Unsupervised Part of Speech Tagging Using Unambiguous Substitutes from a Statistical Language Model

Mehmet Ali Yatbaz
Dept. of Computer Engineering
Koç University
myatbaz@ku.edu.tr

Deniz Yuret
Dept. of Computer Engineering
Koç University
dyuret@ku.edu.tr

Abstract

We show that unsupervised part of speech tagging performance can be significantly improved using likely substitutes for target words given by a statistical language model. We choose unambiguous substitutes for each occurrence of an ambiguous target word based on its context. The part of speech tags for the unambiguous substitutes are then used to filter the entry for the target word in the word–tag dictionary. A standard HMM model trained using the filtered dictionary achieves 92.25% accuracy on a standard 24,000 word corpus.

1 Introduction

We define the unsupervised part-of-speech (POS) tagging problem as predicting the correct part-of-speech tag of a word in a given context using an unlabeled corpus and a dictionary with possible word–tag pairs[0] The performance of an unsupervised POS tagging system depends highly on the quality of the word–tag dictionary (Banko and Moore, 2004). We propose a dictionary filtering procedure based on likely substitutes suggested by a statistical language model. The procedure reduces the word–tag dictionary size and leads to significant improvement in the accuracy of the POS models.

Probabilistic models such as the hidden Markov model (HMM) trained by expectation maximization (EM), maximum a posteriori (MAP) estimation, and Bayesian methods have been used to solve the unsupervised POS tagging problem (Merialdo, 1994; Goldwater and Griffiths, 2007). All of these approaches first learn the parameters connecting the hidden structure to the observed sequence of variables and then identify the most probable values of the hidden structure for a given observed sequence. They differ in the way they estimate the model parameters. HMM-EM estimates model parameters by using the maximum likelihood estimation (MLE), MAP defines a prior distribution over parameters and finds the parameter values that maximize the posterior distribution given data, and Bayesian methods integrate over the posterior of the parameters to incorporate all possible parameter settings into the estimation process. Some baseline results and performance reports from the literature are presented in Table 1.

(Johnson, 2007) criticizes the standard EM based HMM approaches because of their poor performance on the unsupervised POS tagging and their tendency to assign equal number of words to each hidden state. (Mitzenmacher, 2004) further claims that words have skewed POS tag distributions, and a Bayesian method with sparse priors over the POS tags may perform better than HMM estimated with EM. (Goldwater and Griffiths, 2007) uses a fully Bayesian HMM model that averages over all possible parameter values. Their model achieves 86.8% tagging accuracy with sparse POS priors and outperforms 74.50% accuracy of the standard second order HMM-EM (3-gram tag model) on a 24K word subset of the Penn Treebank corpus. (Smith and Eisner, 2005) take a different approach and use the conditional random fields estimated using contrastive estimation which outperforms the HMM-EM and

[0]In the POS literature the term "unsupervised" is typically used to describe systems that do not directly use the tagged data. However, many of the unsupervised systems, including ours, uses the tag–word dictionary.

Coling 2008: Poster Volume, pages 1391–1398,
Beijing, August 2010

Accuracy	System
64.2	Random baseline
74.4	Second order HMM
82.0	First order HMM
86.8	Fully Bayesian approach with sparse priors (Goldwater and Griffiths, 2007)
88.6	CRF/CE (Smith and Eisner, 2005)
91.4	EM-HMM with language specific information, good initialization and manual adjustments to standard dictionary (Goldberg et al., 2008)
91.8	Minimized models for EM-HMM with 100 random restarts (Ravi and Knight, 2009).
94.0	Most frequent tag baseline

Table 1: Tagging accuracy on a 24K-word corpus. All the systems – except (Goldwater and Griffiths, 2007) – use the same 45 tag dictionary that is constructed from the Penn Treebank.

Bayesian methods by achieving 88.6% accuracy on the same 24K corpus.

Despite the fact that HMM-EM has a poor reputation in POS literature (Goldberg et al., 2008) has shown that with good initialization together with some language specific features and language dependent constraints HMM-EM achieves 91.4% accuracy. Aside from the language specific information and the good initialization, they also manually reduce the noise in the word–tag dictionary.

