Simple Semi-Supervised POS Tagging

Karl Stratos **Michael Collins**

Columbia University, New York, NY 10027, USA
{stratos, mcollins}@cs.columbia.edu

Abstract

We tackle the question: how much supervision is needed to achieve state-of-the-art performance in part-of-speech (POS) tagging, if we leverage lexical representations given by the model of Brown et al. (1992)? It has become a standard practice to use automatically induced "Brown clusters" in place of POS tags. We claim that the underlying sequence model for these clusters is particularly well-suited for capturing POS tags. We empirically demonstrate this claim by drastically reducing supervision in POS tagging with these representations. Using either the bit-string form given by the algorithm of Brown et al. (1992) or the (less well-known) embedding form given by the canonical correlation analysis algorithm of Stratos et al. (2014), we can obtain 93% tagging accuracy with just 400 labeled *words* and achieve state-of-the-art accuracy ($> 97\%$) with less than 1 percent of the original training data.

1 Introduction

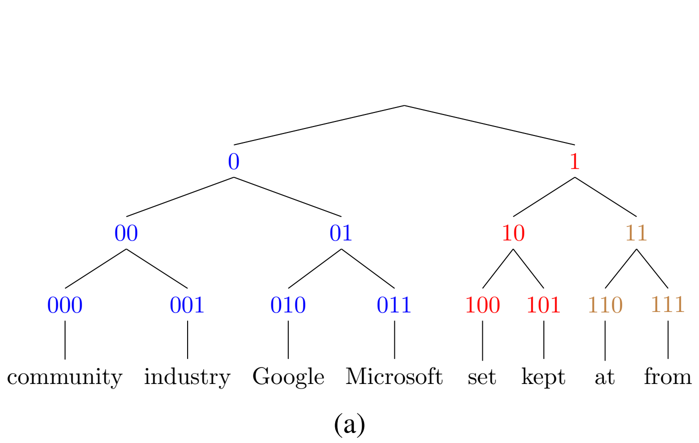
While fully supervised POS tagging is largely considered a solved problem today, this is hardly the case for unsupervised POS tagging. Despite much previous work (Smith and Eisner, 2005; Johnson, 2007; Toutanova and Johnson, 2007; Haghighi and Klein, 2006; Berg-Kirkpatrick et al., 2010), results on this task are complicated by varying assumptions and unclear evaluation metrics (Christodoulopoulos et al., 2010). Perhaps most importantly, they are not good enough to be practical. Even with indirect supervision, for example the prototype-driven method of Haghighi and Klein (2006) which assumes a set of word examples for each tag type, the best per-position accuracy remains in the range of mid-70%.

Recent work has taken a middle ground between fully supervised and unsupervised setups by exploiting existing resources, for example by projecting POS tags from a supervised language or using tag dictionaries (Das and Petrov, 2011; Li et al., 2012; Täckström et al., 2013).

In this work, we focus on *minimizing* the amount of labeled data required to obtain a good POS tagger. The key to our approach is the use of lexical representations induced by the clustering model of Brown et al. (1992). We argue that this model is particularly appropriate for representing POS tags given their nearly deterministic nature (Section 2). This sheds light on why the representations derived under this model reveal the underlying POS tag information of words.

We empirically demonstrate the validity of our observation by using these representations to drastically reduce the number of training examples required for good POS tagging performance on English, German, and Spanish newswire datasets. For instance, on the 12-tag English dataset, we obtain tagging accuracy of 93% with just 400 labeled *words*. We obtain tagging accuracy of 97.03% (about a half percent behind fully supervised models) with just 0.74% of the original training data.

Our aim is orthogonal to the discussion in Manning (2011) who investigates what is needed to go beyond the current state-of-the-art POS tagging performance. Our focus is on reaching that performance with as little supervision as possible.



