Semi-Supervised Learning of Sequence Models with the Method of Moments

Zita Marinho[*♯] **André F. T. Martins**[†♡◇] **Shay B. Cohen**[♣] **Noah A. Smith**[♠]

[*]Instituto de Sistemas e Robótica, Instituto Superior Técnico, 1049-001 Lisboa, Portugal
[†]Instituto de Telecomunicações, Instituto Superior Técnico, 1049-001 Lisboa, Portugal
[♯]School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213, USA
[♡]Unbabel Lda, Rua Visconde de Santarém, 67-B, 1000-286 Lisboa, Portugal
[◇]Priberam Labs, Alameda D. Afonso Henriques, 41, 2°, 1000-123 Lisboa, Portugal
[♣]School of Informatics, University of Edinburgh, Edinburgh EH8 9AB, UK
[♠]Computer Science & Engineering, University of Washington, Seattle, WA 98195, USA

zmarinho@cmu.edu, andre.martins@unbabel.com,
scohen@inf.ed.ac.uk, nasmith@cs.washington.edu

Abstract

We propose a fast and scalable method for semi-supervised learning of sequence models, based on anchor words and moment matching. Our method can handle hidden Markov models with feature-based log-linear emissions. Unlike other semi-supervised methods, no decoding passes are necessary on the unlabeled data and no graph needs to be constructed—only one pass is necessary to collect moment statistics. The model parameters are estimated by solving a small quadratic program for each feature. Experiments on part-of-speech (POS) tagging for Twitter and for a low-resource language (Malagasy) show that our method can learn from very few annotated sentences.

1 Introduction

Statistical learning of NLP models is often limited by the scarcity of annotated data. Weakly supervised methods have been proposed as an alternative to laborious manual annotation, combining large amounts of unlabeled data with limited resources, such as tag dictionaries or small annotated datasets (Merialdo, 1994; Smith and Eisner, 2005; Garrette et al., 2013). Unfortunately, most semi-supervised learning algorithms for the structured problems found in NLP are computationally expensive, requiring multiple decoding passes through the unlabeled data, or expensive similarity graphs. More scalable learning algorithms are in demand.

In this paper, we propose a moment-matching method for semi-supervised learning of sequence models. Spectral learning and moment-matching approaches have recently proved a viable alternative to expectation-maximization (EM) for unsupervised learning (Hsu et al., 2012; Balle and Mohri, 2012; Bailly et al., 2013), supervised learning with latent variables (Cohen and Collins, 2014; Quattoni et al., 2014; Stratos et al., 2013) and topic modeling (Arora et al., 2013; Nguyen et al., 2015). These methods have learnability guarantees, do not suffer from local optima, and are computationally less demanding.

Unlike spectral methods, ours does not require an orthogonal decomposition of any matrix or tensor. Instead, it considers a more restricted form of supervision: words that have unambiguous annotations, so-called **anchor words** (Arora et al., 2013). Rather than identifying anchor words from unlabeled data (Stratos et al., 2016), we extract them from a small labeled dataset or from a dictionary. Given the anchor words, the estimation of the model parameters can be made efficient by collecting moment statistics from unlabeled data, then solving a small quadratic program for each word.

Our contributions are as follows:

- We adapt anchor methods to semi-supervised learning of generative sequence models.
- We show how our method can also handle log-linear feature-based emissions.
- We apply this model to POS tagging. Our experiments on the Twitter dataset introduced by Gimpel et al. (2011) and on the dataset introduced by Garrette et al. (2013) for Malagasy, a low-resource language, show that our method does particularly well with very little labeled data, outperforming semi-supervised EM and self-training.

2 Sequence Labeling

In this paper, we address the problem of sequence labeling. Let $\boldsymbol{x}_{1:L} = \langle x_1, \ldots, x_L \rangle$ be a sequence of L input observations (for example, words in a sentence). The goal is to predict a sequence of labels $\boldsymbol{h}_{1:L} = \langle h_1, \ldots, h_L \rangle$, where each h_i is a label for the observation x_i (for example, the word's POS tag).