(Ravi and Knight, 2009) focus on the POS tag collection to find the smallest POS model that explain the data. They apply integer programming to construct a minimal bi-gram POS tag set and use this set to constrain the training phase of the EM algorithm. The model trained by EM is used to reduce the dictionary and these steps are iteratively repeated until no further improvement is observed. Their model achieves 91.6% accuracy on the 24K word corpus (with 100 random starts this goes up to 91.8%). The main advantage of this model is the restriction of the tag set so that rare POS tags or the noise in the corpus do not get incorporated into the estimation process.

Language models for disambiguation: Recent work has shown that statistical language models trained on large amounts of unlabeled text can be used to improve the performance on various disambiguation problems. The language model is used to generate likely substitutes for the target word in the given context and these benefit the disambiguation process to the extent that the likely substitutes are unambiguous or have different ambiguities compared to the target word. Using statistical language models based on large corpora for unsupervised word sense disambiguation and lexical substitution has been explored in (Yuret, 2007; Hawker, 2007; Yuret and Yatbaz, 2010). Unsupervised morphological disambiguation in agglutinative languages using likely substitutes has been shown to improve on standard methods in (Yatbaz and Yuret, 2009).

In this paper we use the statistical language model to reduce the possible number of tags per word to help the disambiguation process. Specifically we assume that the same hidden tag sequence that has generated a particular test sentence can also generate artificial sentences where one of the words has been replaced with a likely substitute. POS tags of the likely substitutes can then be used to reduce the tag set of the target word. Thus, the substitutes are implicitly incorporated into the disambiguation process for reducing the noise and the rare tags in the dictionary.

Currency gyrations can **whipsaw(VB/NN)** the funds .
Currency gyrations can **withdraw(VB)** the funds .
Currency gyrations can **restore(VB)** the funds .
Currency gyrations can **modify(VB)** the funds .
Currency gyrations can **justify(VB)** the funds .
Currency gyrations can **regulate(VB)** the funds .

Table 2: Sample artificial sentences generated for a test sentence from the Penn Treebank.

Table 2 presents an example where the likely unambiguous replacements of the target word "whipsaw" for a given sentence taken from the Penn Treebank (Marcus et al., 1994) are listed. In this example each substitute is an unambiguous verb (VB), confirming our assumption that each artificial sentence comes from the same hidden sequence. For all occurrences of the word "whipsaw", our reduction algorithm will count the POS tags of the likely substitutes and remove the tags

that have not been observed from the dictionary. Assuming that the first sentence in Table 2 is the only sentence in which we observe "whipsaw", the "NN" tag of "whipsaw" will be removed.

The next section describes the details of our dictionary reduction method. Section 3 explains the details of statistical language model. We experimentally demonstrate that the word–tag dictionary reduced by the substitutes improve the performance by constraining the unsupervised model in Section 4. Finally, Section 5 comments on the results and discusses the possible extensions of our method.

2 Dictionary Reduction

Our main assumption is that likely replacements of a target word should have the same POS tag as the target word in a given context. Motivated by this idea we propose a new procedure that automatically reduces the dictionary size by using the unambiguous replacements of the target word. For all occurrences of the target word the procedure counts the POS tags of the replacement words and removes the unobserved POS tags of the target word from the dictionary.

Our approach is based on the idea that similar words in a given context should have the same tag sequence. To reduce the dictionary with the help of the replacement words similar to a target word w, we follow three rules:

1. Choose the replacement word from unambiguous substitutes that are likely to appear in the target word context.

2. Substitutes must be observed in the training corpus.

3. Count the tags of the replacement for all occurrences of the target word.

4. Remove the tags that are not observed as the tag of replacements in any occurrences of the target word.

The first rule is used to increase the likelihood of getting a replacement word with the same POS tag. The second rule makes sure that the size of the vocabulary does not change. The third rule determines the unused POS tags in all occurrences of w and finally, last rule removes the unobserved tags of w from the dictionary.

