Unsupervised Part-Of-Speech Tagging with Anchor Hidden Markov Models

Karl Stratos, Michael Collins* and **Daniel Hsu**
Department of Computer Science, Columbia University
{stratos, mcollins, djhsu}@cs.columbia.edu

Abstract

We tackle unsupervised part-of-speech (POS) tagging by learning hidden Markov models (HMMs) that are particularly well-suited for the problem. These HMMs, which we call *anchor HMMs*, assume that each tag is associated with at least one word that can have no other tag, which is a relatively benign condition for POS tagging (e.g., "the" is a word that appears only under the determiner tag). We exploit this assumption and extend the non-negative matrix factorization framework of Arora et al. (2013) to design a consistent estimator for anchor HMMs. In experiments, our algorithm is competitive with strong baselines such as the clustering method of Brown et al. (1992) and the log-linear model of Berg-Kirkpatrick et al. (2010). Furthermore, it produces an interpretable model in which hidden states are automatically lexicalized by words.

1 Introduction

Part-of-speech (POS) tagging without supervision is a quintessential problem in unsupervised learning for natural language processing (NLP). A major application of this task is reducing annotation cost: for instance, it can be used to produce rough syntactic annotations for a new language that has no labeled data, which can be subsequently refined by human annotators.

Hidden Markov models (HMMs) are a natural choice of model and have been a workhorse for this problem. Early works estimated vanilla HMMs with standard unsupervised learning methods such as the expectation-maximization (EM) algorithm, but it quickly became clear that they performed very poorly in inducing POS tags (Merialdo, 1994). Later works improved upon vanilla HMMs by incorporating specific structures that are well-suited for the task, such as a sparse prior (Johnson, 2007) or a hard-clustering assumption (Brown et al., 1992).

In this work, we tackle unsupervised POS tagging with HMMs whose structure is deliberately suitable for POS tagging. These HMMs impose an assumption that each hidden state is associated with an observation state ("anchor word") that can appear under no other state. For this reason, we denote this class of restricted HMMs by *anchor HMMs*. Such an assumption is relatively benign for POS tagging; it is reasonable to assume that each POS tag has at least one word that occurs only under that tag. For example, in English, "the" is an anchor word for the determiner tag; "laughed" is an anchor word for the verb tag.

We build on the non-negative matrix factorization (NMF) framework of Arora et al. (2013) to derive a consistent estimator for anchor HMMs. We make several new contributions in the process. First, to our knowledge, there is no previous work directly building on this framework to address unsupervised sequence labeling. Second, we generalize the NMF-based learning algorithm to obtain extensions that are important for empirical performance (Table 1). Third, we perform extensive experiments on unsupervised POS tagging and report competitive results against strong baselines such as the clustering method of Brown et al. (1992) and the log-linear

*Currently on leave at Google Inc. New York.

Transactions of the Association for Computational Linguistics, vol. 4, pp. 245–257, 2016. Action Editor: Hinrich Schütze.
Submission batch: 1/2016; Revision batch: 3/2016; Published 6/2016.

model of Berg-Kirkpatrick et al. (2010).

One characteristic of the approach is the immediate interpretability of inferred hidden states. Because each hidden state is associated with an observation, we can examine the set of such anchor observations to qualitatively evaluate the learned model. In our experiments on POS tagging, we find that anchor observations correspond to possible POS tags across different languages (Table 3). This property can be useful when we wish to develop a tagger for a new language that has no labeled data; we can label only the anchor words to achieve a complete labeling of the data.

This paper is structured as follows. In Section 2, we establish the notation we use throughout. In Section 3, we define the model family of anchor HMMs. In Section 4, we derive a matrix decomposition algorithm for estimating the parameters of an anchor HMM. In Section 5, we present our experiments on unsupervised POS tagging. In Section 6, we discuss related works.

2 Notation

We use $[n]$ to denote the set of integers $\{1, \ldots, n\}$. We use $\mathbf{E}[X]$ to denote the expected value of a random variable X. We define $\Delta^{m-1} := \{v \in \mathbb{R}^m : v_i \geq 0 \; \forall i, \sum_i v_i = 1\}$, i.e., the $(m-1)$-dimensional probability simplex. Given a vector $v \in \mathbb{R}^m$, we use $\text{diag}(v) \in \mathbb{R}^{m \times m}$ to denote the diagonal matrix with $v_1 \ldots v_m$ on the main diagonal. Given a matrix $M \in \mathbb{R}^{n \times m}$, we write $M_i \in \mathbb{R}^m$ to denote the i-th row of M (as a column vector).

3 The Anchor Hidden Markov Model

Definition 3.1. *An **anchor HMM (A-HMM)** is a 6-tuple $(n, m, \pi, t, o, \mathcal{A})$ for positive integers n, m and functions $\pi, t, o, \mathcal{A}$ where*

- *$[n]$ is a set of observation states.*
- *$[m]$ is a set of hidden states.*
- *$\pi(h)$ is the probability of generating $h \in [m]$ in the first position of a sequence.*
- *$t(h'|h)$ is the probability of generating $h' \in [m]$ given $h \in [m]$.*
- *$o(x|h)$ is the probability of generating $x \in [n]$ given $h \in [m]$.*
- *$\mathcal{A}(h) := \{x \in [n] : \; o(x|h) > 0 \wedge o(x|h') = 0 \; \forall h' \neq h\}$ is non-empty for each $h \in [m]$.*

In other words, an A-HMM is an HMM in which each hidden state h is associated with at least one "anchor" observation state that can be generated by, and only by, h. Note that the anchor condition implies $n \geq m$.

An equivalent definition of an A-HMM is given by organizing the parameters in matrix form. Under this definition, an A-HMM has parameters (π, T, O) where $\pi \in \mathbb{R}^m$ is a vector and $T \in \mathbb{R}^{m \times m}, O \in \mathbb{R}^{n \times m}$ are matrices whose entries are set to:

- $\pi_h = \pi(h)$ for $h \in [m]$
- $T_{h',h} = t(h'|h)$ for $h, h' \in [m]$
- $O_{x,h} = o(x|h)$ for $h \in [m], x \in [n]$

The anchor condition implies that $\text{rank}(O) = m$. To see this, consider the rows $O_{a_1} \ldots O_{a_m}$ where $a_h \in \mathcal{A}(h)$. Since each O_{a_h} has a single non-zero entry at the h-th index, the columns of O are linearly independent. We assume $\text{rank}(T) = m$.

