Bigram HMM with Context Distribution Clustering for Unsupervised Chinese Part-of-Speech tagging

Lidan Zhang
Department of Computer Science
the University of Hong Kong
Hong Kong
lzhang@cs.hku.hk

Kwok-Ping Chan
Department of Computer Science
the University of Hong Kong
Hong Kong
kpchan@cs.hku.hk

Abstract

This paper presents an unsupervised Chinese Part-of-Speech (POS) tagging model based on the first-order HMM. Unlike the conventional HMM, the number of hidden states is not fixed and will be increased to fit the training data. In favor of sparse distribution, the Dirichlet priors are introduced with variational inference method. To reduce the emission variables, words are represented by their contexts and clustered based on the distributional similarities between contexts. Experiment results show the output state sequence of HMM are highly correlated to the latent annotations of gold POS tags, in context of clustering similarity measures. The other experiments on a real application, unsupervised dependency parsing, reveal that the output sequence can replace the manually annotated tags without loss of accuracies.

1 Introduction

Recently latent variable model has shown great potential in recovering the underlying structures. For example, the task of POS tagging is to recover the appropriate sequence structure given the input word sequence (Goldwater and Griffiths, 2007). One of the most popular example of latent models is Hidden Markov Model (HMM), which has been extensively studied for many years (Rabiner, 1989). The key problem of HMM is how to find an optimal hidden state number and the topology appropriately.

In most cases, the topology of HMM is predefined by exploiting the domain or empirical knowledge. This topology will be fixed during the whole process. Therefore how to select the optimal topology for a certain application or a set of training data is still a problem, because many researches show that varying the size of the state space greatly affects the performance of HMM. Generally there are two ways to adjust the state number: top-down and bottom-up methods. In the bottom-up methods (Brand, 1999), the state number is initialized with a relatively large number. During the training, the states are merged or trimmed and ended with a small set of states. On the other hand, the top-down methods (Siddiqi et al., 2007) start from a small state set and split one or some states until no further improvement can be obtained. The bottom-up approaches require huge computational cost in deciding the states to be merged, which makes it impractical for applications with large state space. In this paper, we focus on the latter approaches.

Another problem in HMM is that EM algorithm might yield local maximum value. Johnson (2007) points out that training HMM with EM gives poor results because it leads to a fairly flat distribution of hidden states when the empirical distribution is highly skewed. A multinomial prior, which favors sparse distribution, is a good choice for natural language tasks. In this paper, we proposed a new procedure for inferring the HMM topology and estimating its parameters simultaneously. Gibbs sampling has been used in infinite HMM (iHMM) (Beal et al., 2001; Fox et al., 2008; Van Gael et al., 2008) for inference. Unfortunately Gibbs sampling is slow and diffi-

cult to be converged. In this paper, we proposed the variational Bayesian inference for the adaptive HMM model with Dirichlet prior. It involves a modification to the Baum-Welch algorithm. In each iteration, we replaced only one hidden state with two new states until convergence.

To reduce the number of observation variables, the words are pre-clustered and represented by the exemplar within the same cluster. It is a one-to-many clustering, because the same word play different roles under different contexts. We evaluate the similarity between the distribution of contexts, with the assumption that the context distribution implies syntactic pattern of the given word (Zelling, 1968; Weeds and Weir, 2003). With this clustering, more contextual information can be considered without increasing the model complexity. A relatively simple model is important for unsupervised task in terms of computational burden and data sparseness. This is the reason why we do not increase the order of HMM(Kaji and Kitsuregawa, 2008; Headden et al., 2008).

With unsupervised algorithms, there are two aspects to be evaluated (Van Gael et al., 2009). Fist one is how good the outcome clusters are. We compare the HMM results with the manually POS tags and report the similarity measures based on information theory. On the other hand, we test how good the outputs act as an intermediate results. In many natural language tasks, the inputs are word class, not the actual lexical item, for reason of sparsity. In this paper, we choose the unsupervised dependency parsing as the application to investigate whether our clusters can replace the manual labeled tags or not.