We start by describing two generative sequence models: hidden Markov models (HMMs, §2.1), and their generalization with emission features (§2.2). Later, we propose a weakly-supervised method for estimating these models' parameters (§3–§4) based only on observed statistics of words and contexts.

2.1 Hidden Markov Models

We define random variables $\boldsymbol{X} := \langle X_1, \ldots, X_L \rangle$ and $\boldsymbol{H} := \langle H_1, \ldots, H_L \rangle$, corresponding to observations and labels, respectively. Each X_i is a random variable over a set $\mathcal{X}$ (the vocabulary), and each H_i ranges over $\mathcal{H}$ (a finite set of "states" or "labels"). We denote the vocabulary size by $V = |\mathcal{X}|$, and the number of labels by $K = |\mathcal{H}|$. A first-order HMM has the following generative scheme:

$$p(\boldsymbol{X} = \boldsymbol{x}_{1:L}, \boldsymbol{H} = \boldsymbol{h}_{1:L}) := \tag{1}$$
$$\prod_{\ell=1}^{L} p(X_\ell{=}x_\ell \mid H_\ell{=}h_\ell) \prod_{\ell=0}^{L} p(H_{\ell+1}{=}h_{\ell+1} \mid H_\ell{=}h_\ell),$$

where we have defined $h_0 = \textsc{start}$ and $h_{L+1} = \textsc{stop}$. We adopt the following notation for the parameters:

- The **emission matrix** $\mathbf{O} \in \mathbb{R}^{V \times K}$, defined as $O_{x,h} := p(X_\ell = x \mid H_\ell = h), \forall h \in \mathcal{H}, x \in \mathcal{X}$.
- The **transition matrix** $\mathbf{T} \in \mathbb{R}^{(K+2)\times(K+2)}$, defined as $T_{h,h'} := p(H_{\ell+1} = h \mid H_\ell = h')$, for every $h, h' \in \mathcal{H} \cup \{\textsc{start}, \textsc{stop}\}$. This matrix satisfies $\mathbf{T}^\top \mathbf{1} = \mathbf{1}$.[1]

Throughout the rest of the paper we will adopt $X \equiv X_\ell$ and $H \equiv H_\ell$ to simplify notation, whenever the index ℓ is clear from the context. Under this generative process, predicting the most probable label sequence $\boldsymbol{h}_{1:L}$ given observations $\boldsymbol{x}_{1:L}$ is accomplished with the Viterbi algorithm in $O(LK^2)$ time.

If labeled data are available, the model parameters $\mathbf{O}$ and $\mathbf{T}$ can be estimated with the maximum likelihood principle, which boils down to a simple counting of events and normalization. If we only have unlabeled data, the traditional approach is the expectation-maximization (EM) algorithm, which alternately decodes the unlabeled examples and updates the model parameters, requiring multiple passes over the data. The same algorithm can be used in semi-supervised learning when labeled and unlabeled data are combined, by initializing the model parameters with the supervised estimates and interpolating the estimates in the M-step.

2.2 Feature-Based Hidden Markov Models

Sequence models with log-linear emissions have been considered by Smith and Eisner (2005), in a discriminative setting, and by Berg-Kirkpatrick et al. (2010), as generative models for POS induction. Feature-based HMMs (FHMMs) define a feature function for words, $\boldsymbol{\phi}(X) \in \mathbb{R}^W$, which can be discrete or continuous. This allows, for example, to indicate whether an observation, corresponding to a word, starts with an uppercase letter, contains digits or has specific affixes. More generally, it helps with the treatment of out-of-vocabulary words. The emission probabilities are modeled as K conditional distributions parametrized by a log-linear model, where the $\boldsymbol{\theta}_h \in \mathbb{R}^W$ represent feature weights:

$$p(X = x \mid H = h) := \exp(\boldsymbol{\theta}_h^\top \boldsymbol{\phi}(x))/Z(\boldsymbol{\theta}_h). \tag{2}$$

Above, $Z(\boldsymbol{\theta}_h) := \sum_{x' \in \mathcal{X}} \exp(\boldsymbol{\theta}_h^\top \boldsymbol{\phi}(x'))$ is a normalization factor. We will show in §4 how our moment-based semi-supervised method can also be used to learn the feature weights $\boldsymbol{\theta}_h$.