An important special case of A-HMM introduced by Brown et al. (1992) is defined below. Under this A-HMM, every observation state is an anchor of some hidden state.

Definition 3.2. *A **Brown model** is an A-HMM in which $\mathcal{A}(1) \ldots \mathcal{A}(m)$ partition $[n]$.*

4 Parameter Estimation for A-HMMs

We now derive an algorithm for learning A-HMMs. The algorithm reduces the learning problem to an instance of NMF from which the model parameters can be computed in closed-form.

4.1 NMF

We give a brief review of the NMF method of Arora et al. (2013). (Exact) NMF is the following problem: given an $n \times d$ matrix $A = BC$ where $B \in \mathbb{R}^{n \times m}$ and $C \in \mathbb{R}^{m \times d}$ have non-negativity constraints, recover B and C. This problem is NP-hard in general (Vavasis, 2009), but Arora et al. (2013) provide an exact and efficient method when A has the following special structure:

Condition 4.1. *A matrix $A \in \mathbb{R}^{n \times d}$ satisfies this condition if $A = BC$ for $B \in \mathbb{R}^{n \times m}$ and $C \in \mathbb{R}^{m \times d}$ where*

Anchor-NMF
Input: $A \in \mathbb{R}^{n \times d}$ satisfying Condition 4.1 with $A = BC$ for some $B \in \mathbb{R}^{n \times m}$ and $C \in \mathbb{R}^{m \times d}$, value m

- For $h = 1 \dots m$, find a vertex a_h as

$$U \leftarrow \text{Gram-Schmidt}(\{A_{a_l}\}_{l=1}^{h-1})$$
$$a_h \leftarrow \arg\max_{x \in [n]} \left|\left|A_x - UU^\top A_x\right|\right|_2$$

  where Gram-Schmidt$(\{A_{a_l}\}_{l=1}^{h-1})$ is the Gram-Schmidt process that orthonormalizes $\{A_{a_l}\}_{l=1}^{h-1}$.

- For $x = 1 \dots n$, recover the x-th row of B as

$$\overline{B}_x \leftarrow \arg\min_{u \in \Delta^{m-1}} \left|\left|A_x - \sum_{h=1}^{m} u_h A_{a_h}\right|\right|_2 \quad (1)$$

- Set $\overline{C} = [A_{a_1} \dots A_{a_m}]^\top$.

Output: $\overline{B}$ and $\overline{C}$ such that $\overline{B}_h^\top = B_{\rho(h)}^\top$ and $\overline{C}_h = C_{\rho(h)}$ for some permutation $\rho : [m] \to [m]$

Figure 1: Non-negative matrix factorization algorithm of Arora et al. (2012).

1. *For each $x \in [n]$, $B_x \in \Delta^{m-1}$. I.e., each row of B defines a probability distribution over $[m]$.*

2. *For each $h \in [m]$, there is some $a_h \in [n]$ such that $B_{a_h,h} = 1$ and $B_{a_h,h'} = 0$ for all $h' \neq h$.*

3. *rank$(C) = m$.*

Since rank(B) = rank$(C) = m$ (by property 2 and 3), the matrix A must have rank m. Note that by property 1, each row of A is given by a convex combination of the rows of C: for $x \in [n]$,

$$A_x = \sum_{h=1}^{m} B_{x,h} \times C_h$$

Furthermore, by property 2 each $h \in [m]$ has an associated row $a_h \in [n]$ such that $A_{a_h} = C_{a_h}$. These properties can be exploited to recover B and C.

A concrete algorithm for factorizing a matrix satisfying Condition 4.1 is given in Figure 1 (Arora et al., 2013). It first identifies $a_1 \dots a_m$ (up to some permutation) by greedily locating the row of A furthest away from the subspace spanned by the vertices selected so far. Then it recovers each B_x as the convex coefficients required to combine $A_{a_1} \dots A_{a_m}$ to yield A_x. The latter computation (1) can be achieved with any constrained optimization method; we use the Frank-Wolfe algorithm (Frank and Wolfe, 1956). See Arora et al. (2013) for a proof of the correctness of this algorithm.

4.2 Random Variables

To derive our algorithm, we make use of certain random variables under the A-HMM. Let $(X_1, \dots, X_N) \in [n]^N$ be a random sequence of $N \geq 2$ observations drawn from the model, along with the corresponding hidden state sequence $(H_1, \dots, H_N) \in [m]^N$; independently, pick a position $I \in [N-1]$ uniformly at random. Let $Y_I \in \mathbb{R}^d$ be a d-dimensional vector which is conditionally independent of X_I given H_I, i.e., $P(Y_I | H_I, X_I) = P(Y_I | H_I)$. We will provide how to define such a variable in Section 4.4.1: Y_I will be a function of $(X_1, \dots, X_N)$ serving as a "context" representation of X_I revealing the hidden state H_I.

Define unigram probabilities $u^\infty, u^1 \in \mathbb{R}^n$ and bigram probabilities $\mathcal{B} \in \mathbb{R}^{n \times n}$ where

$$u_x^\infty := P(X_I = x) \qquad \forall x \in [n]$$
$$u_x^1 := P(X_I = x | I = 1) \qquad \forall x \in [n]$$
$$\mathcal{B}_{x,x'} := P(X_I = x, X_{I+1} = x') \qquad \forall x, x' \in [n]$$

Additionally, define $\bar{\pi} \in \mathbb{R}^m$ where

$$\bar{\pi}_h = P(H_I = h) \qquad \forall h \in [m] \quad (2)$$

We assume $\bar{\pi}_h > 0$ for all $h \in [m]$.

4.3 Derivation of a Learning Algorithm

The following proposition provides a way to use the NMF algorithm in Figure 1 to recover the emission parameters O up to scaling.