The paper is organized as below: in section 2, we describe the definition of HMM and its variance inference. We present our dynamic HMM in section 3. To overcome the context limitation in the first-order HMM, we present our distributional similarity clustering in section 4. In section 5, we reported the results of the mentioned experiments while section 6 concludes the paper.

2 Terminology

The task of POS tagging is to assign a syntactic category sequence to the input words. Let $\mathcal{S}$ be defined as the set of all possible hidden states, which are expected to be highly correlated to POS tags. Σ represents the set of all words. Therefore the task is to find a sequence of tag sequence $S = s_1...s_n \in \mathcal{S}$ given a sequence of words (i.e. a sentence, $W = w_1...w_n \in \Sigma$). The optimal tags is to maximize the conditional probability $p(S|W)$, which is equal to:

$$\begin{aligned} \max_S p(S|W) &= \max_S p(S)p(W|S) \\ &= \max_S p(W,S) \end{aligned} \tag{1}$$

In this paper,we consider the first-order HMM, where the POS tags are regarded as hidden states and words as observed variables. According to the Markov assumption, the best sequence of tags S for a given sequence of words W is done by maximizing (with $s_0 = 0$) the joint probability:

$$p(W,S) = \prod_{i=1}^{n} p(s_i|s_{i-1})p(w_i|s_i) \tag{2}$$

where w_0 is the special boundary marker of sentences.

2.1 Variational Inference for HMM

Let the HMM be modeled with parameter $\theta = (A, B, \pi)$, where $A = \{a_{ij}\} = \{P(s_t = j|s_{t-1} = i)\}$ is the transition matrix governing the dynamic of the HMM. $B = \{b_t(i)\} = \{P(w_t = i|s_t)\}$ is the state emission matrix and $\pi = \{\pi_i\} = \{P(s_1 = i)\}$ assigns the initial probabilities to all hidden states. In favor of sparse distributions, a natural choice is to encode Dirichlet prior into parameters $p(\theta)$. In particular, we have:

$$\begin{aligned} p(A) &= \prod_{i=1}^{N} Dir(\{a_{i1},...,a_{iN}\}\,|u^{(A)}) \\ p(B) &= \prod_{i=1}^{N} Dir(\{b_{i1},...,b_{iN}\}\,|u^{(B)}) \\ p(\pi) &= Dir(\{\pi_1,...,\pi_N\}\,|u^{(\pi)}) \end{aligned} \tag{3}$$

where the Dirichlet distribution of order N with hyperparameter vector u is defined as:

$$Dir(x|u) = \frac{\Gamma(\sum_{i=1}^{N} u_i)}{\prod_{i=1}^{N} \Gamma(u_i)} \prod_{i=1}^{N} x_i^{u_i - 1}. \quad (4)$$

In this paper, we consider the symmetric Dirichlet distribution with a fixed length, i.e. $u = [\sum_{i=1}^{N} u_i/N, ..., \sum_{i=1}^{N} u_i/N]$.

In the Bayesian framework, the model parameters are also regarded as hidden variables. The marginal likelihood can be calculated by summing up all hidden variables. According to the Jensen's inequality, the lower bound of marginal likelihood is defined as:

$$\begin{aligned} \ln p(W) &= \ln \int \sum_{S} p(\theta) p(W, S|\theta) d\theta \\ &\geq \int \sum_{S} q(\theta, S) \ln \frac{p(W, S, \theta)}{q(\theta, S)} d\theta \quad (5) \\ &= \mathcal{F} \end{aligned}$$

Generally, Variational Bayesian Inference aims to find a tractable distribution $q(\theta, s)$ that maximizes the lower bound $\mathcal{F}$. To make inference flexible, the posterior distribution can be assumed to be factorized according to the mean-field assumption. We have:

$$p(W, S, \theta) \approx q(S, \theta) = q_\theta(\theta) q_S(S) \quad (6)$$

Then an extension of EM algorithm (called Baum-Welch algorithm) can be used to alternately optimize the q_S and q_θ. The EM process is described as follows:

- **E Step**: Forward-Backward algorithm to find the optimal state sequence $S^{(t+1)} = \arg\max p(S^{(t)}|W, \theta^{(t)})$
- **M Step**: The parameters $\theta^{(t+1)}$ are re-estimated given the optimal state $S^{(t+1)}$

The E and M steps are repeated until a convergence criteria is satisfied. Beal (2003) proved that only need to do minor modifications in M step (in 1) is needed, when Dirichlet prior is introduced.