3 Semi-Supervised Learning via Moments

We now describe our moment-based semi-supervised learning method for HMMs. Throughout, we assume the availability of a small labeled dataset $\mathcal{D}_L$ and a large unlabeled dataset $\mathcal{D}_U$.

The full roadmap of our method is shown as Algorithm 1. Key to our method is the decomposition of a **context-word moment matrix** $\mathbf{Q} \in \mathbb{R}^{C \times V}$, which counts co-occurrences of words and contexts,

[1] That is, it satisfies $\sum_{h=1}^{K} p(H_{\ell+1} = h \mid H_\ell = h') + p(H_{\ell+1} = \text{STOP} \mid H_\ell = h') = 1$; and also $\sum_{h=1}^{K} p(H_1 = h \mid H_0 = \text{START}) = 1$.

Algorithm 1 Semi-Supervised Learning of HMMs with Moments

```
Input: Labeled dataset $\mathcal{D}_L$, unlabeled dataset $\mathcal{D}_U$
Output: Estimates of emissions $\mathbf{O}$ and transitions $\mathbf{T}$
Estimate context-word moments $\widehat{\mathbf{Q}}$ from $\mathcal{D}_U$ (Eq. 5)
for each label $h \in \mathcal{H}$ do
    Extract set of anchor words $\mathcal{A}(h)$ from $\mathcal{D}_L$ (§3.2)
end for
Estimate context-label moments $\widehat{\mathbf{R}}$ from anchors and $\mathcal{D}_U$ (Eq. 12)
for each word $w \in [V]$ do
    Solve the QP in Eq. 14 to obtain $\gamma_w$ from $\widehat{\mathbf{Q}}, \widehat{\mathbf{R}}$
end for
Estimate emissions $\mathbf{O}$ from $\mathbf{\Gamma}$ via Eq. 15
Estimate transitions $\mathbf{T}$ from $\mathcal{D}_L$
Return $\langle \mathbf{O}, \mathbf{T} \rangle$
```

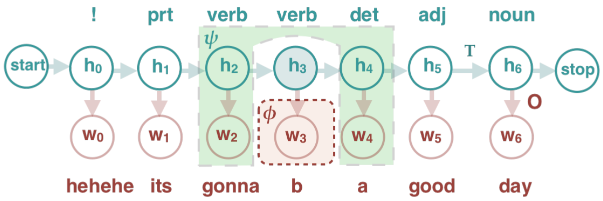

Figure 1: HMM, context (green) conditionally independent of present (red) w_ℓ given state h_ℓ.

and will be formally defined in §3.1. Such co-occurrence matrices are often collected in NLP, for various problems, ranging from dimensionality reduction of documents using latent semantic indexing (Deerwester et al., 1990; Landauer et al., 1998), distributional semantics (Schütze, 1998; Levy et al., 2015) and word embedding generation (Dhillon et al., 2015; Osborne et al., 2016). We can build such a moment matrix entirely from the unlabeled data $\mathcal{D}_U$. The same unlabeled data is used to build an estimate of a **context-label moment matrix** $\mathbf{R} \in \mathbb{R}^{C \times K}$, as explained in §3.3. This is done by first identifying words that are unambiguously associated with each label h, called **anchor words**, with the aid of a few labeled data; this is outlined in §3.2. Finally, given empirical estimates of $\mathbf{Q}$ and $\mathbf{R}$, we estimate the emission matrix $\mathbf{O}$ by solving a small optimization problem independently per word (§3.4). The transition matrix $\mathbf{T}$ is obtained directly from the labeled dataset $\mathcal{D}_L$ by maximizing the likelihood.