Proposition 4.1. *Let $X_I \in [n]$ and $Y_I \in \mathbb{R}^d$ be respectively an observation and a context vector drawn from the random process described in Section 4.2. Define a matrix $\Omega \in \mathbb{R}^{n \times d}$ with rows*

$$\Omega_x = \boldsymbol{E}[Y_I | X_I = x] \qquad \forall x \in [n] \quad (3)$$

If rank$(\Omega) = m$, then Ω satisfies Condition 4.1:

$$\Omega = \widetilde{O}\Theta$$

where $\widetilde{O}_{x,h} = P(H_I = h|X_I = x)$ *and* $\Theta_h = \boldsymbol{E}[Y_I|H_I = h]$.

Proof.

$$\begin{aligned}
&\mathbf{E}[Y_I|X_I = x] \\
&= \sum_{h=1}^{m} P(H_I = h|X_I = x) \times \mathbf{E}[Y_I|H_I = h, X_I = x] \\
&= \sum_{h=1}^{m} P(H_I = h|X_I = x) \times \mathbf{E}[Y_I|H_I = h]
\end{aligned}$$

The last equality follows by the conditional independence of Y_I. This shows $\Omega = \widetilde{O}\Theta$. By the anchor assumption of the A-HMM, each $h \in [m]$ has at least one $x \in \mathcal{A}(h)$ such that $P(H_I = h|X_I = x) = 1$, thus Ω satisfies Condition 4.1. □

A useful interpretation of Ω in Proposition 4.1 is that its rows $\Omega_1 \dots \Omega_n$ are d-dimensional vector representations of observation states forming a convex hull in $\mathbb{R}^d$. This convex hull has m vertices $\Omega_{a_1} \dots \Omega_{a_m}$ corresponding to anchors $a_h \in \mathcal{A}(h)$ which can be convexly combined to realize all $\Omega_1 \dots \Omega_n$.

Given $\widetilde{O}$, we can recover the A-HMM parameters as follows. First, we recover the stationary state distribution $\bar{\pi}$ as:

$$\begin{aligned}
\bar{\pi}_h &= \sum_{x \in [n]} P(H_I = h|X_I = x) \times P(X_I = x) \\
&= \sum_{x \in [n]} \widetilde{O}_{x,h} \times u^{\infty}_x
\end{aligned}$$

The emission parameters O are given by Bayes' theorem:

$$\begin{aligned}
O_{x,h} &= \frac{P(H_I = h|X_I = x) \times P(X_I = x)}{\sum_{x \in [n]} P(H_I = h|X_I = x) \times P(X_I = x)} \\
&= \frac{\widetilde{O}_{x,h} \times u^{\infty}_x}{\bar{\pi}_h}
\end{aligned}$$

Using the fact that the emission probabilities are position-independent, we see that the initial state distribution π satisfies $u^1 = O\pi$:

$$\begin{aligned}
u^1_x &= P(X_I = x|I = 1) \\
&= \sum_{h \in [m]} P(X_I = x|H_I = h, I = 1) \times P(H_I = h|I = 1) \\
&= \sum_{h \in [m]} O_{x,h} \times \pi_h
\end{aligned}$$

Learn-Anchor-HMM
Input: Ω in Proposition 4.1, number of hidden states m, bigram probabilities $\mathcal{B}$, unigram probabilities u^{∞}, u^1

- Compute $(\widetilde{O}, \Theta) \leftarrow$ **Anchor-NMF**(Ω, m).
- Recover the parameters:

$$\bar{\pi} \leftarrow \widetilde{O}^{\top} u^{\infty} \quad (4)$$

$$O \leftarrow \text{diag}(\bar{\pi})^{-1}\text{diag}(u^{\infty})\widetilde{O} \quad (5)$$

$$\pi = O^{+}u^1 \quad (6)$$

$$T \leftarrow (\text{diag}(\bar{\pi})^{-1}O^{+}\mathcal{B}(O^{\top})^{+})^{\top} \quad (7)$$

Output: A-HMM parameters (π, T, O)

Figure 2: NMF-based learning algorithm for A-HMMs. The algorithm **Anchor-NMF** is given in Figure 1.

Thus π can be recovered as $\pi = O^{+}u^1$ where O^{+} is the Moore–Penrose pseudoinverse of O. Finally, it can be algebraically verified that $\mathcal{B} = O\text{diag}(\bar{\pi})T^{\top}O^{\top}$ (Hsu et al., 2012). Since all the involved matrices have rank m, we can directly solve for T as

$$T = (\text{diag}(\bar{\pi})^{-1}O^{+}\mathcal{B}(O^{\top})^{+})^{\top}$$

Figure 2 shows the complete algorithm. As input, it receives a matrix Ω satisfying Proposition 4.1, the number of hidden states, and the probabilities of observed unigrams and bigrams. It first decomposes Ω using the NMF algorithm in Figure 1. Then it computes the A-HMM parameters whose solution is given analytically.

The following theorem guarantees the consistency of the algorithm.

Theorem 4.1. *Let* (π, T, O) *be an A-HMM such that rank*$(T) = m$ *and* $\bar{\pi}$ *defined in* (2) *has strictly positive entries* $\bar{\pi}_h > 0$. *Given random variables* Ω *satisfying Proposition 4.1 and* $\mathcal{B}, u^{\infty}, u^1$ *under this model, the algorithm* ***Learn-Anchor-HMM*** *in Figure 2 outputs* (π, T, O) *up to a permutation on hidden states.*

Proof. By Proposition 4.1, Ω satisfies Condition 4.1 with $\Omega = \widetilde{O}\Theta$, thus $\widetilde{O}$ can be recovered up to a permutation on columns with the algorithm **Anchor-NMF**. The consistency of the recovered parameters follows from the correctness of (4–7) under the rank conditions. □

4.3.1 Constrained Optimization for π and T

Note that (6) and (7) require computing the pseudoinverse of the estimated O, which can be expensive and vulnerable to sampling errors in practice. To make our parameter estimation more robust, we can explicitly impose probability constraints. We recover π by solving:

$$\pi = \arg\min_{\pi' \in \Delta^{m-1}} \left|\left| u^1 - O\pi' \right|\right|_2 \tag{8}$$

which can again be done with algorithms such as Frank-Wolfe. We recover T by maximizing the log likelihood of observation bigrams

$$\sum_{x,x'} \mathcal{B}_{x,x'} \log \left(\sum_{h,h' \in [m]} \bar{\pi}_h O_{x,h} T_{h',h} O_{x',h'} \right) \tag{9}$$

subject to the constraint $(T^\top)_h \in \Delta^{m-1}$. Since (9) is concave in T with other parameters O and $\bar{\pi}$ fixed, we can use EM to find the global optimum.