3 Adaptive Hidden Markov Model

As aforementioned, the key problem of HMM is how to initialize the number of hidden states and select the topology of HMM. In this paper, we use the top-down scheme: starting from a small number of states, only one state is chosen in each step and splitted into two new states. This binary split scheme is described in Figure 1.

Algorithm 1 Outline of our adpative HMM

Initialization: Initialize: $t = 0$, $N^{(t)}$
repeat
  Optimization: Find the optimal parameters for current N^t
  Candidate Generation: Split states and generate candidate HMMs
  Candidate Selection: Select the optimal HMM from the candidates, whose hidden state number is N^{t+1}
until No further improvement can be achieved after splitting

In the following, we will discuss the details of each step one by one.

3.1 Candidate Generation

Let $N^{(t)}$ represent the number of hidden states at timestep t. The problem is how to choose the states for splitting. A straightforward way is to select all states and generate $N^{(t)} + 1$ candidate HMMs, including the original un-splitted one. Obviously the exhaustive search is inefficient especially for large state space. To make the algorithm more efficient, some constraints must be set to narrow the search space.

Intuitively entropy implies uncertainty. So hidden states with large conditional entropies are desirable to be splitted. We can define the conditional entropy of the state sequences given observation W as:

$$H(S|W) = -\sum_{S} [P(S|W) \log P(S|W)] \quad (8)$$

Our assumption is the state to be splitted must be the states sequence with the highest conditional entropy value. This entropy can be recursively calculated with complexity $O(N^2 T)$ (Hernando et al., 2005). Here N is the number of

$$A^{(t+1)} = \{a_{ij}^{(t+1)}\} = \exp[\psi(\omega_{ij}^{(A)}) - \psi(\sum_{j=1}^{N} \omega_{ij}^{(A)})] \; ; \qquad \omega_{ij}^{(A)} = u_j^{(A)} + \mathbb{E}_{q(s)}[n_{ij}]$$

$$B^{(t+1)} = \{b_{ik}^{(t+1)}\} = \exp[\psi(\omega_{ik}^{(B)}) - \psi(\sum_{k=1}^{T} \omega_{ik}^{(B)})] \; ; \qquad \omega_{ik}^{(B)} = u_k^{(B)} + \mathbb{E}_{q(s)}[n'_{ik}] \tag{7}$$

$$\pi^{(t+1)} = \{\pi_i^{(t+1)}\} = \exp[\psi(\omega_i^{(\pi)}) - \psi(\sum_{i=1}^{N} \omega_j^{(\pi)})]; \qquad \omega_i^{(\pi)} = u_i^{(\pi)} + \mathbb{E}_{q(s)}[n''_i]$$

Figure 1: Parameters update equations in M-step. Here $\mathbb{E}$ is the expectation with respect to the model parameters. And n_{ij} is the expected number of transition from state s_i to state s_j; n'_{ik} is the expected number of times word w_k occurs with state s_i; n''_i is the occurrence of $s_0 = i$

states and T is the length of sequence. Using this entropy constraint, the size of candidate state set is always smaller than the minimal value between N and T.

3.2 Candidate Selection

Given the above candidate set, the parameters of each HMM are to be updated. Note that we just update the parameters related to the split state, whilst keep the others fixed. Suppose the i-th hidden state is replaced by two new states. First the transition matrix is enlarged from $N^{(t)} \times N^{(t)}$ dimension to $(N^{(t)} + 1) \times (N^{(t)} + 1)$ dimension, by inserting one column and row after the i-th column and row. In the process of update, we only change the items in the two (i and $i + 1$) rows and columns. The other elements irrelevant to the split state are not involved in the update procedure. Similarly EM algorithm is used to find the optimal parameters. Note that most of the calculations can be skipped by making use of the forward and backward probability matrix achieved in the previous step. Therefore the convergence is fast.

