Semi-Supervised Sequence Modeling with Syntactic Topic Models

Wei Li and Andrew McCallum
Computer Science Department
University of Massachusetts Amherst
{weili, mccallum}@cs.umass.edu

Abstract

Although there has been significant previous work on semi-supervised learning for classification, there has been relatively little in sequence modeling. This paper presents an approach that leverages recent work in manifold-learning on sequences to discover word clusters from language data, including both syntactic classes and semantic topics. From unlabeled data we form a smooth, low-dimensional feature space, where each word token is projected based on its underlying role as a function or content word. We then use this projection as additional input features to a linear-chain conditional random field trained on limited labeled training data. On standard part-of-speech tagging and Chinese word segmentation data sets we show as much as 14% error reduction due to the unlabeled data, and also statistically-significant improvements over a related semi-supervised sequence tagging method due to Miller et al.

1. Introduction

Semi-supervised learning has received significant attention as a method to reduce the need for expensive labeled data when unlabeled data is plentiful and easily obtained. It has been particularly attractive for application to many natural language processing (NLP) tasks, where the number of features and parameters are typically extremely large, and even many thousands of labeled examples only sparsely cover the parameter space.

In sequence modeling tasks (such as part-of-speech tagging, word segmentation and named entity recognition) obtaining labeled training data is more difficult than for classification tasks (such as document classification) because hand-labeling individual words and word boundaries is considerably more detail-oriented and difficult than assigning coarse-grained class labels.

Although highly desirable, semi-supervised learning for sequential data is not as widely studied as in other semi-supervised settings, such as document classification (Nigam *et al.* 2000) (Zhu, Ghahramani, & Lafferty 2003). Previous work in semi-supervised classification may, however, provide some insight into sequence-labeling tasks.

There are both generative and discriminative approaches to semi-supervised classification. With generative models, it is natural to include unlabeled data using expectation-maximization (Nigam *et al.* 2000). However, generative models have generally not provided as high accuracy as discriminative models. On the other hand, unlabeled data will cancel out of the objective function if directly used in traditional i.i.d. conditional-probability models. But if dependencies between labels of nearby instances are incorporated, unlabeled data will have an effect on training, as shown by Zhu, Ghahramani, & Lafferty (2003) and Li & McCallum (2004). These models are trained to encourage nearby data points to have the same class label, and they can obtain impressive accuracy using a very small amount of labeled data. However since they model pairwise similarities among data points and require joint inference over the whole dataset at test time, this approach is not efficient for large datasets.

Another way to achieve a similar effect is to force nearby data points to share features and their parameters, which will then of course express the same preferences for certain class labels. But when there is only limited labeled data, it is common to see two documents contain non-overlapping vocabularies even though they are essentially related to the same topics and class labels. So the model has difficulty capturing their closeness based solely on words. One way to measure the similarity between data points beyond surface text is to introduce underlying word clusters or topics to encode the similarities among words. For example, Buntine (2004) performs PCA on a large amount of unlabeled data and then uses the resulting word components as additional features in learning a SVM. These new features help to generalize from the observed words in labeled data and also provide more reliable probability estimation.

Miller, Guinness, & Zamanian (2004) take a similar approach to named entity recognition with discriminative models. They generate word clusters from unlabeled corpus using Peter Brown's method (Brown *et al.* 1992), which measures word similarities based on their distributions over the following words and performs a bottom-up agglomerative clustering. It can be viewed as a hard-clustering version of the information bottleneck method to maximize the mutual information between adjacent word cluster (Tishby, Pereira, & Bialek 1999). The clustering results in a binary tree, where the root node represents a cluster containing the

whole vocabulary, each internal node corresponds to a cluster that contains all the words that it dominates and each leaf node corresponds to a single word in the vocabulary. Note that this structure does not distinguish between different meanings of one word. For example, the word *order* would always belong to the same set of clusters whether it is used as a verb or as a noun. However, words with multiple possible POS tags or senses often cause ambiguity and are more difficult to label correctly in sequence labeling tasks. Solving this problem is crucial to improving the manifolds and thus accuracy. So we seek a clustering method that not only allows one word to probabilistically belong to multiple clusters, but also can determine the most likely cluster for each word-occurrence based on its own context.