3.1 Moments of Contexts and Words

To formalize the notion of "context," we introduce the shorthand $\boldsymbol{Z}_\ell := \langle \boldsymbol{X}_{1:(\ell-1)}, \boldsymbol{X}_{(\ell+1):L} \rangle$. Importantly, the HMM in Eq. 1 entails the following conditional independence assumption: X_ℓ is conditionally independent of the surrounding context Z_ℓ given the hidden state H_ℓ. This is illustrated in Figure 1, using POS tagging as an example task.

We introduce a vector of **context features** $\boldsymbol{\psi}(\boldsymbol{Z}_\ell) \in \mathbb{R}^C$, which may look arbitrarily within the context $\boldsymbol{Z}_\ell$ (left or right), but not at X_ℓ itself. These features could be "one-hot" representations or other reduced-dimensionality embeddings (as described later in §5). Consider the word $w \in \mathcal{X}$ an instance of $X \equiv X_\ell$. A pivotal matrix in our formulation is the matrix $\mathbf{Q} \in \mathbb{R}^{C \times V}$, defined as:

$$Q_{c,w} := \mathbb{E}[\psi_c(\boldsymbol{Z}) \mid X = w]. \tag{3}$$

Expectations here are taken with respect to the probabilistic model in Eq. 1 that generates the data. The following quantities will also be necessary:

$$q_c := \mathbb{E}[\psi_c(\boldsymbol{Z})], \quad p_w := p(X = w). \tag{4}$$

Since all the variables in Eqs. 3–4 are observed, we can easily obtain empirical estimates by taking expectations over the unlabeled data:

$$\widehat{Q}_{c,w} = \frac{\sum_{x,\boldsymbol{z} \in \mathcal{D}_U} \psi_c(\boldsymbol{z}) \mathbb{1}(x = w)}{\sum_{x,\boldsymbol{z} \in \mathcal{D}_U} \mathbb{1}(x = w)}, \tag{5}$$

$$\widehat{q}_c = \sum_{x,\boldsymbol{z} \in \mathcal{D}_U} \psi_c(\boldsymbol{z}) \Big/ |\mathcal{D}_U|, \tag{6}$$

$$\widehat{p}_w = \sum_{x,\boldsymbol{z} \in \mathcal{D}_U} \mathbb{1}(x = w) \Big/ |\mathcal{D}_U|. \tag{7}$$

where we take $\mathbb{1}(x = w)$ to be the indicator for word w. Note that, under our modeling assumptions, $\mathbf{Q}$ decomposes in terms of its hidden states:

$$\mathbb{E}[\psi_c(\boldsymbol{Z}) \mid X = w] = \sum_{h \in \mathcal{H}} p(H = h \mid X = w) \mathbb{E}[\psi_c(\boldsymbol{Z}) \mid H = h] \tag{8}$$

The reason why this holds is that, as stated above, $\boldsymbol{Z}$ and X are conditionally independent given H.

3.2 Anchor Words

Following Arora et al. (2013) and Cohen and Collins (2014), we identify **anchor words** whose hidden state is assumed to be deterministic, regardless of context. In this work, we generalize this notion to more than one anchor word per label, for improved context estimates. This allows for more flexible forms of anchors with weak supervision. For each state $h \in \mathcal{H}$, let its set of anchor words be

$$\begin{aligned} \mathcal{A}(h) &= \{w \in \mathcal{X} : p(H = h \mid X = w) = 1\} \\ &= \left\{w \in \mathcal{X} : O_{w,h}{>}0 \wedge O_{w,h'}{=}0, \forall h' {\neq} h\right\}. \end{aligned} \tag{9}$$

That is, $\mathcal{A}(h)$ is the set of unambiguous words that always take the label h. This can be estimated from the labeled dataset $\mathcal{D}_L$ by collecting the most frequent unambiguous words for each label.