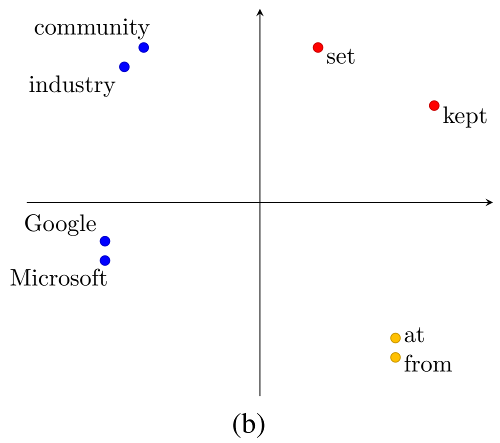


Figure 1: Two representational schemes under the Brown model. (a) Bit-string representations: the path from the root to a word is encoded as a bit-string. (b) CCA vector representations.

2 Motivation

2.1 POS tags are almost deterministic

Certain words are genuinely ambiguous (e.g., `set` can be a verb or a noun): this was the motivation of the use of statistical models in the early days of computational linguistics (Church, 1988). However, it is also true that many words are deterministically mapped to correct POS tags (e.g., `the` is always a determiner). A simple experiment highlights this property. Let **count**(w, t) be the number of times word w is tagged as t in the training data (likewise, **count**(w) and **count**(t) are counts of word w and tag t). Define a deterministic mapping $f : w \mapsto t$ from words to tags as

$$f(w) = \begin{cases} \arg\max_t \textbf{count}(w,t) & \text{if } \textbf{count}(w) > 0 \\ \arg\max_t \textbf{count}(t) & \text{otherwise} \end{cases}$$

In our datasets, this naïve procedure in fact yields reasonable tagging accuracies: 92.22% for coarse tags and 88.50% for fine-grained tags (averaged across three languages).

This observation suggests that the following restricted class of HMMs might be sufficient for modeling the characteristics of POS tags:

- $\pi(t)$ is the prior probability of tag type t.
- $t(t'|t)$ is the probability of transitioning from tag type t to tag type t'.
- $o(w|t)$ is the probability of emitting word type w from tag type t.
- (Restriction) For each word type w, we have $o(w|t) > 0$ only for a *unique* tag type t and $o(w|t') = 0$ for all other tag types $t' \neq t$.

In other words, we assume that tag types partition word types while imposing a first-order sequence structure on tag types.

2.2 Brown et al. (1992) model

This class of restricted HMMs is precisely the model proposed by Brown et al. (1992)—henceforth the Brown model. A popular use of this model is agglomerative word clustering: the result is a hierarchy over word types, such as the one shown in Figure 1(a). In practice, each word type is represented as a bit-string indicating the path from the root. These bit-strings have been used as discrete (binary) features in various natural language tasks such as named-entity recognition (Miller et al., 2004) and dependency parsing (Koo et al., 2008).

Recently, Stratos et al. (2014) showed that a variant of canonical correlation analysis (CCA) (Hotelling, 1936) can be used to provably recover the clusters under the Brown model. Under this method, each word is represented as an m-dimensional vector where m is the number of hidden states in the model: see Figure 1(b) for illustration. This can be used as m real-valued features in discrminative models. Note that real-valued representations can reflect ambiguity (e.g., `set` in the illustration) which can be seen as a benefit over discrete representations.

By the above observation, we conjecture that the hidden states of the Brown model capture POS tags. Then the bit-string and embedding representations essentially "give away" the POS tag associated with a word. In experiments, we show that this is indeed the case and not far removed from the idealized illustration in Figure 1.

3 Method

In this section, we describe our tagging framework MINITAGGER, which is pleasantly simple but surprisingly effective. It uses an off-the-shelf discriminative classifier to map a word's context to a POS tag. Concretely, given a sentence $\boldsymbol{x}$ and a position i in $\boldsymbol{x}$, we extract a feature vector $\phi(\boldsymbol{x}, i) \in \mathbb{R}^d$ and train a multi-class classifier to map $\phi(\boldsymbol{x}, i)$ to a POS tag. To clarify, this is *not* the the HMM model described in the previous section: the HMM model underlies Brown bit-strings and CCA embeddings.