In this paper, we present a method of semi-supervised learning for sequence-labeling tasks using discriminative models. Similarly to the work by Miller, Guinness, & Zamanian (2004), we create word clusters from unlabeled data and make them available in supervised training, which does not require more inference work than needed at test time. On the other hand, we want to capture multiple possible senses for each word in the same spirit as Buntine (2004) using PCA/LDA-style clustering methods.

To accomplish this, we employ a new version of LDA for sequences: HMM-LDA (Griffiths *et al.* 2005). This model can be viewed as an extension to traditional unsupervised learning for HMMs, where there is one state specially designated as topic state. The non-topic states operate the same way as ordinary HMM states. Each of them emits words from one multinomial distribution. The topic state is different in that it has a distribution over topics, which themselves are multinomial distributions over words. The distribution over topics changes per document, and we sample them from a Dirichlet prior, just as in LDA. By distinguishing between function and content words, this model simultaneously discovers syntactic classes and semantic topics. The word clusters form a smooth manifold on the feature space, and we perform statistical inference to determine the most likely projection for each word-occurrence based on both local dependencies between nearby words and long-term dependencies within one document. With HMM-LDA, it is possible to have different cluster features for the same word in different contexts.

In applications like document classification, each document is viewed as a bag of words, with no structure within the document. Thus the LDA model, with its exchangeability assumption is appropriate, and the underlying word clusters it creates are not embedded any structure either. However, for sequence modeling tasks, we should capture the linear chain dependencies as well as the semantic topics, and HMMs provide this dependency structure—not only providing Markov dependencies at test time, but also leveraging this structure to help guide the discovery of the clusters and the dimensionality reduction or "manifold learning" they provide. We foresee that for data with more complicated dependency structures, more sophiscated structured forms of manifold learning will become of interest. So this paper can be viewed as a preliminary step toward the use of manifold learning in structured data, and we expect further benefits from related work in this area in the future.

We apply our method to two sequence labeling tasks: part-of-speech tagging and Chinese word segmentation. We observe statistically-significant improvements using HMM-LDA clusters over the hierarchical clustering method based on mutual information.

The rest of the paper is organized as follows. First we briefly describe word clustering techniques in Section 2. Then we discuss how to use word cluster features in conditional random fields (Lafferty, McCallum, & Pereira 2001) in Section 3. Section 4 presents experiments applying HMM-LDA and discussion of the results. We conclude the paper in Section 5.

2. Word Clustering

Word clustering provides a method for obtaining more reliable probability estimation for low-frequency features, especially those not observed in the limited labeled data. It organizes words into clusters that form a smooth, low-level feature space. In many sequence modeling tasks, such as part-of-speech tagging, the accuracy for out-of-vocabulary words is much lower than the words observed in training data. This problem can be reduced by introducing word clusters. The OOV words will share features with the observed words in the same clusters and thus have similar preferences for certain labels. We describe two clustering algorithms below.

MI Clustering

Miller, Guinness, & Zamanian (2004) present improved results for named entity recognition with word clusters. The core of the clustering algorithm is a class-based bigram language model, in which the probability of generating a word depends on its own class and also the class of the preceding word (Brown *et al.* 1992). The clustering algorithm tries to partition words into different classes so that the likelihood of generating the whole corpus is maximized. This is essentially the same as maximizing the average mutual information (AMI) between adjacent word classes:

$$I = \sum_{c_1,c_2} P(c_1,c_2)\log\frac{P(c_1,c_2)}{P(c_1)P(c_2)},$$

where $P(c_1, c_2)$ is the probability that words in classes c_1 and c_2 occur in sequence.

The algorithm proceeds in an agglomerative way. It starts with each word in one cluster and repeatedly merges the pair that causes minimum reduction of AMI. The result of running this algorithm is a binary tree containing the whole vocabulary. All the words are at the leaf level and the internal nodes are clusters with different levels of granularities. Nodes at higher levels correspond to larger clusters and lower levels correspond to smaller clusters. Each word has a deterministic path from the root to its leaf.