4.4 Construction of the Convex Hull Ω

In this section, we provide several ways to construct a convex hull Ω satisfying Proposition 4.1.

4.4.1 Choice of the Context Y_I

In order to satisfy Proposition 4.1, we need to define the context variable $Y_I \in \mathbb{R}^d$ with two properties:

- $P(Y_I|H_I, X_I) = P(Y_I|H_I)$
- The matrix Ω with rows

  $$\Omega_x = \mathbf{E}[Y_I|X_I = x] \qquad \forall x \in [n]$$

  has rank m.

A simple construction (Arora et al., 2013) is given by defining $Y_I \in \mathbb{R}^n$ to be an indicator vector for the next observation:

$$[Y_I]_{x'} = \begin{cases} 1 & \text{if } X_{I+1} = x' \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

The first condition is satisfied since X_{I+1} does not depend on X_I given H_I. For the second condition, observe that $\Omega_{x,x'} = P(X_{I+1} = x'|X_I = x)$, or in matrix form

$$\Omega = \text{diag}\,(u^\infty)^{-1}\,\mathcal{B} \tag{11}$$

Under the rank conditions in Theorem 4.1, (11) has rank m.

More generally, we can let Y_I be an observation (encoded as an indicator vector as in (10)) randomly drawn from a window of $L \in \mathbb{N}$ nearby observations. We can either only use the identity of the chosen observation (in which case $Y_I \in \mathbb{R}^n$) or additionally indicate the relative position in the window (in which case $Y_I \in \mathbb{R}^{nL}$). It is straightforward to verify that the above two conditions are satisfied under these definitions. Clearly, (11) is a special case with $L = 1$.

4.4.2 Reducing the Dimension of Ω_x

With the definition of Ω in the previous section, the dimension of Ω_x is $d = O(n)$ which can be difficult to work with when $n \gg m$. Proposition 4.1 allows us to reduce the dimension as long as the final matrix retains the form in (3) and has rank m. In particular, we can multiply Ω by any rank-m projection matrix $\Pi \in \mathbb{R}^{d \times m}$ on the right side: if Ω satisfies the properties in Proposition 4.1, then so does $\Omega\Pi$ with m-dimensional rows

$$(\Omega\Pi)_x = \mathbf{E}[Y_I\Pi|X_I = x]$$

Since $\text{rank}(\Omega) = m$, a natural choice of Π is the projection onto the best-fit m-dimensional subspace of the row space of Ω.

We mention that previous works on the NMF-learning framework have employed various projection methods, but they do not examine relative merits of their choices. For instance, Arora et al. (2013) simply use random projection, which is convenient for theoretical analysis. Cohen and Collins (2014) use a projection based on canonical correlation analysis (CCA) without further exploration. In contrast, we give a full comparison of valid construction methods and find that the choice of Ω is crucial in practice.

4.4.3 Construction of Ω for the Brown Model

We can formulate an alternative way to construct a valid Ω when the model is further restricted to be a Brown model. Since every observation is an anchor, $O_x \in \mathbb{R}^m$ has a single nonzero entry for every x. Thus the rows defined by $\Omega_x = O_x / ||O_x||$ (an indicator vector for the unique hidden state of x) form

Input: bigram probabilities $\mathcal{B}$, unigram probabilities u^∞, number of hidden states m, construction method τ

Scaled Matrices: ($\sqrt{\cdot}$ is element-wise)

$$\overline{\mathcal{B}} := \text{diag}\,(u^\infty)^{-1/2}\,\mathcal{B}\text{diag}\,(u^\infty)^{-1/2}$$

$$\widetilde{\mathcal{B}} := \text{diag}\left(\sqrt{u^\infty}\right)^{-1/2}\sqrt{\mathcal{B}}\text{diag}\left(\sqrt{u^\infty}\right)^{-1/2}$$

Singular Vectors: $U(M)$ ($V(M)$) is an $n \times m$ matrix of the left (right) singular vectors of M corresponding to the largest m singular values

- If $\tau \neq$ **brown**: set

$$\Omega \leftarrow \text{diag}\,(u^\infty)^{-1}\,\mathcal{B}\Pi$$

where the projection matrix $\Pi \in \mathbb{R}^{n\times m}$ is given by

$$\begin{aligned}
\Pi_{i,j} &\sim \mathcal{N}(0, 1/m) && \text{if } \tau = \textbf{random}\\
\Pi &= V(\text{diag}\,(u^\infty)^{-1}\,\mathcal{B}) && \text{if } \tau = \textbf{best-fit}\\
\Pi &= \text{diag}\,(u^\infty)^{-1/2}\,V(\overline{\mathcal{B}}) && \text{if } \tau = \textbf{cca}
\end{aligned}$$

- If $\tau =$ **brown**: compute the transformed emission matrix as $f(O) = U(\widetilde{\mathcal{B}})$ and set

$$\Omega \leftarrow \text{diag}(v)^{-1} f(O)$$

where $v_x := ||f(O)_x||_2$ is the length of the x-th row of $f(O)$.

Output: $\Omega \in \mathbb{R}^{n\times m}$ in Proposition 4.1

Figure 3: Algorithm for constructing a valid Ω with different construction methods. For simplicity, we only show the bigram construction (context size $L = 1$), but an extension for larger context ($L > 1$) is straightforward.

a trivial convex hull in which every point is a vertex. This corresponds to choosing an oracle context $Y_I \in \mathbb{R}^m$ where

$$[Y_I]_h = \begin{cases} 1 & \text{if } H_I = h \\ 0 & \text{otherwise} \end{cases}$$

It is possible to recover the Brown model parameters O up to element-wise scaling and rotation of rows using the algorithm of Stratos et al. (2015). More specifically, let $f(O) \in \mathbb{R}^{n\times m}$ denote the output of their algorithm. Then they show that for some vector $s \in \mathbb{R}^m$ with strictly positive entries and an orthogonal matrix $Q \in \mathbb{R}^{m\times m}$:

$$f(O) = O^{\langle 1/4\rangle}\text{diag}(s)Q^\top$$

where $O^{\langle 1/4\rangle}$ is an element-wise exponentiation of O by $1/4$. Since the rows of $f(O)$ are simply some scaling and rotation of the rows of O, using $\Omega_x = f(O)_x / ||f(O)_x||$ yields a valid Ω.