This framework has compelling benefits. First, it allows for learning from partially labeled sentences since each word is an independent sample. Second, training and tagging can be very fast since they do not involve dynamic programming required for structured models. Third, arbitrary features can be easily and effectively incorporated. Finally, there are many well-oiled public implementations of discriminative classifiers such as support-vector machines (SVMs), thus building an efficient and effective MINITAGGER takes only a minimal effort.

3.1 Feature templates

We use a baseline feature function **base** that maps a sentence-position pair $(\boldsymbol{x}, i)$ to a a 0-1 vector **base**$(\boldsymbol{x}, i)$ indicating

- Identities of $x_{i-1}, x_{i+1}, x_i, x_{i-2}, x_{i+2}$
- Prefixes and suffixes of x_i up to length 4
- Whether x_i is capitalized, numeric, or non-alphanumeric

Let **bit**(x) be a binary vector indicating prefixes of the Brown bit-string corresponding to x.[1] Let **cca**$(x) \in \mathbb{R}^m$ be an m-dimensional CCA embedding corresponding to x.

[1]Past work has used various complex schemes on which prefixes to use, e.g., see Koo et al. (2008). For simplicity, we use every prefix in this work.

$\phi(\boldsymbol{x}, i)$	Definition
BASE	$\mathbf{base}(\boldsymbol{x}, i)$
BIT	$\mathbf{base}(\boldsymbol{x}, i) \oplus \mathbf{bit}(x_i) \oplus \mathbf{bit}(x_{i-1}) \oplus \mathbf{bit}(x_{i+1})$
CCA	$\frac{\mathbf{base}(\boldsymbol{x},i)}{\lvert\lvert\mathbf{base}(\boldsymbol{x},i)\rvert\rvert} \oplus \frac{\mathbf{cca}(x_i)}{\lvert\lvert\mathbf{cca}(x_i)\rvert\rvert} \oplus \frac{\mathbf{cca}(x_{i-1})}{\lvert\lvert\mathbf{cca}(x_{i-1})\rvert\rvert} \oplus \frac{\mathbf{cca}(x_{i+1})}{\lvert\lvert\mathbf{cca}(x_{i+1})\rvert\rvert}$

Table 1: Feature templates for MINITAGGER: $\oplus$ is the vector concatenation operation.

Table 1 shows the feature templates we use to obtain a vector representation of $(\boldsymbol{x}, i)$. $\oplus$ is the vector concatenation operation. BASE is a baseline template which uses only the spelling features of the current word and the identities of neighboring words. BIT is the same as BASE but augmented with Brown bit-strings. CCA is the same as BASE but augmented with CCA embeddings with appropriate normalization.

3.2 Sampling methods

3.2.1 Active learning

We also deploy *active learning* to find the most informative words for labeling in attempt to reduce the amount of training data. While there is a rich literature on this topic (see Settles (2010) for a survey), we focus on a simple form of margin sampling in this work. Every time the model is allowed to have an additional label, it actively selects the word from a pool of unannotated words whose predicted tag is the least confident.

To be precise, let $s(\boldsymbol{x}, i, y)$ be the score of label y for sentence-position pair $(\boldsymbol{x}, i)$. For example, a linear SVM may define $s(\boldsymbol{x}, i, y) = w_y^\top \phi(\boldsymbol{x}, i)$ where w_y is the model parameter and $\phi(\boldsymbol{x}, i)$ is a feature template in Table 1. To obtain M labeled examples, we specify the initial seed size $k \leq M$ and the step size ξ (for simplicity, assume ξ divides $M - k$) and proceed as follows:

1. Select the top k most frequent word types (a random occurrence of each type) for labeling.
2. For $(M - k)/\xi$ times:
   (a) Train a model with the current set of labeled examples.