We use the standard first order HMM to test the performance of our method. In a standard n^{th} order HMM each hidden state is conditioned by its n preceding hidden states and each observation is conditioned by its corresponding hidden state. In POS tagging, the observed variable sequence is a sentence s and the hidden variables t_i are the POS tags of the words w_i in s. The HMM parameters θ can be estimated by using Baum-Welch EM algorithm on an unlabeled training corpus D (Baum, 1972). The tag sequence that maximizes $\Pr(t|s,\hat{\theta})$ can be identified by the Viterbi search algorithm.

3 Statistical Language Modeling

In order to estimate highly probable replacement words for a given word w in the context c_w, we use an n-gram language model. The context is defined as the $2n-1$ word window $w_{-n+1}\dots w_0\dots w_{n-1}$ and it is centered at the target word position. The probability of a word in a given context can be estimated as:

$$\begin{aligned} P(w_0 = w|c_w) &\propto P(w_{-n+1}\dots w_0\dots w_{n-1}) && (1)\\ &= P(w_{-n+1})P(w_{-n+2}|w_{-n+1}) \\ &\quad \dots P(w_{n-1}|w_{-n+1}^{n-2}) && (2)\\ &\propto P(w_0|w_{-n+1}^{-1})P(w_1|w_{-n+2}^{0}) \\ &\quad \dots P(w_{n-1}|w_0^{n-2}) && (3)\end{aligned}$$

where w_i^j represents the sequence of words $w_i w_{i+1}\dots w_j$. In Equation 1, $\Pr(w|c_w)$ is proportional to $\Pr(w_{-n+1}\dots w_0\dots w_{n+1})$ since the context of the target word replacements is fixed. Terms without w_0 are common for every replacement in Equation 2 therefore they have been dropped in Equation 3. Finally, because of the Markov property of n-gram language model, only $n-1$ words are used as a conditional context.

The probabilities in Equation 3 are estimated using a 4 gram language model for all the words in the vocabulary of D that are unambiguous and have a common tag with the target word w. The words with the highest $\Pr(r|c_w)$ where $r \in D$ are selected as the replacement words of w in c_w.

To get accurate domain independent probability estimates we used the Web 1T data-set (Brants and Franz, 2006), which contains the counts of word sequences up to length five in a 10^{12} word corpus derived from publicly accessible Web pages. The SRILM toolkit is used to train 5-gram language model (Stolcke, 2002). The language model parameters are optimized by using a randomly selected 24K words corpus from Penn Treebank. In order to efficiently apply the language model to a given test corpus, the vocabulary size is limited to the words seen in the test corpus.

4 Experiments

In this section we present a number of experiments measuring the performance of several variants of our algorithm. The models in this section are trained[1] and tested on the same unlabeled data therefore there aren't any out-of-vocabulary words. The experiments in this section focus on: (1) the analysis of the dictionary reduction (2) the number of the substitutes used for each ambiguous word and (3) the size of the word–tag dictionary.

4.1 Dataset

We trained HMM-EM models on a corpus that consists of the first 24K words of the Penn Treebank corpus. To be consistent with the POS tagging literature, the tag dictionary is constructed by listing all observed tags for each word in the entire Penn Treebank. Nearly 55% of the words in Penn Treebank corpus are ambiguous and the average number of tags is 2.3.

Groups	Member POS tags	Count	%
Noun	NN/NNP/NNS/NNPS	7511	31.30
Verb	VBD/VB/VBZ/VBN/VBG/VBP	3285	13.69
Adj	JJ/JJR/JJS	1718	7.16
Adv	RB/RBR	742	3.09
Pronoun	CD/PRP/PRP$	1397	5.82
Content	Noun/Verb/Adj/Adv/Pronoun	14653	61.05
Function	Other	9347	38.95
Total	All 45 POS tags	24K	100.00

Table 3: Group names, members, number and percentage of the words according to their gold POS tags.

[1]The GMTK tool is used to train HMM-EM model on an unlabeled corpus (Bilmes and Zweig, 2002).

Table 3 shows the POS speech groups and their distributions in the 24K word corpus. We report the model accuracy on several POS groups. Our motivation is to determine HMM-EM model accuracies on the subgroups before and after implementing the dictionary reduction procedure.