This algorithm can be viewed as a hard-clustering version of a more general family of algorithms based on mutual information, i.e., the information bottleneck method (Tishby, Pereira, & Bialek 1999). Given a random variable X and a relevant variable Y, the information bottleneck method tries

to find X', a partition of X, such that the mutual information $I(X';Y)$ is maximized under a constraint of the information $I(X';X)$. In this case, X is the vocabulary of words, and X' is the set of word clusters. Information bottleneck tries to find the best partition by maximizing the mutual information between adjacent word clusters. Although the method can be realized using soft-clustering algorithms, which permit one word to probabilistically belong to multiple clusters, it traditionally does not do this, and it still lacks the ability to determine the appropriate cluster to use for each word-occurrence based on their contexts. Therefore, it cannot capture cases where different meanings of one word should have different cluster features.

HMM-LDA

Based on the above considerations, we want our clustering algorithm not only to have the flexibility that each word can probabilistically belong to multiple clusters, but also be able to distinguish between different usages of one word in different contexts. So an LDA-style (Blei, Ng, & Jordan 2003) clustering method is more suitable for this purpose. It provides us with the ability to perform inference for each word-occurrence to decide their most likely clusters. However, it is also very important to capture the local word dependencies for sequence labeling tasks, while LDA only focuses on long-term dependencies within one document.

HMM-LDA (Griffiths *et al.* 2005) is a probabilistic generative model that distinguishes between function words and content words. It incorporates two different components to generate them correspondingly. The syntactic component is a Hidden-Markov-Model (HMM). It handles the local dependencies between nearby words. The semantic component is an LDA model to capture the long-term dependencies within one document, i.e., words in the same document are generally about the same topics. Each component organizes words into finer classes. HMM has a set of states that correspond to different syntactic word classes. It has one special state that hosts the LDA model, and it divides content words into different topics.

More formally, the HMM-LDA model is composed of C classes s_1, s_2, ..., s_C and T topics t_1, t_2, ... t_T. State s_1 is the special state that corresponds to the semantic component, while other states are syntactic word classes. The parameters include $\pi(s_i)$: the transition distribution from state s_i to other states; $\phi(s_i)$ for $i \neq 1$: the multinomial distribution to emit words for syntactic states; $\phi(t_i)$: the topic distribution over words; and $\theta(d)$: the distribution over topics for each document d. Furthermore, these parameters are drawn from Dirichlet prior distributions. $\theta(d)$ is sampled from Dirichlet(α), $\phi(t_i)$ is sampled from Dirichlet(β), $\phi(s_i)$ is sampled from Dirichlet(δ) and $\pi(s_i)$ is sampled from Dirichlet(γ).

Given a sequence of words $\mathbf{w} = \{w_1, w_2, ..., w_n\}$, there are two types of hidden variables: a sequence of class assignments $\mathbf{c} = \{c_1, c_2, ..., c_n\}$, with each c_i being one of the C classes, and a sequence of topic assignments $\mathbf{z} = \{z_1, z_2, ..., z_n\}$, with each z_i being one of the T topics. Note that z_i does not carry meaningful information when $c_i \neq s_1$.

The sampling process for each document d is described below:

1. Sample $\theta(d)$ from Dirichlet(α);
2. For each word w_i in the document,
   - Sample topic z_i from $\theta(d)$;
   - Sample syntactic class c_i from $\pi(c_{i-1})$;
   - If $c_i = s_1$, sample w_i from $\phi(z_i)$, else sample w_i from $\phi(c_i)$;

Inference

There are various algorithms to perform inference and estimate parameters in topic models. The EM algorithm does not perform well due to the large number of parameters and local maxima. Following Griffiths *et al.* (2005), we apply the Gibbs sampling procedure to construct a Markov chain by repeatedly drawing the topic assignment z and class assignment c from their distributions conditioned on all other variables.

Given the words $\mathbf{w}$, the class assignments $\mathbf{c}$, other topic assignments $\mathbf{z}_{-i}$, and hyper-parameters α, β, γ and δ, we sample the topic assignment z_i according to the following distributions:

$$P(z_i|w_i = k, \mathbf{w}_{-i}, \mathbf{z}_{-i}, c_i \neq s_1) \propto \frac{n(d,k)+\alpha}{n(d)+T\alpha},$$

and

$$P(z_i|w_i = k, \mathbf{w}_{-i}, \mathbf{z}_{-i}, c_i = s_1) \propto \frac{n(z_i,k)+\beta}{n(z_i)+V\beta}\frac{n(d,z_i)+\alpha}{n(d)+T\alpha},$$

where $n(z_i,k)$ is the number of times word k is assigned to topic z_i, $n(z_i)$ is the total number of times that any word is assigned to topic z_i, $n(d,k)$ is the number of occurrences of word k in document d, $n(d,z_i)$ is the number of occurrences of topic z_i in d and $n(d)$ is the total number of tokens in d. V is the vocabulary size and T is the number of topics.