Given the candidate selection, we can use a modified Baum-Welch algorithm to find optimal states and parameters. Here we use the algorithm in (Siddiqi et al., 2007) with some modifications for the Dirichlet prior. In particular, in E step, we follow their partial Forward-Background algorithm to calculate $\mathbb{E}[n_{ij}]$ and $\mathbb{E}[n'_{ik}]$, if s_i or s_j is candidate state to be splitted. Then in M-step, only rows and columns related to the candidate state are updated according to equation (7). The detailed description is given as appendix.

Finally it is natural to use variational bound of marginal likelihood in equation (5) for model scoring and convergence criterion.

4 Distributional Clustering

To reduce the number of observation variables, the words are clustered before HMM training. Intuitively, the words share the similar contexts have similar syntactic property. The categories of many words are varied in different contexts. In other words, the cluster of a given word is heavily dependent on the context it appears. For example, 发现 can be a noun (meaning: discovery) if it acts as an object, or a verb (meaning: to discover) if it is followed with a noun. Furthermore the introduction of context can overcome the limited context in the first-order HMM.

The underlying hypothesis of clustering based on distributional similarity is that the words occurring in similar contexts behave as similar syntactic roles. In this work, the context of a word is a trigram consist of the word immediately preceding the target and the word immediately following it. The similarity between two words is measured by Pointwise Mutual Information (PMI) between the context pair in which they appear:

$$PMI(w_i, w_j) = \log \frac{P(c_i, c_j)}{P(c_i)P(c_j)} \tag{9}$$

where c_i denotes the context of w_i. $P(c_i, c_j)$ is the co-occurrence probability of c_i and c_j, and

$P(c_i) = \sum_j P(c_i, c_j)$ is the occurrence probability of c_i. In our experiments, the cutoff context count is set to 10, which means the frequency less than the threshold is labeled as the unknown context.

The above distributional similarity can be used as a distance measure. Hence any clustering algorithm can be adopted. In this paper, we use the affinity propagation algorithm (Frey and Dueck, 2007). Its parameter 'dampfact' is set to 0.9, and the other parameters are set as default. After running the clustering algorithm, the contexts are clustered into 1869 clusterings. It is noted that one word might be classified into several clusters , if its contexts are clustered into several clusters.

5 Experiments

As aforementioned, the outputs of our HMM model are evaluated in two ways, clustering metric and parsing performance. The data used in all experiments are the Chinese data set in CoNLL-2007 shared task. The number of tokens in training, development and test sets are 609,060, 49,620 and 73,153 respectively. We use all training data set for training the model, whose maximum length is 242.

The hyper parameters of Dirichlet priors are initialized in a homogeneous way. The initial hidden state is set to 40 in all experiments. After several iterations, the hidden states number converged to 247, which is much larger than the size of the manually defined POS tags. Our expectation is the refinement variables can reveal the deep granularity of the POS tags.

5.1 Clustering Evaluation

In this paper, we use information theoretic based metrics to quantify the information shared by two clusters. The most common information-based clustering metric is the variational of Information (VI)(Meilă, 2007). Given the clustering result C_r and the gold clustering C_g, VI sums up the conditional entropy of one cluster distribution given the other one:

$$\begin{aligned} \mathrm{VI}(C_r, C_g) &= H(C_r) + H(C_g) - 2I(C_r, C_g) \\ &= H(C_r|C_g) + H(C_g|C_r) \end{aligned} \tag{10}$$

where $H(C_r)$ is the entropy associated with the clustering C_r, and mutual information $I(C_r, C_g)$ quantifies the mutual dependence between two clusterings, or say the shared information between two variables. It is easy to see that $\mathrm{VI} \in [0, \log(N)]$, where N is the number of data points. However, the standard VI is not normalized, which favors clusterings with a small number of clusters. It can be normalized by dividing by $\log(N)$, because the number of training instances are fixed. However the normalized VI score is misleadingly large, if the N is very large which is the case in our task. In this paper only un-normalized VI scores are reported to show the score ranking.

To standardize the measures to have fixed bounds, (Strehl and Ghosh, 2003) defined the normalized Mutual Information (NMI) as:

$$NMI(C_r, C_g) = \frac{I(C_r, C_g)}{\sqrt{H(C_r)H(C_g)}} \tag{11}$$

NMI takes its lower bound of 0 if no information is shared by two clusters and the upper bound of 1 if two clusterings are identical. The NMI however, still has problems, whose variation is sensitive to the choice of the number of clusters.