4.2 Baseline

Table 4 presents some standard baselines for comparison. We define a random and a most frequent tag (MFT) baseline on the 24K corpus. The random baseline is calculated by randomly picking one of the tags of each word and it also represents the amount of ambiguity in the corpus. The MFT baseline simply selects the most frequent POS tag of each word from the 1M word Penn Treebank corpus (counts of the first 24K words is not included in the 1M word corpus). If the target word does not exist in the training set, the MFT baseline randomly picks one of the possible tags of the missing word.

The first and second order HMMs can be treated as the unsupervised baselines. These unsupervised baselines are calculated by training uniformly initialized first and second order HMMs on the target corpus without any smoothing. All the initial parameters of HMM-EM are uniformly initialized to observe only the effects of the artificial sentences on the performance of HMM-EM.

The success of the MFT baseline on the $Noun$, Adj, $Pronoun$ and function word groups shows that tag distributions of the words in these groups are more skewed towards to one of the available tags. The MFT baseline performs poorly, compared to the above groups, on $Verb$, and Adv which is due to the less skewed POS tag behavior of these tags.

The POS tagging literature widely uses the second order HMM as the baseline model; however, the performance of this model can be outperformed by an unsupervised first order HMM model or a simple MFT baseline as presented in Table 4. A point worth noting is that although the first order HMM and the MFT baseline have similar content word accuracies, the MFT baseline is significantly better on the function words. This is expected since EM tends to assign words uniformly to the available POS tags. Thus EM can

	Noun	Verb	Adj	Adv	Pronoun	Content	Function	Total(%)
Random Baseline	76.98	53.87	68.46	72.98	87.64	71.59	52.64	64.21
3-gram HMM	77.43	68.16	78.06	73.32	94.85	76.88	70.45	74.38
2-gram HMM	92.22	83.84	85.22	83.96	95.56	89.42	70.49	82.05
MFT Baseline	96.11	80.30	88.56	83.15	98.75	91.28	98.25	93.99

Table 4: Percentages of words tagged correctly by different models using standard dictionary.

not capture the highly skewed behavior of function words. Moreover the amount of skewness affects the accuracy of the EM such that the performance gain of the MFT baseline over the first order HMM on function words is around 28%-30% while the performance gain on $Noun$, Adj and $Pronoun$ is around 3%-4%.

4.3 Reduced Dictionary

EM can not capture the sparse structure of the word distributions therefore it tends to assign equal number of words to each POS tag. Together with the noisy word–tag dictionary great portion of the function words are tagged with very rare POS tags. The abuse of the rare tags is presented in Table 5 in a similar fashion with (Ravi and Knight, 2009). The count of replacement word POS tags and the removed rare POS tags of 2 erroneous function words are also shown in Table 5.

Word	Tag dictionary	Gold tagging	EM tagging	Replacement POS counts
of	{RB,RP,IN}	IN(632)	IN(0)	IN(2377)
		RP(0)	RP(632)	**RP(0)**
		RB(0)	RB(0)	RB(850)
a	{LS,SYM,NNP,	DT(458)	DT(0)	DT(513)
	FW,JJ,IN,DT}	IN(1)	IN(0)	IN(317)
		JJ(2)	JJ(0)	JJ(1329)
		SYM(1)	SYM(258)	**SYM(0)**
		LS(0)	LS(230)	**LS(0)**

Table 5: Removed POS tags of the given words are shown in bold.

The results obtained with the dictionary that is reduced by using 5 replacements are presented in Table 6. Note that with reduced dictionary the uniformly initialized first order HMM-EM achieves 91.85% accuracy. Dictionary reduction also removes some of the useful tags therefore the upper–bound (oracle score) of the 24K dataset becomes 98.15% after the dictionary reduction. We execute 100 random restarts of the EM algorithm and select the model with the highest corpus likelihood, our model achieves 92.25% accuracy which is the highest accuracy reported for the 24K corpus so far.

As Table 6 shows, the effect of the dictionary reduction on the function words is higher than the effect on the content words. The main reason for this situation is, function words are frequently tagged with one of its tags which is also the reason for the high accuracy of the majority voting based baseline on the function words.