Similarly, the class assignment c_i given other variables is drawn from:

$$P(c_i|w_i = k, \mathbf{w}_{-i}, \mathbf{z}, c_i \neq s_1) \propto \frac{n(c_i,k)+\delta}{n(c_i)+V\delta}P(c_i|c_{i-1}),$$

and

$$P(c_i|w_i = k, \mathbf{w}_{-i}, \mathbf{z}, c_i = s_1) \propto \frac{n(z_i,k)+\beta}{n(z_i)+V\delta}P(c_i|c_{i-1}),$$

where

$$P(c_i|c_{i-1}) \propto \frac{(n(c_{i-1},c_i)+\gamma)(n(c_i,c_{i+1})+I(c_{i-1}=c_i)I(c_i=c_{i+1})+\gamma)}{n(c_i)+I(c_{i-1}=c_i)+C\gamma}.$$

In the above equations, $n(c_i,k)$ is the number of times word k is assigned to class c_i, $n(c_i)$ is the total number of times that any word is assigned to class c_i, $n(c_{i-1},c_i)$ is the number of transitions from c_{i-1} to c_i, $n(c_i,c_{i+1})$ is the number of transitions from c_i to c_{i+1}, and C is the total number of syntactic classes. I is an indicator function that returns 1 if the argument is true and 0 otherwise.

3. CRFs with Cluster Features

We choose conditional random fields (CRFs) (Lafferty, McCallum, & Pereira 2001) as our discriminative training model. CRFs are undirected graphical models that calculate the conditional probability of values on designated output nodes given values on other designated input nodes. In a special case where the output nodes form a linear chain, CRFs make a first-order Markov independence assumption, and define the conditional probability of a state sequence $\mathbf{s} =< s_1, s_2, ..., s_T >$ given an observation sequence $\mathbf{o} =< o_1, o_2, ..., o_T >$ as:

$$P_\Lambda(\mathbf{s}|\mathbf{o}) = \frac{1}{Z_\mathbf{o}}(\exp(\sum_{t-1}^{T}\sum_{k}\lambda_k f_k(s_{t-1}, s_t, \mathbf{o}, t))),$$

where $f_k(s_{t-1}, s_t, \mathbf{o}, t)$ is a feature function. As discriminative models, CRFs are able to incorporate a large number of overlapping non-independent features. We can ask specific questions about the input via feature functions. For example, one feature could ask whether the current word is capitalized and the previous word is *the*. It takes a value of 1 if the answer is true and 0 otherwise. Each feature is assigned a weight λ_k, and we learn them via training. $Z_\mathbf{o}$ is a normalization factor to make all the probabilities sum to 1.

To train a CRF with labeled data, we maximize the log-likelihood of state sequences given observation sequences:

$$L_\Lambda = \sum_{i=1}^{N} \log P_\Lambda(\mathbf{s}^{(i)}|\mathbf{o}^{(i)}),$$

where $\{\langle \mathbf{o}^{(i)}, \mathbf{s}^{(i)} \rangle\}$ is the labeled training data. We use L-BFGS (Byrd, Nocedal, & Schnabel 1994) to find the parameters that optimize the objective function.

To incorporate word clusters in CRFs, we need to create one corresponding feature function for each cluster. The feature takes a value of 1 if the word is assigned to the cluster, and 0 otherwise. When we use the hierarchical MI clustering, each token will have several cluster features on because we simultaneously consider multiple clusters with different levels of granularity. When we use the HMM-LDA model, only one cluster feature is on for each token, which is the most likely HMM state, as determined by inference.

4. Experiment Results

Word Clusters

We apply the HMM-LDA model to two datasets. One is the Wall Street Journal (WSJ) collection labeled with part-of-speech tags. There are a total of 2312 documents in this corpus, 38665 unique words and 1.2M word tokens. The other dataset is the Chinese Treebank collection (CTB2 and CTB4) and two subsets of the bakeoff collection (CTB and PK) (Sproat & Emerson 2003). The total number of documents in the Chinese dataset is 1480. The vocabulary size is 5019 and the number of tokens is about 3.2M.