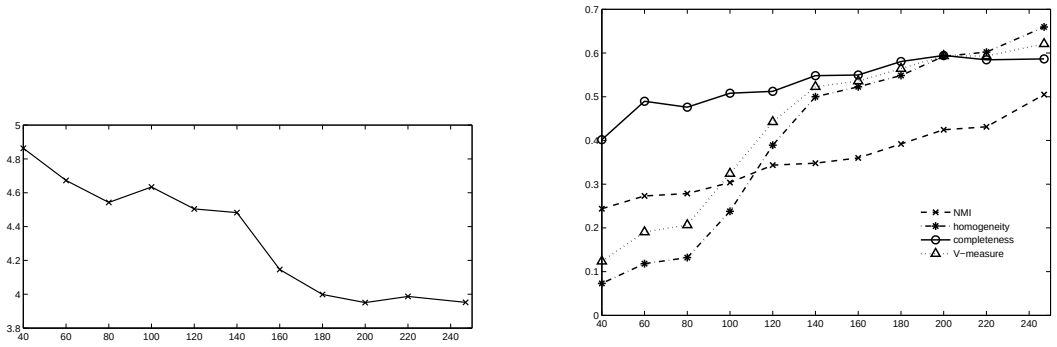
Rosenberg and Hirschberg (2007) proposed V-measure to combine two desirable properties of clustering: homogeneity (h) and completeness (c) as follows:

$$\begin{aligned} h &= 1 - H(C_g|C_r)/H(C_g) \\ c &= 1 - H(C_r|C_g)/H(C_r) \\ V &= 2hc/(h + c) \end{aligned} \tag{12}$$

Generally homogeneity and completeness runs in opposite way, whose harmonic mean (i.e. V-measure) is a comprise score, just like F-score for the precision and recall.

Let us first examine the contextual word clustering performance. The VI score between distributional word categories and gold standard is 2.39. The NMI and V-measure score are 0.53 and 0.48, respectively.

The clustering performance of the HMM outputs are reported in Figure 2. The best VI score achieved was 3.9524, while V-measure was 62.09% and NMI reached 0.8051. Previous



(a) VI score

(b) normalized scores

Figure 2: Clustering evaluation metrics against number of hidden states

work of Chinese tagging focuses on the tagging accuracies, e.g. Wang (Wang and Schuurmans,) and Huang et al. (Huang et al., 2007). To our knowledge, this is the first work to report the distributional clustering similarity measures based on informatics view for Chinese . Similar works can be found on English of WSJ corpus (Van Gael et al., 2009). Their best results of VI, V-measure, achieved with Pitman-Yor prior, were 3.73 and 59%. We believe the Chinese results are not good as English correspondences because of the rich unknown words in Chinese (Tseng et al., 2005).

5.2 Dependency Parsing Evaluation

The next experiment is to test the goodness of the outcome states of our model in the context of real tasks. In this work, we consider unsupervised dependency parsing for a fully unsupervised system. The dependency parsing is to extract the dependency graph whose nodes are the words of the given sentence. The dependency graph is a directed acyclic graph in which every edge links from a head word to its dependent. Because we work on unsupervised methods in this paper, we choose a simple generative head-outward model (Dependency Model with Valence, DMV) (Klein and Manning, 2004; Headden III et al., 2009) for parsing. The data through the experiment is restricted to the sentences up to length 10 (excluding punctuation).

Because the main purpose is to test the HMM output rather than to improve the parsing performance, we select the original DMV model without extensions or modifications. Starting from the root, DMV generates the head, and then each head recursively generates its left and right dependents. In each direction, the possible dependents are repeatedly chosen until a STOP marker is seen. DMV use inside-outside algorithm for re-estimation. We choose the "harmonic" initializer proposed in (Klein and Manning, 2004) for initialization. The valence information is the simplest binary value indicating the adjacency. For different HMM candidates with varied hidden state number, we directly use the outputs as the input of the DMV and trained a set of models. Performing test on these individual models, we report the directed dependency accuracies (the fraction of words assigned the correct parent) in Figure 3.