The reduced dictionary (RD) removes the rare problematic POS tags of the words as a result the accuracy on the content and function words shows a drastic improvement compared to HMM models trained with the original dictionary.

Pos groups	2-gram HMM accuracy(%)	2-gram HMM RD accuracy(%)
Noun	92.22	94.01
Verb	83.84	84.90
Adj	85.22	89.52
Adv	83.96	85.18
Pronoun	95.56	95.92
Content	89.42	91.18
Function	70.49	92.92
All	82.05	91.85

Table 6: Percentages of the correctly tagged words by different models with modified dictionary. The dictionary size is reduced by using the top 5 replacements of each target word.

4.4 More Data

In this set of experiments we doubled the size of the data and trained HMM-EM models on a corpus that consists of the first 48K words of the Penn Treebank corpus. Our aim is to observe the effect of more data on our dictionary reduction proce-

dure. Using the 5 replacements of each ambiguous word we reduce the dictionary and train a new HMM-EM model using this dictionary. The additional data together with 100 random starts increases the model accuracy to 92.47% on the 48K corpus.

Pos groups	3-gram HMM RD accuracy(%)	2-gram HMM RD accuracy(%)
Noun	89.45	93.47
Verb	85.56	88.99
Adj	86.02	87.53
Adv	94.44	95.92
Pronoun	94.08	94.04
Content	88.91	91.97
Function	92.44	92.26
All	90.31	92.09

Table 7: Percentages of the correctly tagged words by the first and second order HMM-EM model trained on the 48K corpus with reduced dictionary. The dictionary size is reduced by using the top 5 replacements of each target word.

As we mentioned before, when the model is trained using the original dictionary, the performance gap between the first order HMM the second order HMM is around 8% as presented in Table 4. On the other hand, when we use the reduced dictionary together with more data the accuracy gap between the second order and the first order HMM-EM becomes less than 2% as shown in Table 7. This confirms the hypothesis that the low performance of the second order HMM is due to data sparsity in the 24K-word dataset, and better results may be achieved with the second order HMM in larger datasets.

4.5 Number of Replacements

In this set of experiments we vary the number of artificial replacement words per each ambiguous word in s. We run our method on the 24K corpus with 1, 5, 10, 25 and 50 replacement words per ambiguous word and we present the results in Table 8. The performance of our method affected by the the number of replacements and highest score is achieved when 5 replacements are used. Incorporating the probability of the substitutes into the model rather than using a hard cutoff may be a better solution.

Number of replacements	2-gram HMM RD accuracy(%)
none	82.05
1	89.65
5	**91.85**
10	90.09
25	89.97
50	89.83

Table 8: Percentages of the correctly tagged words by the models trained on the 24K corpus with different reduced dictionaries. The dictionary size is reduced by using different number replacements.

4.6 17-Tagset

To observe the effect our method on a model with coarse grained dictionary, we collapsed the 45–tagset treebank dictionary to a 17–tagset coarse dictionary (Smith and Eisner, 2005). The POS literature after the work of Smith and Eisner follows this tradition and also tests the models on this 17–tagset. Table 9 summarizes the previously reported results on coarse grained POS tagging. Our system achieves 92.9% accuracy where the oracle accuracy of 24K dataset with the reduced 17–tagset dictionary is 98.3% and the state-of-the-art system IP+EM scores 96.8%.

Model	Accuracy	Data Size
BHMM	87.3	24K
CE+spl	88.7	24K
RD	**92.9**	24K
LDA+AC	93.4	1M
InitEM-HMM	93.8	1M
IP+EM	96.8	24K

Table 9: Performance of different systems using the coarse grained dictionary.

The IP+EM system constructs a model that describes the data by using minimum number of bigram POS tags then uses this model to reduce the dictionary size (Ravi and Knight, 2009). InitEM-HMM uses the language specific information together with good initialization and it achieves 93.8% accuracy on the 1M word treebank corpus. LDA+AC semi-supervised Bayesian model with strong ambiguity class component given the morphological features of words and scores 93.4% on the 1M word treebank corpus. (Toutanova and Johnson, 2007). CE+spl is HMM model estimated

by contrastive estimation method and achieves 88.7% accuracy (Smith and Eisner, 2005). Finally, BHMM is a fully Bayesian approach that uses sparse POS priors and scores 87.3% (Goldwater and Griffiths, 2007).