For the WSJ dataset, we run the algorithm using 50 topics and 40 syntactic classes chosen somewhat arbitrarily to be approximately the same as the number of POS tags. And for the Chinese dataset, we arbitrarily use 100 topics and 50 classes. We did not find factor of 2 differences in the number of topics to have a significant impact on the performance. When running Gibbs Sampling, we take a total of 40 samples with a lag of 100 iterations between them and an initial burn-in period of 4000 iterations. Some of the example syntactic classes and topics generated from the WSJ dataset are shown in Table 1 and Table 2.

Most clusters have obvious correlations with certain POS tags. For example, the cluster displayed in the first column in Table 1 contains mostly verbs that take direct objects. Topic words are usually nouns. In the third cluster in Table 1, we see that the word *offer* co-occurs with mostly nouns. This feature will help determine the POS tag for this word if we only see *offer* used as a verb in the limited labeled data.

To compare with the HMM-LDA model, we also implement MI clustering. After running this algorithm, we create a binary tree for the whole vocabulary. Each word can be uniquely represented as a bit string by following the path from the root to its leaf, acquiring a 0 or 1 at each step based on the branch it takes. The word cluster features are encoded using prefixes of these bit strings as in Miller, Guinness, & Zamanian (2004). We can include clusters with different levels of granularities by choosing multiple prefix lengths. As we mention before, all occurrences of the same word share the same set of cluster features.

Evaluation

We evaluate these methods on two sequence labeling tasks: English POS tagging and Chinese word segmentation.

POS Tagging We conduct POS tagging on the WSJ dataset. The output set for this task is composed of 45 POS tags and the input features include: word unigrams, word bigrams, 15 types of spelling features such as capitalization and number patterns, and word suffixes of lengths of 2, 3 and 4. In addition to these features, semi-supervised learning methods can also use word cluster features generated from unlabeled data. After running the HMM-LDA model, we obtain not only 40 syntactic classes and 50 topics, but also 40 samples of the class and topic assignments. We then determine the most likely class for each token from counting over all the samples and use this single class as an additional feature for supervised training. In the MI clustering method, we use prefixes of lengths 8, 12, 16 and 20 of bit strings. No external lexicons are used.

We conduct various experiments using different numbers of labeled instances. The performance is evaluated using the average token error rate. We present these results on all tokens and OOV tokens in Table 3. Overall, both semi-supervised methods outperform the purely supervised approach with no cluster features. The improvement is more obvious when there is less labeled data. For example, when there is only 10K labeled words, we obtain a 14.74% reduction in error rate using HMM-LDA clusters. Furthermore, HMM-LDA performs better than MI clustering. According to the sign test, the improvement is statistically significant. HMM-LDA is particularly advantageous for OOV words. As the number of labeled words increases, MI clustering shows little improvement over supervised method for OOV words, while HMM-LDA consistently reduces the error rate.

make	0.0279	of	0.7448	way	0.0172	last	0.0767	to	0.6371
sell	0.0210	in	0.0828	agreement	0.0140	first	0.0740	will	0.1061
buy	0.0174	for	0.0355	price	0.0136	next	0.0479	would	0.0665
take	0.0164	from	0.0239	time	0.0121	york	0.0433	could	0.0298
get	0.0157	and	0.0238	bid	0.0103	third	0.0424	can	0.0298
do	0.0155	to	0.0185	effort	0.0100	past	0.0368	and	0.0258
pay	0.0152	;	0.0096	position	0.0098	this	0.0361	may	0.0256
go	0.0113	with	0.0073	meeting	0.0098	dow	0.0295	should	0.0129
give	0.0104	that	0.0055	offer	0.0093	federal	0.0288	might	0.0103
provide	0.0086	or	0.0039	day	0.0092	fiscal	0.0262	must	0.0083