While we need to impose an additional assumption (the Brown model restriction) in order to justify this choice of Ω, we find in our experiments that it performs better than other alternatives. We speculate that this is because a Brown model is rather appropriate for the POS tagging task; many words are indeed unambiguous with respect to POS tags (Table 5). Also, the general effectiveness of $f(O)$ for representational purposes has been demostrated in previous works (Stratos et al., 2014; Stratos et al., 2015). By restricting the A-HMM to be a Brown model, we can piggyback on the proven effectiveness of $f(O)$.

Figure 3 shows an algorithm for constructing Ω with these different construction methods. For simplicity, we only show the bigram construction (context size $L = 1$), but an extension for larger context ($L > 1$) is straightforward as discussed earlier. The construction methods **random** (random projection), **best-fit** (projection to the best-fit subspace), and **cca** (CCA projection) all compute (11) and differ only in how the dimension is reduced. The construction method **brown** computes the transformed Brown parameters $f(O)$ as the left singular vectors of a scaled covariance matrix and then normalizes its rows. We direct the reader to Stratos et al. (2015) for a derivation of this calculation.

4.4.4 Ω with Feature Augmentation

The x-th row of Ω is a d-dimensional vector representation of x lying in a convex set with m vertices. This suggests a natural way to incorporate domain-specific features: we can add additional dimensions that provide information about hidden states from the surface form of x.

For instance, consider the the POS tagging task. In the simple construction (11), the representation of word x is defined in terms of neighboring words x':

$$[\Omega_x]_{x'} = \mathbf{E}\left[\mathbb{1}\left(X_{I+1} = x'\right)|X_I = x\right]$$

where $\mathbb{1}(\cdot) \in \{0, 1\}$ is the indicator function. We can augment this vector with s additional dimen-

sions indicating the spelling features of x. For instance, the $(n+1)$-th dimension may be defined as:

$$[\Omega_x]_{n+1} = \mathbf{E}\left[\mathbb{1}\left(x \text{ ends in "ing"}\right) | X_I = x\right]$$

This value will be generally large for verbs and small for non-verbs, nudging verbs closer together and away from non-verbs. The modified $(n+s)$-dimensional representation is followed by the usual dimension reduction. Note that the spelling features are a deterministic function of a word, and we are implicitly assuming that they are independent of the word given its tag. While this is of course not true in practice, we find that these features can significantly boost the tagging performance.

5 Experiments

We evaluate our A-HMM learning algorithm on the task of unsupervised POS tagging. The goal of this task is to induce the correct sequence of POS tags (hidden states) given a sequence of words (observation states). The anchor condition corresponds to assuming that each POS tag has at least one word that occurs only under that tag.

5.1 Background on Unsupervised POS Tagging

Unsupervised POS tagging has long been an active area of research (Smith and Eisner, 2005a; Johnson, 2007; Toutanova and Johnson, 2007; Haghighi and Klein, 2006; Berg-Kirkpatrick et al., 2010), but results on this task are complicated by varying assumptions and unclear evaluation metrics (Christodoulopoulos et al., 2010). Rather than addressing multiple alternatives for evaluating unsupervised POS tagging, we focus on a simple and widely used metric: many-to-one accuracy (i.e., we map each hidden state to the most frequently coinciding POS tag in the labeled data and compute the resulting accuracy).

5.1.1 Better Model v.s. Better Learning

Vanilla HMMs are notorious for their mediocre performance on this task, and it is well known that they perform poorly largely because of *model misspecification*, not because of suboptimal parameter estimation (e.g., because EM gets stuck in local optima). More generally, a large body of work points to the inappropriateness of simple generative models for unsupervised induction of linguistic structure (Merialdo, 1994; Smith and Eisner, 2005b; Liang and Klein, 2008).

Consequently, many works focus on using more expressive models such as log-linear models (Smith and Eisner, 2005a; Berg-Kirkpatrick et al., 2010) and Markov random fields (MRF) (Haghighi and Klein, 2006). These models are shown to deliver good performance even though learning is approximate. Thus one may question the value of having a consistent estimator for A-HMMs and Brown models in this work: if the model is wrong, what is the point of learning it accurately?

However, there is also ample evidence that HMMs are competitive for unsupervised POS induction when they incorporate domain-specific structures. Johnson (2007) is able to outperform the sophisticated MRF model of Haghighi and Klein (2006) on one-to-one accuracy by using a sparse prior in HMM estimation. The clustering method of Brown et al. (1992) which is based on optimizing the likelihood under the Brown model (a special case of HMM) remains a baseline difficult to outperform (Christodoulopoulos et al., 2010).

We add to this evidence by demonstrating the effectiveness of A-HMMs on this task. We also check the anchor assumption on data and show that the A-HMM model structure is in fact appropriate for the problem (Table 5).

5.2 Experimental Setting

We use the universal treebank dataset (version 2.0) which contains sentences annotated with 12 POS tag types for 10 languages (McDonald et al., 2013). All models are trained with 12 hidden states. We use the English portion to experiment with different hyperparameter configurations. At test time, we fix a configuration (based on the English portion) and apply it across all languages.

The list of compared methods is given below:

BW The Baum-Welch algorithm, an EM algorithm for HMMs (Baum and Petrie, 1966).

CLUSTER A parameter estimation scheme for HMMs based on Brown clustering (Brown et al., 1992). We run the Brown clustering algorithm[1] to obtain 12 word clusters $C_1 \dots C_{12}$. Then we set

[1] We use the implementation of Liang (2005).

the emission parameters $o(x|h)$, transition parameters $t(h'|h)$, and prior $\pi(h)$ to be the maximum-likelihood estimates under the fixed clusters.