Algorithms for identifying $\mathcal{A}(h)$ from unlabeled data alone were proposed by Arora et al. (2013) and Zhou et al. (2014), with application to topic models. Our work differs in which we do not aim to discover anchor words from pure unlabeled data, but rather exploit the fact that small amounts of labeled data are commonly available in many NLP tasks—better anchors can be extracted easily from such small labeled datasets. In §5 we give a more detailed description of the selection process.

3.3 Moments of Contexts and Labels

We define the matrix $\mathbf{R} \in \mathbb{R}^{C \times K}$ as follows:

$$R_{c,h} := \mathbb{E}[\psi_c(\boldsymbol{Z}) \mid H = h]. \tag{10}$$

Since the expectation in Eq. 10 is conditioned on the (unobserved) label h, we cannot directly estimate it using moments of observed variables, as we do for $\mathbf{Q}$. However, if we have identified sets of anchor words for each label $h \in \mathcal{H}$, we have:

$$\begin{aligned} &\mathbb{E}[\psi_c(\boldsymbol{Z}) \mid X \in \mathcal{A}(h)] = \\ &= \sum_{h'} \mathbb{E}[\psi_c(\boldsymbol{Z}) \mid H = h'] \underbrace{p(H = h' \mid X \in \mathcal{A}(h))}_{=\mathbb{1}(h'=h)} \\ &= R_{c,h}. \end{aligned} \tag{11}$$

Therefore, given the set of anchor words $\mathcal{A}(h)$, the hth column of $\mathbf{R}$ can be estimated in a single pass over the unlabeled data, as follows:

$$\widehat{R}_{c,h} = \frac{\sum_{x,\boldsymbol{z} \in \mathcal{D}_U} \psi_c(\boldsymbol{z}) \mathbb{1}(x \in \mathcal{A}(h))}{\sum_{x,\boldsymbol{z} \in \mathcal{D}_U} \mathbb{1}(x \in \mathcal{A}(h))} \tag{12}$$

3.4 Emission Distributions

We can now put all the ingredients above together to estimate the emission probability matrix $\mathbf{O}$. The procedure we propose here is computationally very efficient, since only one pass is required over the unlabeled data, to collect the co-occurrence statistics $\widehat{\mathbf{Q}}$ and $\widehat{\mathbf{R}}$. The emissions will be estimated from these moments by solving a small problem independently for each word. Unlike EM and self-training, no decoding is necessary, only counting and normalizing; and unlike label propagation methods, there is requirement to build a graph with the unlabeled data.

The crux of our method is the decomposition in Eq. 8, which is combined with the one-to-one correspondence between labels h and anchor words $\mathcal{A}(h)$. We can rewrite Eq. 8 as:

$$Q_{c,w} = \sum_h R_{c,h} \; p(H = h \mid X = w). \tag{13}$$

In matrix notation, we have $\mathbf{Q} = \mathbf{R}\boldsymbol{\Gamma}$, where $\boldsymbol{\Gamma} \in \mathbb{R}^{K \times V}$ is defined as $\Gamma_{h,w} := p(H = h \mid X = w)$.

If we had infinite unlabeled data, our moment estimates $\widehat{\mathbf{Q}}$ and $\widehat{\mathbf{R}}$ would be perfect and we could solve the system of equations in Eq. 13 to obtain $\boldsymbol{\Gamma}$ exactly. Since we have finite data, we resort to a least squares solution. This corresponds to solving a simple quadratic program (QP) per word, independent from all the other words, as follows. Denote by $\mathbf{q}_w := \mathbb{E}[\boldsymbol{\psi}(\boldsymbol{Z}) \mid X = w] \in \mathbb{R}^C$ and by $\boldsymbol{\gamma}_w := p(H = \cdot \mid X = w) \in \mathbb{R}^K$ the wth columns of $\mathbf{Q}$ and $\boldsymbol{\Gamma}$, respectively. We estimate the latter distribution following Arora et al. (2013):

$$\begin{aligned} \widehat{\boldsymbol{\gamma}}_w \;=\; &\arg\min_{\boldsymbol{\gamma}_w} \quad \|\mathbf{q}_w - \mathbf{R}\boldsymbol{\gamma}_w\|_2^2 \\ &\text{s.t.} \qquad \mathbf{1}^\top \boldsymbol{\gamma}_w = 1, \; \boldsymbol{\gamma}_w \geq \mathbf{0}. \end{aligned} \tag{14}$$

Note that this QP is very small—it has only K variables—hence, we can solve it very quickly (1.7 ms on average, in Gurobi, with $K = 12$).