Table 1: Sample syntactic word clusters, each column displaying the top 10 words in one cluster and their probabilities

bank	0.0918	computer	0.0610	jaguar	0.0824	ad	0.0314	court	0.0413
loans	0.0327	computers	0.0301	ford	0.0641	advertising	0.0298	judge	0.0306
banks	0.0291	ibm	0.0280	gm	0.0353	agency	0.0268	law	0.0210
loan	0.0289	data	0.0200	shares	0.0249	brand	0.0181	lawyers	0.0210
thrift	0.0264	machines	0.0191	auto	0.0172	ads	0.0177	case	0.0195
assets	0.0235	technology	0.0182	express	0.0144	saatchi	0.0162	attorney	0.0161
savings	0.0220	software	0.0176	maker	0.0136	brands	0.0142	suit	0.0143
federal	0.0179	digital	0.0173	car	0.0134	account	0.0120	state	0.0138
regulators	0.0146	systems	0.0169	share	0.0128	industry	0.0106	federal	0.0138
debt	0.0142	business	0.0151	saab	0.0116	clients	0.0105	trial	0.0126

Table 2: Sample semantic word clusters, each column displaying the top 10 words in one cluster and their probabilities

		No Cluster	MI	HMMLDA
10K	Overall	10.04	9.46 (5.78)	8.56 (14.74)
24.46%	OOV	22.32	21.56 (3.40)	18.49 (17.16)
30K	Overall	6.08	5.85 (3.78)	5.40 (11.18)
15.31%	OOV	17.34	17.35 (-0.00)	15.01 (13.44)
50K	Overall	5.34	5.12 (4.12)	4.79 (10.30)
12.49%	OOV	16.36	16.21 (0.92)	14.45 (11.67)

Table 3: POS tagging error rates (%) for CRFs with no cluster features, using clusters from MI clustering and clusters from HMM-LDA. The first column displays the numbers of labeled words and OOV rates for three different settings. We report the overall error rates for all tokens in the test sets and also error rates for OOV words only. The numbers in the parenthesis are error reduction rates of the two semi-supervised methods over supervised learning with no clusters.

Chinese Word Segmentation We use very limited features for Chinese word segmentation experiments. The only features are word unigrams, word bigrams and word cluster features generated from HMM-LDA. We conduct two-label "BI" segmentation, which corresponds to beginning and inner part respectively. A segment is considered correctly labeled only if its two boundaries are identified correctly. We evaluate our method with the inherent training-test splits of the two bakeoff datasets: CTB and PK. Their statistics are listed in Table 4. We report errors in precision, recall and F1 in Table 5 and also error reduction by HMM-LDA clusters. Similar to POS tagging, experiments with cluster features created from unlabeled corpus using HMM-LDA statistically-significantly improve performance over purely supervised experiments with no cluster features.

	# Training words	# Test words	OOV rate
CTB	250K	40K	18.1%
PK	1.1M	17K	6.9%

Table 4: statistics for Chinese word segmentation datasets

		P	R	F1
CTB	No Cluster	16.31	16.40	16.36
	HMMLDA	15.06(7.67)	15.81(3.72)	15.44(5.62)
PK	No cluster	8.48	8.62	8.55
	HMMLDA	7.83(7.67)	8.10(6.03)	7.96(6.13)

Table 5: Chinese word segmentation error rates for precision, recall and F1 in %. We compare supervised learning with no cluster features and semi-supervised learning with clusters from HMM-LDA on two datasets: CTB and PK. The numbers in the parenthesis are error reduction rates of HMM-LDA over supervised learning.

Discussion

We show improvements from using HMM-LDA clusters for both POS tagging and Chinese word segmentation. With this model, We are able to capture the different usages of one word and assign them into different clusters. For example, in the Chinese dataset, the word *san1* has different senses in the following contexts:

1. "*san1* xia2 gong1 cheng2"
2. "jin1 *san1* jiao3"
3. "ji1 zhe3 *san1* yue4 ba1 ri4 dian4"

In the first two cases, *san1* appears in two different place names, and is assigned to semantic classes. In the third case, the two-character phrase beginning with "san1" means "March". Because dates appear in a regular syntax throughout the corpus, here "san1" is placed in one of the syntactic classes. HMM-LDA successfully recognizes its different roles in different contexts.

To understand how cluster features can help determine the correct tags in ambiguous situations, we consider the following example for POS tagging:

"With other anxious calls pouring, ..."

When there are no cluster features, the words *anxious* and *calls* are mistakenly tagged as NNS and VBZ respectively. It is probably because *calls* is mostly used as a verb in the training data. When we add cluster features that assign *anxious* to a cluster that contains adjectives such as *new*, *more*, *serious*, etc, and *calls* to topic words, they are correctly labeled as JJ and NNS respectively.