ANCHOR Our algorithm **Learn-Anchor-HMM** in Figure 2 but with the constrained optimization (8) and (9) for estimating π and T.[2]

ANCHOR-FEATURES Same as ANCHOR but employs the feature augmentation scheme described in Section 4.4.4.

LOG-LINEAR The unsupervised log-linear model described in Berg-Kirkpatrick et al. (2010). Instead of emission parameters $o(x|h)$, the model maintains a miniature log-linear model with a weight vector w and a feature function ϕ. The probability of a word x given tag h is computed as

$$p(x|h) = \frac{\exp(w^\top \phi(x,h))}{\sum_{x \in [n]} \exp(w^\top \phi(x,h))}$$

The model can be trained by maximizing the likelihood of observed sequences. We use L-BFGS to directly optimize this objective.[3] This approach obtains the current state-of-the-art accuracy on fine-grained (45 tags) English WSJ dataset.

We use maximum marginal decoding for HMM predictions: i.e., at each position, we predict the most likely tag given the entire sentence.

5.3 Practical Issues with the Anchor Algorithm

In our experiments, we find that **Anchor-NMF** (Figure 1) tends to propose extremely rare words as anchors. A simple fix is to search for anchors only among relatively frequent words. We find that any reasonable frequency threshold works well; we use the 300 most frequent words. Note that this is not a problem if these 300 words include anchor words corresponding to all the 12 tags.

We must define the context for constructing Ω. We use the previous and next words (i.e., context size $L = 2$) marked with relative positions. Thus Ω has $2n$ columns before dimension reduction. Table 1 shows the performance on the English portion with different construction methods for Ω. The Brown construction ($\tau =$ **brown** in Figure 3) clearly performs the best: essentially, the anchor algorithm is used to extract the HMM parameters from the CCA-based word embeddings of Stratos et al. (2015).

Choice of Ω	Accuracy
Random	48.2
Best-Fit	53.4
CCA	57.0
Brown	**66.1**

Table 1: Many-to-one accuracy on the English data with different choices of the convex hull Ω (Figure 3). These results do not use spelling features.

We also explore feature augmentation discussed in Section 4.4.4. For comparison, we employ the same word features used by Berg-Kirkpatrick et al. (2010):

- Indicators for whether a word is capitalized, contains a hyphen, or contains a digit
- Suffixes of length 1, 2, and 3

We weigh the l_2 norm of these extra dimensions in relation to the original dimensions: we find a small weight (e.g., 0.1 of the norm of the original dimensions) works well. We also find that these features can sometimes significantly improve the performance. For instance, the accuracy on the English portion can be improved from 66.1% to 71.4% with feature augmentation.

Another natural experiment is to refine the HMM parameters obtained from the anchor algorithm (or Brown clusters) with a few iterations of the Baum-Welch algorithm. In our experiments, however, it did not significantly improve the tagging performance, so we omit this result.

5.4 Tagging Accuracy

Table 2 shows the many-to-one accuracy on all languages in the dataset. For the Baum-Welch algorithm and the unsupervised log-linear models, we report the mean and the standard deviation (in parentheses) of 10 random restarts run for 1,000 iterations.

Both ANCHOR and ANCHOR-FEATURES compete favorably. On 5 out of 10 languages, ANCHOR-FEATURES achieves the highest accuracy, often

[2] https://github.com/karlstratos/anchor

[3] We use the implementation of Berg-Kirkpatrick et al. (2010) (personal communication).

Model	de	en	es	fr	id	it	ja	ko	pt-br	sv
BW	(4.8) 45.5	(3.4) 59.8	(2.2) 60.6	(3.6) 60.1	(3.1) 49.6	(2.6) 51.5	(2.1) 59.5	(0.6) 51.7	(3.7) 59.5	(3.0) 42.4
CLUSTER	60.0	62.9	67.4	66.4	59.3	**66.1**	60.3	47.5	**67.4**	**61.9**
ANCHOR	61.1	66.1	69.0	68.2	63.7	60.4	65.3	53.8	64.9	51.1
ANCHOR-FEATURES	63.4	**71.4**	**74.3**	**71.9**	**67.3**	60.2	69.4	**61.8**	65.8	61.0
LOG-LINEAR	(1.8) **67.5**	(3.5) 62.4	(3.1) 67.1	(4.5) 62.1	(3.9) 61.3	(2.9) 52.9	(2.9) **78.2**	(3.6) 60.5	(2.2) 63.2	(2.5) 56.7

Table 2: Many-to-one accuracy on each language using 12 universal tags. The first four models are HMMs estimated with the Baum-Welch algorithm (BW), the clustering algorithm of Brown et al. (1992), the anchor algorithm without (ANCHOR) and with (ANCHOR-FEATURES) feature augmentation. LOG-LINEAR is the model of Berg-Kirkpatrick et al. (2010) trained with the direct-gradient method using L-BFGS. For BW and LOG-LINEAR, we report the mean and the standard deviation (in parentheses) of 10 random restarts run for 1,000 iterations.

closely followed by ANCHOR. The Brown clustering estimation is also competitive and has the highest accuracy on 3 languages. Not surprisingly, vanilla HMMs trained with BW perform the worst (see Section 5.1.1 for a discussion).

LOG-LINEAR is a robust baseline and performs the best on the remaining 2 languages. It performs especially strongly on Japanese and Korean datasets in which poorly segmented strings such as "1950年11月5日には" (on November 5, 1950) and "40.3%로" (by 40.3%) abound. In these datasets, it is crucial to make effective use of morphological features.

5.5 Qualitative Analysis

5.5.1 A-HMM Parameters

An A-HMM can be easily interpreted since each hidden state is marked with an anchor observation. Table 3 shows the 12 anchors found in each language. Note that these anchor words generally have a wide coverage of possible POS tags.

We also experimented with using true anchor words (obtained from labeled data), but they did not improve performance over automatically induced anchors. Since anchor discovery is inherently tied to parameter estimation, it is better to obtain anchors in a data-driven manner. In particular, certain POS tags (e.g., X) appear quite infrequently, and the model is worse off by being forced to allocate a hidden state for such a tag.