NOUN		VERB		ADP	ADJ	ADV	NUM
community	Microsoft	kept	is	from	romantic	boldly	1
enterprise	AT&T	made	was	between	mystical	selfishly	2
law	Unocal	invented	were	for	macho	frequently	3
anthem	Chrysler	squandered	are	[on	heroic	cynically	4
invader	Hexcel	lent	had	in	straight-ahead	wisely	6
pastime	Tosco	memorized	becomes	towards	piquant	clumsily	7
heritage	Geico	witnessed	‘s	betwen	cushy	effectively	8
curriculum	Starwave	enjoyed	has	betweeen	ghoulish	carefully	5

Table 2: Nearest neighbors of some example words along with their associated universal tags (measuring the l_2 distance between CCA word vectors).

(b) Label ξ examples $(\boldsymbol{x}, i)$ with the smallest "confidence" values $s(\boldsymbol{x}, i, y_1) - s(\boldsymbol{x}, i, y_2)$ where

$$y_1 := \arg\max_{y} s(\boldsymbol{x}, i, y)$$
$$y_2 := \arg\max_{y \neq y_1} s(\boldsymbol{x}, i, y)$$

During active learning, the model operates on unannotated data provided that we supply labels for selected examples (we simulate this setting with labeled data). Note that the selection of examples is inherently tied to the choice of features. Indeed, our experiments show that it is crucial to use lexical representations for active learning to work effectively.

Large values of k and ξ can be used to speed up active learning (possibly at the cost of performance loss). Since our focus is on maximizing performance with minimal supervision, we use $k = \xi = 1$. We leave the speed-performance tradeoff of active learning for future work.

3.2.2 Random and frequent-word sampling

In addition to active learning, we consider the following methods for obtaining M labeled words.

- Random sampling: Select M words uniformly at random (without replacement).
- Frequent-word sampling: Select random occurrences of the M most frequent word types.

Note that frequent-word sampling is optimal if there really is a deterministic mapping from word types to tag types. But since the assumption does not hold perfectly, it has severe limitations in practice. We have found that frequent-word sampling outperforms random sampling for small values of M but starts to lag behind as M increases.

4 Experiments

4.1 Setting

We experimented on 3 languages: English, German, and Spanish. For all these languages, we used the train/dev/test datasets of the universal treebank (McDonald et al., 2013)—both the reduced tagset version, which we denote by EN12, GE12, and SP12, and the original tagset version, which we denote by EN45, GE16, and SP24. The number in the dataset name refers to the number of tag types in that dataset: e.g., EN45 is an English dataset with 45 possible tags.

For each language, we derived Brown representations (by which we mean both the bit-string and embedding forms) from a corpus of unlabeled text. For English, we used a corpus of about 772 million words from various sources of (mostly newswire) text. For German and Spanish, we used the n-gram statistics in Google Ngram (Michel et al., 2011). The number of words was about 64 billion for German and 83 billion for Spanish.

We used the implementation of Liang (2005) to derive bit-string word representations for English: for German and Spanish, we used the agglomerative clustering technique of Stratos et al. (2014) since Liang's implementation did not support operating directly on n-grams. We used the CCA algorithm of Stratos et al. (2014) to derive 50-dimensional word embeddings. Table 2 displays nearest neighbors (i.e., words whose associated CCA embeddings are

