164.

Jianfeng Gao and Mark Johnson. 2008. A comparison of bayesian estimators for unsupervised hidden markov model pos taggers. In *Proceedings of the EMNLP*, pages 344–352.

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.

João Graça, Kuzman Ganchev, Ben Taskar, and Fernando Pereira. 2009. Posterior vs. parameter sparsity in latent variable models. In *Proceeding of NIPS*, pages 664–672.

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

Kazi Saidul Hasan and Vincent Ng. 2009. Weakly supervised part-of-speech tagging for morphologically-rich, resource-scarce languages. In *Proceedings of EACL*, pages 363–371.

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

Michael Lamar, Yariv Maron, Marko Johnson, and Elie Bienstock. 2010. Svd Clustering for Unsupervised POS Tagging. In *Proceedings of ACL*, pages 215–219.

Bernard Mérialdo. 1994. Tagging english text with a probabilistic model. *Computational Linguistics*, 20(2):155–171.

Sujith Ravi and Kevin Knight. 2009. Minimized models for unsupervised part-of-speech tagging. In *Proceedings of ACL-IJCNLP*, pages 504–512.

Hinrich Schutze. 1995. Distributional part of speech tagging. In *Proceedings of the 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*.