Table 4 shows words with highest emission probabilities $o(x|h)$ under each anchor. We observe that an anchor is representative of a certain group of words. For instance, the state "loss" represents noun-like words, "1" represents numbers, "on" represents preposition-like words, "one" represents determiner-like words, and "closed" represents verb-like words. The conditional distribution is peaked for anchors that represent function tags (e.g., determiners, punctuation) and flat for anchors that represent content tags (e.g., nouns). Occasionally, an anchor assigns high probabilities to words that do not seem to belong to the corresponding POS tag. But this is to be expected since $o(x|h) \propto P(X_I = x)$ is generally larger for frequent words.

5.5.2 Model Assumptions on Data

Table 5 checks the assumptions in A-HMMs and Brown models on the universal treebank dataset. The anchor assumption is indeed satisfied with 12 universal tags: in every language, each tag has at least one word uniquely associated with the tag. The Brown assumption (each word has exactly one possible tag) is of course not satisfied, since some words are genuinely ambiguous with respect to their POS tags. However, the percentage of unambiguous words is very high (well over 90%). This analysis supports that the model assumptions made by A-HMMs and Brown models are appropriate for POS tagging.

Table 6 reports the log likelihood (normalized by the number of words) on the English portion of different estimation methods for HMMs. BW and CLUSTER obtain higher likelihood than the anchor algorithm, but this is expected given that both EM

de	en	es	fr	id	it	ja	ko	pt-br	sv
empfehlen	loss	y	avait	bulan	radar	お世話に	완전	E	och
wie	1	hizo	commune	tetapi	però	ないと	중에	de	bör
;	on	-	Le	wilayah	sulle	ことにより	경우	partida	grund
Sein	one	especie	de	-	-	されている。	줄	fazer	mellan
Berlin	closed	Además	président	Bagaimana	Stati	ものを	같아요	meses	i
und	are	el	qui	,	Lo	,	많은	os	sociala
,	take	países	(	sama	legge	した	,	:	.
-	,	la	à	.	al	それは	볼	diretor	bli
der	vice	España	États	dan	far-	、	자신의	2010	den
im	to	en	Unis	Utara	di	幸福の	받고	,	,
des	York	de	Cette	pada	la	ことが	맛있는	uma	tid
Region	Japan	municipio	quelques	yang	art.	通常の	위한	O	Detta

Table 3: Anchor words found in each language (model ANCHOR-FEATURES).

loss	year (.02)	market (.01)	share (.01)	company (.01)	stock (.01)	quarter (.01)	shares (.01)	price (.01)
1	1 (.03)	10 (.02)	30 (.02)	15 (.02)	8 (.02)	2 (.01)	20 (.01)	50 (.01)
on	of (.14)	in (.12)	. (.08)	for (.06)	on (.04)	by (.04)	from (.04)	and (.03)
one	the (.23)	a (.12)	“ (.03)	an (.03)	$ (.02)	its (.02)	that (.02)	this (.02)
closed	said (.05)	’s (.02)	is (.02)	says (.02)	was (.01)	has (.01)	had (.01)	expected (.01)
are	and (.08)	is (.08)	are (.05)	was (.04)	’s (.04)	“ (.04)	has (.03)	of (.03)
take	be (.04)	% (.02)	have (.02)	million (.02)	But (.02)	do (.01)	The (.01)	make (.01)
,	, (.53)	. (.25)	and (.05)	” (.04)	% (.01)	million (.01)	– (.01)	that (.01)
vice	’s (.03)	The (.02)	“ (.02)	New (.01)	and (.01)	new (.01)	first (.01)	chief (.01)
to	to (.39)	. (.11)	a (.06)	will (.04)	$ (.03)	n’t (.03)	would (.02)	% (.02)
York	the (.15)	a (.05)	The (.04)	of (.04)	’s (.04)	million (.01)	% (.01)	its (.01)
Japan	Mr. (.03)	it (.02)	” (.02)	$ (.02)	he (.02)	that (.02)	which (.01)	company (.01)

Table 4: Most likely words under each anchor word (English model ANCHOR-FEATURES). Emission probabilities $o(x|h)$ are given in parentheses.

and Brown clustering directly optimize likelihood. In contrast, the anchor algorithm is based on the method of moments and does not (at least directly) optimize likelihood. Note that high likelihood does not imply high accuracy under HMMs.

6 Related Work

6.1 Latent-Variable Models

There has recently been great progress in estimation of models with latent variables. Despite the NP-hardness in general cases (Terwijn, 2002; Arora et al., 2012), many algorithms with strong theoretical guarantees have emerged under natural assumptions. For example, for HMMs with full-rank conditions, Hsu et al. (2012) derive a consistent estimator of the marginal distribution of observed sequences. Anandkumar et al. (2014) propose an exact tensor decomposition method for learning a wide class of latent variable models with similar non-degeneracy conditions. Arora et al. (2013) derive a provably correct learning algorithm for topic models with a certain parameter structure.

The anchor-based framework has been originally formulated for learning topic models (Arora et al., 2013). It has been subsequently adopted to learn other models such as latent-variable probabilistic context-free grammars (Cohen and Collins, 2014). In our work, we have extended this framework to address unsupervised sequence labeling.

Zhou et al. (2014) also extend Arora et al. (2013)’s framework to learn various models including HMMs, but they address a more general problem. Consequently, their algorithm draws from Anandkumar et al. (2012) and is substantially different from ours.

6.2 Unsupervised POS Tagging

Unsupervised POS tagging is a classic problem in unsupervised learning that has been tackled with various approaches. Johnson (2007) observes that

	de	en	es	fr	id	it	ja	ko	pt-br	sv
% anchored tags	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
% unambig words	96.6	91.5	94.0	94.2	94.8	94.8	99.5	98.4	94.8	97.4

.	VERB	PRON	ADP	NOUN	ADV	CONJ	DET	NUM	ADJ	X	PRT
,	said	it	from	Mr.	n't	or	which	billion	new	bono	na
53928	6339	5147	4856	4436	3582	2748	2458	1935	1542	8	5

Table 5: Verifying model assumptions on the universal treebank. The anchor assumption is satisfied in every language. The Brown assumption (each word has exactly one possible tag) is violated but not by a large margin. The lower table shows the most frequent anchor word and its count under each tag on the English portion.