# labels	Sampling	Features	EN12	GE12	SP12	EN45	GE16	SP24
200	random	BASE	74.52	72.91	77.47	70.00	60.39	67.01
	frequent-word	BASE	79.42	78.36	77.88	74.58	68.83	73.92
	active	BASE	66.67	71.00	69.90	61.05	61.01	64.60
		BIT	88.53	**81.71**	85.18	79.89	71.18	81.39
		CCA	**89.34**	81.30	**87.63**	**81.68**	**76.60**	**86.54**
400	random	BASE	80.18	76.13	79.96	75.26	69.08	76.66
	frequent-word	BASE	85.44	82.69	79.93	82.07	75.68	80.44
	active	BASE	76.06	77.56	80.69	77.24	67.93	79.29
		BIT	**93.00**	87.53	89.94	88.38	77.82	86.53
		CCA	92.29	**88.17**	**91.70**	**88.55**	**80.48**	**89.06**
1000	random	BASE	85.39	81.31	85.92	81.72	73.61	82.53
	frequent-word	BASE	89.94	85.04	88.93	87.53	79.31	85.93
	active	BASE	85.65	86.27	88.16	85.02	78.54	84.91
		BIT	**95.21**	91.18	93.45	**92.81**	83.91	91.73
		CCA	95.03	**92.48**	**94.37**	92.22	**84.40**	**91.97**

Table 3: Dev performance of MINITAGGER when only 200, 400, and 1000 labeled words are used.

the closest in l_2 distance) of some example words. We see that these embeddings are remarkably good at relating the POS tag information to Euclidean distance, confirming our hypothesis in Section 2.

We built a MINITAGGER using the liblinear package of Fan et al. (2008).[2] We primarily compared our model with conditional random fields (CRFs) (Lafferty et al., 2001). We used the implementation of Okazaki (2007).

While we do not rigorously compare runtime performances in this work, we note that the computational advantages of MINITAGGER are very useful in practice. Training/tagging takes only a few seconds with baseline features; with more complex features such as word embeddings, it still takes much less time than what is required by a CRF (which takes hours with baseline features).

4.2 Effect of Brown representations

4.2.1 With limited supervision

We first look at the effect of using Brown representations when only a limited amount of training data is available: a scenario in which the role of such lexical representations can be prominent. We select a subset of training data (by words) with various sampling schemes described in Section 3.2.

Fixed amount of data. Table 3 shows the performance on the development portion when MINITAGGER is trained on 200, 400, and 1000 labeled words. Active learning together with Brown representations gives dramatic improvement in accuracy when the amount of training data is limited. With 200 randomly sampled labels, the baseline model obtains an average accuracy of 74.97% across EN12, GE12, and SP12. This improves to 86.09% when labels are actively selected with CCA features. A striking result is that we can obtain an accuracy of 93% with only 400 labeled words on the 12-tag English data.

The performance of various sampling methods and features on EN12 is plotted in Figure 2: it is

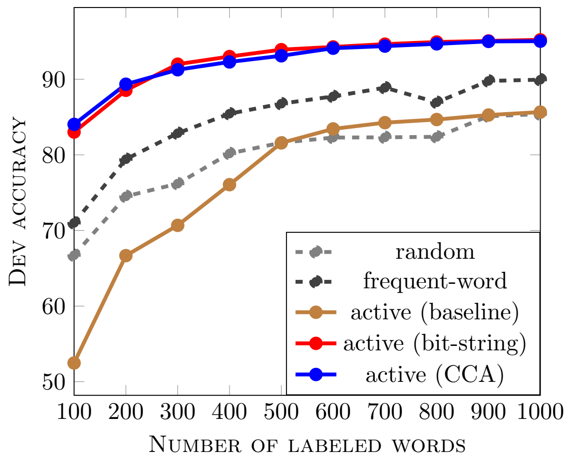
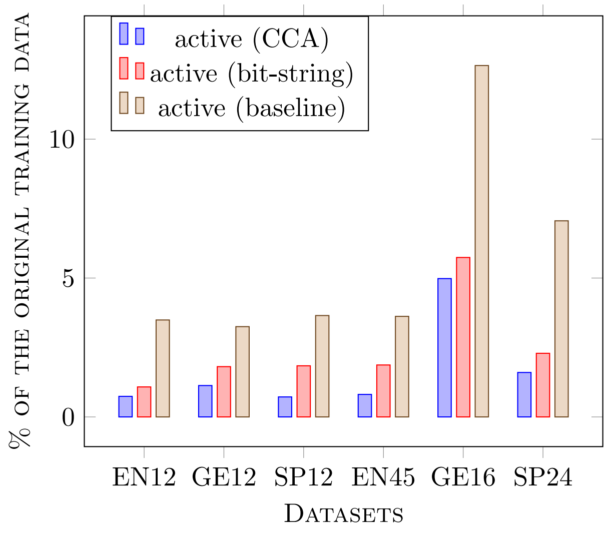