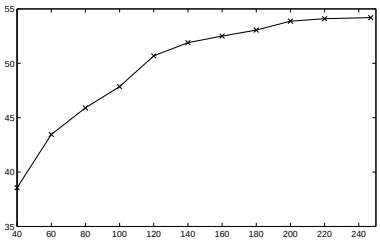


Figure 3: Directed accuracies for different hidden states

It is noted that the accuracy monotonically

increases when the number of states increases. The most drastic increase happened when state changes from 40 to 120. The accuracy increased from 38.56% to 50.60%. If the state number is larger than 180, the increase is not obvious. The final best accuracy is 54.20%, which improve the standard DMV model by 5.6%. Therefore we can see that the introduction of more annotations can help the parsing results. However, the improvement is limited and stable when the number of state number is large. To further improve the parsing performance, one might turn to the extension of DMV model, e.g. introducing more knowledge (prior or lexical information) or more sophistical smoothing techniques. However, the development of parser is not the focus of this paper.

6 Conclusion and Future Work

This paper works on the unsupervised Chinese POS tagging based on the first-order HMM. Our contributions are: 1). The number of hidden states can be adjusted to fit the data. 2). For inference, we use the variational inference, which is faster and is guaranteed theoretically to convergence. 3). To overcome the context limitation in HMM, the words are clustered based on distributional similarities. It is a 1-to-many clustering, which means one word might be classified into different clusters under different contexts. Finally, experiments show the hidden states are correlated to the latent annotations of the standard POS tags.

The future work includes to improve the performance by incorporating a small amount of supervision. The typical supervision used before is dictionary extracted from a large corpus like Chinese Gigaword. Another interesting idea is to select some exemplars (Haghighi and Klein, 2006).

References

Beal, Matthew J., Zoubin Ghahramani, and Carl Edward Rasmussen. 2001. The infinite hidden markov model. In *NIPS*, pages 577–584.

Beal, M. J. 2003. Variational algorithms for approximate bayesian inference. *Phd Thesis. Gatsby Computational Neuroscience Unit, University College London*.

Brand, Matthew. 1999. An entropic estimator for structure discovery. In *Proceedings of the 1998 conference on Advances in neural information processing systems II*, pages 723–729, Cambridge, MA, USA. MIT Press.

Fox, Emily B., Erik B. Sudderth, Michael I. Jordan, and Alan S. Willsky. 2008. An hdp-hmm for systems with state persistence. In *ICML '08: Proceedings of the 25th international conference on Machine learning*.

Frey, Brendan J. and Delbert Dueck. 2007. Clustering by passing messages between data points. *Science*, 315:972–976.

Goldwater, Sharon and Tom Griffiths. 2007. A fully bayesian approach to unsupervised part-of-speech tagging. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 744–751, Prague, Czech Republic, June. Association for Computational Linguistics.

Haghighi, Aria and Dan Klein. 2006. Prototype-driven learning for sequence models. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 320–327.

Headden, III, William P., David McClosky, and Eugene Charniak. 2008. Evaluating unsupervised part-of-speech tagging for grammar induction. In *COLING '08: Proceedings of the 22nd International Conference on Computational Linguistics*, pages 329–336, Morristown, NJ, USA. Association for Computational Linguistics.

Headden III, William P., Mark Johnson, and David McClosky. 2009. Improving unsupervised dependency parsing with richer contexts and smoothing. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 101–109, Boulder, Colorado, June. Association for Computational Linguistics.

Hernando, D., V. Crespi, and G. Cybenko. 2005. Efficient computation of the hidden markov model entropy for a given observation sequence. volume 51, pages 2681–2685.

Huang, Zhongqiang, Mary Harper, and Wen Wang. 2007. Mandarin part-of-speech tagging and discriminative reranking. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages

1093–1102, Prague, Czech Republic, June. Association for Computational Linguistics.

Johnson, Mark. 2007. Why doesn't EM find good HMM POS-taggers? In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 296–305, Prague, Czech Republic, June. Association for Computational Linguistics.