Model	Normalized LL	Acc
BW	-6.45	59.8
CLUSTER	-6.71	62.9
ANCHOR	-7.06	66.1
ANCHOR-FEATURES	-7.05	71.4

Table 6: Log likelihood normalized by the number of words on English (along with accuracy). For BW, we report the mean of 10 random restarts run for 1,000 iterations.

Models	Accuracy
BW	62.6 (1.1)
CLUSTER	65.6
ANCHOR	67.2
ANCHOR-FEATURES	67.7
LOG-LINEAR	74.9 (1.5)

Table 7: Many-to-one accuracy on the English data with 45 original tags. We use the same setting as in Table 2. For BW and LOG-LINEAR, we report the mean and the standard deviation (in parentheses) of 10 random restarts run for 1,000 iterations.

EM performs poorly in this task because it induces flat distributions; this is not the case with our algorithm as seen in the peaky distributions in Table 4. Haghighi and Klein (2006) assume a set of prototypical words for each tag and report high accuracy. In contrast, our algorithm automatically finds such prototypes in a subroutine.

Berg-Kirkpatrick et al. (2010) achieve the state-of-the-art result in unsupervised fine-grained POS tagging (mid-70%). As described in Section 5.2, their model is an HMM in which probabilties are given by log-linear models. Table 7 provides a point of reference comparing our work with Berg-Kirkpatrick et al. (2010) in their setting: models are trained and tested on the entire 45-tag WSJ dataset. Their model outperforms our approach in this setting: with fine-grained tags, spelling features become more important, for instance to distinguish "played" (VBD) from "play" (VBZ). Nonetheless, we have shown that our approach is competitive when universal tags are used (Table 2).

Many past works on POS induction predate the introduction of the universal tagset by Petrov et al. (2012) and thus report results with fine-grained tags. More recent works adopt the universal tagset but they leverage additional resources. For instance, Das and Petrov (2011) and Täckström et al. (2013) use parallel data to project POS tags from a supervised source language. Li et al. (2012) use tag dictionaries built from Wiktionary. Thus their results are not directly comparable to ours.[4]

7 Conclusion

We have presented an exact estimation method for learning anchor HMMs from unlabeled data. There are several directions for future work. An important direction is to extend the method to a richer family of models such as log-linear models or neural networks. Another direction is to further generalize the method to handle a wider class of HMMs by relaxing the anchor condition (Condition 4.1). This will require a significant extension of the NMF algorithm in Figure 1.

[4]Das and Petrov (2011) conduct unsupervised experiments using the model of Berg-Kirkpatrick et al. (2010), but their dataset and evaluation method differ from ours.

Acknowledgments

We thank Taylor Berg-Kirkpatrick for providing the implementation of Berg-Kirkpatrick et al. (2010). We also thank anonymous reviewers for their constructive comments.

References

Animashree Anandkumar, Daniel Hsu, and Sham M. Kakade. 2012. A method of moments for mixture models and hidden Markov models. In *Twenty-Fifth Annual Conference on Learning Theory*.

Animashree Anandkumar, Rong Ge, Daniel Hsu, Sham M. Kakade, and Matus Telgarsky. 2014. Tensor decompositions for learning latent variable models. *Journal of Machine Learning Research*, 15(1):2773–2832.

Sanjeev Arora, Rong Ge, and Ankur Moitra. 2012. Learning topic models–going beyond SVD. In *Foundations of Computer Science (FOCS), 2012 IEEE 53rd Annual Symposium on*, pages 1–10. IEEE.

Sanjeev Arora, Rong Ge, Yonatan Halpern, David Mimno, Ankur Moitra, David Sontag, Yichen Wu, and Michael Zhu. 2013. A practical algorithm for topic modeling with provable guarantees. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 280–288.

Leonard E. Baum and Ted Petrie. 1966. Statistical inference for probabilistic functions of finite state Markov chains. *The Annals of Mathematical Statistics*, 37(6):1554–1563.

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

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.

Shay B. Cohen and Michael Collins. 2014. A provably correct learning algorithm for latent-variable PCFGs. In *Proceedings of ACL*.

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.

Marguerite Frank and Philip Wolfe. 1956. An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, 3(1-2):95–110.

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.

Daniel Hsu, Sham M. Kakade, and Tong Zhang. 2012. A spectral algorithm for learning hidden Markov models. *Journal of Computer and System Sciences*, 78(5):1460–1480.

Mark Johnson. 2007. Why doesn't EM find good HMM POS-taggers? In *EMNLP-CoNLL*, pages 296–305.

Shen Li, Joao V. Graça, and Ben Taskar. 2012. Wikily 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.

Percy Liang and Dan Klein. 2008. Analyzing the errors of unsupervised learning. In *ACL*, pages 879–887.

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

Ryan T. McDonald, Joakim Nivre, Yvonne Quirmbach-Brundage, Yoav Goldberg, Dipanjan Das, Kuzman Ganchev, Keith B. Hall, Slav Petrov, Hao Zhang, Oscar Täckström, Claudia Bedini, Núria B. Castelló, and Jungmee Lee. 2013. Universal dependency annotation for multilingual parsing. In *ACL*, pages 92–97.

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

Slav Petrov, Dipanjan Das, and Ryan McDonald. 2012. A universal part-of-speech tagset. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*.

Noah A. Smith and Jason Eisner. 2005a. 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.

Noah A. Smith and Jason Eisner. 2005b. Guiding unsupervised grammar induction using contrastive estimation. In *Proc. of IJCAI Workshop on Grammatical Inference Applications*, pages 73–82.

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*.

Karl Stratos, Michael Collins, and Daniel Hsu. 2015. Model-based word embeddings from decompositions of count matrices. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1282–1291, Beijing, China, July. Association for Computational Linguistics.

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.

Sebastiaan A. Terwijn. 2002. On the learnability of hidden Markov models. In *Grammatical Inference: Algorithms and Applications*, pages 261–268. Springer.

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.

Stephen A. Vavasis. 2009. On the complexity of nonnegative matrix factorization. *SIAM Journal on Optimization*, 20(3):1364–1377.

Tianyi Zhou, Jeff A. Bilmes, and Carlos Guestrin. 2014. Divide-and-conquer learning by anchoring a conical hull. In *Advances in Neural Information Processing Systems*, pages 1242–1250.