Figure 2: Dev performance of MINITAGGER with varying amounts of supervision (on EN12).

[2]Our code is available at: `https://github.com/karlstratos/minitagger`.

Dataset	Target acc	BASE	BIT	CCA
EN12	97%	33200	10300	**7000**
GE12	94%	8600	4800	**3000**
SP12	96%	13700	6900	**2700**
EN45	96%	34400	17800	**7700**
GE16	92%	33500	15200	**13200**
SP24	95%	26500	8600	**6000**

(a)



(b)

Table 4: (a) Smallest numbers of actively selected labels required by MINITAGGER to reach the target dev performance: 92–97%. The target performance is defined to be the accuracy of the fully supervised baseline rounded down to a whole number (Table 5). (b) As percentages of the original training data.

clear that active sampling with Brown bit-string or CCA embedding features outperforms others consistently and very significantly.

Fixed target accuracy. Table 4(a) shows the smallest numbers of labeled words required to achieve target performance, where the target performance is defined to be the accuracy of the fully supervised baseline rounded down to a whole number (Table 5). We repeatedly increase the training size by 100 and report the first size that allows MINITAGGER to reach the target accuracy. These numbers are presented as percentages of the size of the original training data in Table 4(b).

We see that active learning with lexical representations provides dramatic reduction in training data while maintaining good performance. In all cases, using CCA embeddings as features for active learning outperforms using Brown bit-strings, although sampling takes much longer with CCA embeddings since there are many more non-zero features. MINITAGGER does almost as well as when fully supervised with less than 1% of the training data on English: $>$ 97% accuracy with 0.74% of the data on the 12-tag version, and $>$ 96% accuracy with 0.81% of the data on the 45-tag version.

4.2.2 With full supervision

We also examine the effect of Brown representations in a fully supervised setting. Table 5 shows the performance of different tagging methods on the development portion when all training data is used. We see that Brown representations are helpful even under full supervision: MINITAGGER, a simple greedy model, outperforms CRF when equipped with Brown representations.

Table 5 also shows the performance of MINITAGGER and CRF on the test portion. For MINITAGGER, we additionally consider a model trained only on the actively selected samples with CCA features in Table 4 which are sufficient to reach state-of-the-art performance on the dev portion. As percentages of the original training data, these samples constitute 0.74% for EN12, 1.13% for GE12, 0.72% for SP12, 0.81% for EN45, 4.98% for GE16, and 1.60% for SP24—1.66% on average. The accuracy of MINITAGGER equipped with Brown representations is again generally higher than that of CRF. Furthermore, MINITAGGER achieves competitive performance using only a fraction of the original training set.

Eval	Model	Features	Data	EN12	GE12	SP12	EN45	GE16	SP24
dev	MINI	BASE	all	97.42	94.91	96.75	96.51	92.44	95.42
		BIT	all	97.56	**95.09**	96.98	96.63	92.67	95.89
		CCA	all	**97.67**	94.94	**97.12**	**96.81**	**92.81**	**95.99**
	CRF	default	all	97.37	94.56	96.59	96.59	91.76	95.61
test	MINI	CCA	all	**97.90**	**94.79**	**95.88**	**97.25**	**92.16**	**94.70**
	MINI	CCA	active	97.20 (0.74)	94.09 (1.13)	94.27 (0.72)	96.43 (0.81)	91.59 (4.98)	93.06 (1.60)
	CRF	default	all	97.54	94.33	94.88	97.03	91.12	93.48

Table 5: Performance of MINITAGGER (MINI in short) and CRF on the dev and test portions. The Data column specifies the amount of training data: "all" means all training data is used, "active" means only the labeled examples actively selected (with the same CCA features) in Table 4 are used: the amount of actively selected examples as a percentage of the original training data is shown in parantheses.