Kaji, Nobuhiro and Masaru Kitsuregawa. 2008. Using hidden markov random fields to combine distributional and pattern-based word clustering. In *COLING '08: Proceedings of the 22nd International Conference on Computational Linguistics*, pages 401–408, Morristown, NJ, USA. Association for Computational Linguistics.

Klein, Dan and Christopher Manning. 2004. Corpus-based induction of syntactic structure: Models of dependency and constituency. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04), Main Volume*, pages 478–485, Barcelona, Spain, July.

Meilă, Marina. 2007. Comparing clusterings—an information based distance. volume 98, pages 873–895.

Rabiner, Lawrence R. 1989. A tutorial on hidden markov models and selected applications in speech recognition. In *Proceedings of the IEEE*, pages 257–286.

Rosenberg, Andrew and Julia Hirschberg. 2007. V-measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 410–420.

Siddiqi, Sajid, Geoffrey Gordon, and Andrew Moore. 2007. Fast state discovery for hmm model selection and learning. In *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics (AI-STATS)*.

Strehl, Alexander and Joydeep Ghosh. 2003. Cluster ensembles — a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*, 3:583–617.

Tseng, Huihsin, Daniel Jurafsky, and Christopher Manning. 2005. Morphological features help pos tagging of unknown words across language varieties. pages 32–39.

Van Gael, Jurgen, Yunus Saatci, Yee Whye Teh, and Zoubin Ghahramani. 2008. Beam sampling for the infinite hidden markov model. In *ICML '08: Proceedings of the 25th international conference on Machine learning*.

Van Gael, Jurgen, Andreas Vlachos, and Zoubin Ghahramani. 2009. The infinite HMM for unsupervised PoS tagging. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 678–687, Singapore, August. Association for Computational Linguistics.

Wang, Qin Iris and Dale Schuurmans. Improved estimation for unsupervised part-of-speech tagging. page 2005, Wuhan, China.

Weeds, Julie and David Weir. 2003. A general framework for distributional similarity. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, pages 81–88, Morristown, NJ, USA. Association for Computational Linguistics.

Zelling, Harris. 1968. *Mathematical sturcture of language*. NewYork:Wiley.

APPENDIX

Pseudo-code of the extended Baum-Welch Algorithm in our dynamic HMM

Input: Time step t:
State Candidate: $k \rightarrow (k^{(1)}, k^{(2)})$;
Sate Number: N^t;
Model Parameter: $\theta^{(t)} = (A^{(t)}, B^{(t)}, \pi^{(t)})$;

Initialize
$u^{(l)}[k^{(1)}, k^{(2)}] \leftarrow [\frac{u^{(l)}[k]}{2}, \frac{u^{(l)}[k]}{2}], l \in \{A, B, \pi\}$
$\pi_{k^{(1)}} \leftarrow \frac{1}{2}\pi_k; \pi_{k^{(2)}} \leftarrow \frac{1}{2}\pi_k$
$a_{\bar{k}k^{(i)}} \leftarrow \frac{1}{2}a_{\bar{k}k^{(i)}}; a_{k^{(i)}\bar{k}} \leftarrow a_{k^{(i)}\bar{k}}$;
$a_{k^{(i)}k^{(j)}} \leftarrow \frac{1}{2}a_{k^{(i)}k^{(j)}}$, here $i, j \in 1, 2, \bar{k} \neq k$

repeat
E step:
update forward: $\alpha_t(k^{(1)})$ and $\alpha_t(k^{(2)})$
backward: $\beta_t(k^{(1)})$ and $\beta_t(k^{(2)})$
update $\xi_t(i, j)$ and $\gamma_t(i)$; if $i, j \in \{k^{(1)}, k^{(2)}\}$
update $\mathbb{E}[n_{ij}] = \sum_t \xi_t(i, j) / \sum_t \gamma_t(i)$
$\mathbb{E}[n_{ik}] = \sum_{t, w_t = k} \gamma_t(j) / \sum_t \gamma_t(j)$
M step:
update $\theta^{(t+1)}$ using equation (7)

until $(\triangle \mathcal{F} < \varepsilon)$

Output: $\theta^{(t+1)}, \mathcal{F}$