However, using clusters generated by HMM-LDA sometimes introduces new errors. One common error is to tag adjectives as nouns when they are determined to be topic words. While topic words are usually nouns, there are also words with other tags. One possible improvement would be to take into account the actual topic assignment for each token instead of treating all assignments in the same way.

5. Conclusion and Future Work

In this paper, we present a method of semi-supervised learning for sequence-labeling tasks using conditional random fields. Similar to the approach taken by Buntine (2004) and Miller, Guinness, & Zamanian (2004), we create word clusters from unlabeled data and use them as features for supervised training. Once the cluster features are added, there is no more inference needed than ordinary supervised models. We apply the HMM-LDA model to generate both syntactic and semantic word clusters. By allowing one word to probabilistically belong to multiple clusters, we can model their different senses in different contexts. We evaluate our method on two sequence labeling tasks: part-of-speech tagging and Chinese word segmentation. For both of them, HMM-LDA significantly improves the performance over supervised learning, especially for OOV words. And for the part-of-speech tagging task, we also obtain better results than the MI clustering algorithm.

Compared to the work by Miller, Guinness, & Zamanian (2004), which generates word clusters using a corpus with about 100 million words, we are using a very small amount of unlabeled data for semi-supervised learning. We are interested in applying HMM-LDA to larger collections. Another possible direction of future work is to incorporate a hierarchical structure among topics, just as in the MI clustering method. The advantage of doing so is that we no longer need to specify the number of topics arbitrarily, but can automatically discover clusters with different levels of granularity.

ACKNOWLEDGMENT

We thank Tom Griffiths for graciously sharing his implementation of HMM-LDA with us.

This work was supported in part by the Center for Intelligent Information Retrieval, in part by The Central Intelligence Agency, the National Security Agency and National Science Foundation under NSF grant #IIS-0326249, and in part by the Defense Advanced Research Projec ts Agency (DARPA), through the Department of the Interior, NBC, Acquisition Services Division, under contract number NBCHD030010. Any opinions, findings and conclusions or recommendations expressed in this material are the author(s)' and do not necessarily reflect those of the sponsor.

References

Blei, D. M.; Ng, A. Y.; and Jordan, M. I. 2003. Latent dirichlet allocation. *Machine Learning Research* 3:1107–1135.

Brown, P. F.; Pietra, V. J. D.; deSouza, P. V.; Lai, J. C.; and Mercer, R. L. 1992. Class-based n-gram models of natural language. *Computational Linguistics* 18(4):467–479.

Buntine, W. 2004. Applying discrete pca in data analysis. In *Proceedings of 20th Conference on Uncertainty in Artificial Intelligence (UAI)*, 59–66.

Byrd, R.; Nocedal, J.; and Schnabel, R. 1994. Representations of quasi-newton matrices and their use in limited memory methods. *Mathematical Programming* 63:129–156.

Griffiths, T. L.; Steyvers, M.; Blei, D. M.; and Tenenbaum, J. B. 2005. Integrating topics and syntax. In Saul, L. K.; Weiss, Y.; and Bottou, L., eds., *Advances in Neural Information Processing Systems 17*. Cambridge, MA: MIT Press. 537–544.

Lafferty, J.; McCallum, A.; and Pereira, F. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. 18th International Conf. on Machine Learning*.

Li, W., and McCallum, A. 2004. A note on semi-supervised learning using markov random fields. Technical Report.

Miller, S.; Guinness, J.; and Zamanian, A. 2004. Name tagging with word clusters and discriminative training. In *Proceedings of HLT-NAACL*.

Nigam, K.; McCallum, A. K.; Thrun, S.; and Mitchell, T. M. 2000. Text classification from labeled and unlabeled documents using em. *Machine Learning* 39(2/3):103–134.

Sproat, R., and Emerson, T. 2003. The first international chinese word segmentation bakeoff. In *Proceedings of the Second SIGHAN Workshop on Chinese Language Processing*.

Tishby, N.; Pereira, F. C.; and Bialek, W. 1999. The information bottleneck method. In *Proceedings of the 37-th Annual Allerton Conference on Communication, Control and Computing*, 368–377.

Zhu, X.; Ghahramani, Z.; and Lafferty, J. 2003. Semi-supervised learning using gaussian fields and harmonic functions. In *Proceedings of ICML*.