5 Related work

We make a few remarks on related works not already discussed earlier. Our work extends a rich body of previous work on reducing annotation efforts with seed examples, unlabeled data, and training example selection (Yarowsky, 1995; Blum and Mitchell, 1998; Collins and Singer, 1999; Miller et al., 2004; Koo et al., 2008; Kim and Snyder, 2013). In particular, Miller et al. (2004) investigate semi-supervised named-entity recognition based on Brown clusters and active learning. Koo et al. (2008) investigate semi-supervised dependency parsing based on Brown clusters.

The direction that Ringger et al. (2007) pursue is perhaps the most similar to ours. They attempt to reduce supervision required for high POS tagging performance based on active learning. But a critical difference is that they do not use word representations: in contrast, word representations are central to our approach.

Another closely relevant work is the work of Garrette and Baldridge (2013) who aim to learn a good POS tagger from limited resources. Notably, they faithfully simulate tagging resource-poor languages with human annotators. Our contribution is different in several important ways. Most importantly, our results are much more striking in the aspect of minimizing supervision. We obtain $> 90\%$ accuracy with a few hundred labeled words, whereas Garrette and Baldridge (2013) obtain 71-78% with 1,537-2,650 labeled words and tag dictionaries (i.e., the result of two hours of annotation efforts). They also do not make use of word representations which are the highlight of this work.

6 Conclusion

We have argued that that the sequence model of Brown et al. (1992), often used for deriving lexical representations, is particularly appropriate for capturing POS tags. We have demonstrated this claim by drastically reducing the amount of labeled data required for state-of-the-art POS tagging accuracy with word representations derived under the Brown model. Our simple framework MINITAGGER allows one to learn a functioning POS tagger with merely a few hundred labeled tokens, or an accurate POS tagger with less than 1% of the normally considered amount of training data.

We focused on utilizing lexical representations in a greedy framework, which is well-suited for the per-position accuracy metric (which is the standard metric for POS tagging). However, the result may be quite different if other metric is chosen, for instance the per-sentence accuracy metric. Thus improving tagging performance under different metrics using lexical representations may be a fruitful direction.

While they are not considered in this work, lexical representations *not* derived under the Brown model such as the skip-gram model in the WORD2VEC package Mikolov et al. (2013) can certainly be used for the same task. It may be illuminating to compare such different representations.

References

Taylor Berg-Kirkpatrick, Alexandre Bouchard-Côté, John DeNero, and Dan Klein. 2010. Painless unsupervised learning with features. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Com-*

putational Linguistics, pages 582–590. Association for Computational Linguistics.

Avrim Blum and Tom Mitchell. 1998. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 92–100. ACM.

Peter F Brown, Peter V Desouza, Robert L Mercer, Vincent J Della Pietra, and Jenifer C Lai. 1992. Class-based n-gram models of natural language. *Computational linguistics*, 18(4):467–479.

Christos Christodoulopoulos, Sharon Goldwater, and Mark Steedman. 2010. Two decades of unsupervised pos induction: How far have we come? In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 575–584. Association for Computational Linguistics.

Kenneth Ward Church. 1988. A stochastic parts program and noun phrase parser for unrestricted text. In *Proceedings of the second conference on Applied natural language processing*, pages 136–143. Association for Computational Linguistics.

Michael Collins and Yoram Singer. 1999. Unsupervised models for named entity classification. In *Proceedings of the joint SIGDAT conference on empirical methods in natural language processing and very large corpora*, pages 100–110.