5 Contributions

In this paper we proposed a dictionary reduction method that can be applied to unsupervised tagging problems. With the help of a statistical language model, our system creates artificial replacements that are assumed to have the same POS tag as the target word and use them to reduce the size of the word–tag dictionary. To test our method we used HMM-EM as the unsupervised model. Our method significantly improves the prediction accuracy of the unsupervised first order HMM-EM system in all of the POS groups and achieves 92.25% and 92.47% word tagging accuracy on the 24K and 48K word corpora respectively. We also tested our model on a coarse grained dictionary with 17 tags and achieved an accuracy of 92.8%.

In this work, we show that unambiguous replacements of an ambiguous word can reduce the amount of the ambiguity thus replacement words might also be incorporated into the other unsupervised disambiguation problems.

Acknowledgments

This work was supported in part by the Scientific and Technical Research Council of Turkey (TÜBİTAK Project 108E228).

References

Banko, Michele and Robert C. Moore. 2004. Part of speech tagging in context. In *COLING '04: Proceedings of the 20th international conference on Computational Linguistics*, page 556, Morristown, NJ, USA. Association for Computational Linguistics.

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

Bilmes, J. and G. Zweig. 2002. The Graphical Models Toolkit: An open source software system for speech and time-series processing. In *IEEE International Conference On Acoustics Speech and Signal Processing*, volume 4, pages 3916–3919.

Brants, T. and A. Franz. 2006. Web 1T 5-gram Version 1. *Linguistic Data Consortium, Philadelphia*.

Goldberg, Y., M. Adler, and M. Elhadad. 2008. Em can find pretty good hmm pos-taggers (when given a good start). *Proceedings of ACL-08. Columbus, OH*, pages 746–754.

Goldwater, S. and T. Griffiths. 2007. A fully Bayesian approach to unsupervised part-of-speech tagging. In *Annual Meeting-Assosiation for Computational Linguistics*, volume 45, page 744.

Hawker, Tobias. 2007. Usyd: Wsd and lexical substitution using the web1t corpus. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 446–453, Prague, Czech Republic, June. Association for Computational Linguistics.

Johnson, M. 2007. Why doesnt 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.

Marcus, M.P., B. Santorini, and M.A. Marcinkiewicz. 1994. Building a large annotated corpus of English: The Penn Treebank. *Computational linguistics*, 19(2):313–330.

Merialdo, B. 1994. Tagging english text with a probabilistic model. *Computational linguistics*, 20(2):155–171.

Mitzenmacher, M. 2004. A brief history of generative models for power law and lognormal distributions. *Internet mathematics*, 1(2):226–251.

Ravi, Sujith and Kevin Knight. 2009. Minimized models for unsupervised part-of-speech tagging. In *ACL-IJCNLP '09: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1*, pages 504–512, Morristown, NJ, USA. Association for Computational Linguistics.

Smith, Noah A. and Jason Eisner. 2005. Contrastive estimation: training log-linear models on unlabeled data. In *ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 354–362, Morristown, NJ, USA. Association for Computational Linguistics.

Stolcke, A. 2002. SRILM-an extensible language modeling toolkit. In *Seventh International Conference on Spoken Language Processing*, volume 3.

Toutanova, K. and M. Johnson. 2007. A Bayesian LDA-based model for semi-supervised part-of-speech tagging. In *Proceedings of NIPS*, volume 20.

Yatbaz, Mehmet Ali and Deniz Yuret. 2009. Unsupervised morphological disambiguation using statistical language models. In *NIPS 2009 Workshop on Grammar Induction, Representation of Language and Language Learning.*

Yuret, Deniz and Mehmet Ali Yatbaz. 2010. The noisy channel model for unsupervised word sense disambiguation. *Computational Linguistics*, 36(1), March.

Yuret, Deniz. 2007. KU: Word sense disambiguation by substitution. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 207–214, Prague, Czech Republic, June. Association for Computational Linguistics.