Dipanjan Das and Slav Petrov. 2011. Unsupervised part-of-speech tagging with bilingual graph-based projections. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 600–609. Association for Computational Linguistics.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. Liblinear: A library for large linear classification. *The Journal of Machine Learning Research*, 9:1871–1874.

Dan Garrette and Jason Baldridge. 2013. Learning a part-of-speech tagger from two hours of annotation.

Aria Haghighi 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. Association for Computational Linguistics.

Harold Hotelling. 1936. Relations between two sets of variates. *Biometrika*, 28(3/4):321–377.

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

Young-Bum Kim and Benjamin Snyder. 2013. Optimal data set selection: An application to grapheme-to-phoneme conversion. In *HLT-NAACL*, pages 1196–1205.

Terry Koo, Xavier Carreras, and Michael Collins. 2008. Simple semi-supervised dependency parsing.

John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning*, pages 282–289.

Shen Li, Joao V Graça, and Ben Taskar. 2012. Wiki-ly supervised part-of-speech tagging. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1389–1398. Association for Computational Linguistics.

P. Liang. 2005. Semi-supervised learning for natural language. Master's thesis, Massachusetts Institute of Technology.

Christopher D Manning. 2011. Part-of-speech tagging from 97% to 100%: is it time for some linguistics? In *Computational Linguistics and Intelligent Text Processing*, pages 171–189. Springer.

Ryan T McDonald, Joakim Nivre, Yvonne Quirmbach-Brundage, Yoav Goldberg, Dipanjan Das, Kuzman Ganchev, Keith B Hall, Slav Petrov, Hao Zhang, Oscar Täckström, et al. 2013. Universal dependency annotation for multilingual parsing. In *ACL (2)*, pages 92–97.

Jean-Baptiste Michel, Yuan Kui Shen, Aviva Presser Aiden, Adrian Veres, Matthew K Gray, Joseph P Pickett, Dale Hoiberg, Dan Clancy, Peter Norvig, Jon Orwant, et al. 2011. Quantitative analysis of culture using millions of digitized books. *science*, 331(6014):176–182.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In C.J.C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc.

Scott Miller, Jethran Guinness, and Alex Zamanian. 2004. Name tagging with word clusters and discriminative training. In *HLT-NAACL*, volume 4, pages 337–342. Citeseer.

Naoaki Okazaki. 2007. Crfsuite: a fast implementation of conditional random fields (crfs).

Eric Ringger, Peter McClanahan, Robbie Haertel, George Busby, Marc Carmen, James Carroll, Kevin Seppi, and Deryle Lonsdale. 2007. Active learning for part-of-speech tagging: Accelerating corpus annotation. In *Proceedings of the Linguistic Annotation Workshop*, pages 101–108. Association for Computational Linguistics.

Burr Settles. 2010. Active learning literature survey. *University of Wisconsin, Madison*, 52:55–66.

Noah A Smith and Jason Eisner. 2005. Contrastive estimation: Training log-linear models on unlabeled data.

In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 354–362. Association for Computational Linguistics.

Karl Stratos, Do-kyum Kim, Michael Collins, and Daniel Hsu. 2014. A spectral algorithm for learning class-based n-gram models of natural language. In *Proceedings of the thirtieth conference on Uncertainty in Artificial Intelligence*.

Oscar Täckström, Dipanjan Das, Slav Petrov, Ryan McDonald, and Joakim Nivre. 2013. Token and type constraints for cross-lingual part-of-speech tagging. *Transactions of the Association for Computational Linguistics*, 1:1–12.

Kristina Toutanova and Mark Johnson. 2007. A bayesian lda-based model for semi-supervised part-of-speech tagging. In *Advances in Neural Information Processing Systems*, pages 1521–1528.

David Yarowsky. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd annual meeting on Association for Computational Linguistics*, pages 189–196. Association for Computational Linguistics.