Given the probability tables for $p(H = h \mid X = w)$, we can estimate the emission probabilities $\mathbf{O}$ by direct application of Bayes rule:

$$\widehat{O}_{w,h} = \frac{p(H = h \mid X = w) \times p(X = w)}{p(H = h)} \tag{15}$$

$$= \frac{\widehat{\gamma}_{w,c} \times \overbrace{\widehat{p}_w}^{\text{Eq. 7}}}{\sum_{w'} \widehat{\gamma}_{w',c} \times \widehat{p}_{w'}}. \tag{16}$$

These parameters are guaranteed to lie in the probability simplex, avoiding the need of heuristics for dealing with "negative" and "unnormalized" probabilities required by prior work in spectral learning (Cohen et al., 2013).

3.5 Transition Distributions

It remains to estimate the transition matrix $\mathbf{T}$. For the problems tackled in this paper, the number of labels K is small, compared to the vocabulary size V. The transition matrix has only $O(K^2)$ degrees of freedom, and we found it effective to estimate it using the labeled sequences in $\mathcal{D}_L$ alone, without any refinement. This was done by smoothed maximum likelihood estimation on the labeled data, which boils down to counting occurrences of consecutive labels, applying add-one smoothing to avoid zero probabilities for unobserved transitions, and normalizing.

For problems with numerous labels, a possible alternative is the composite likelihood method (Chaganty and Liang, 2014). Given $\widehat{\mathbf{O}}$, the maximization of the composite log-likelihood function leads to a convex optimization problem that can be efficiently optimized with an EM algorithm. A similar procedure was carried out by Cohen and Collins (2014).[2]

4 Feature-Based Emissions

Next, we extend our method to estimate the parameters of the FHMM in §2.2. Other than contextual features $\boldsymbol{\psi}(\boldsymbol{Z}) \in \mathbb{R}^C$, we also assume a feature encoding function for words, $\boldsymbol{\phi}(X) \in \mathbb{R}^W$. Our framework, illustrated in Algorithm 2, allows for both discrete and continuous word and context features. Lines 2–5 are the same as in Algorithm 1, replacing word occurrences with expected values of word features (we redefine $\mathbf{Q}$ and $\boldsymbol{\Gamma}$ to cope with features instead of words). The main difference with respect to Algorithm 1 is that we do not estimate emission probabilities; rather, we first estimate the **mean parameters** (feature expectations $\mathbb{E}[\boldsymbol{\phi}(X) \mid H = h]$), by solving one QP for each emission feature; and then we solve a convex optimization problem, for each label h, to recover the log-linear weights over emission features (called **canonical parameters**).

Algorithm 2 Semi-Supervised Learning of Feature-Based HMMs with Moments

```
Input: Labeled dataset $\mathcal{D}_L$, unlabeled dataset $\mathcal{D}_U$
Output: Emission log-linear parameters $\mathbf{\Theta}$ and transitions $\mathbf{T}$
 1: Estimate context-word moments $\widehat{\mathbf{Q}}$ from $\mathcal{D}_U$ (Eq. 20)
 2: for each label $h \in \mathcal{H}$ do
 3:     Extract set of anchor words $\mathcal{A}(h)$ from $\mathcal{D}_L$ (§3.2)
 4: end for
 5: Estimate context-label moments $\widehat{\mathbf{R}}$ from the anchors and $\mathcal{D}_U$ (Eq. 12)
 6: for each word feature $j \in [W]$ do
 7:     Solve the QP in Eq. 22 to obtain $\boldsymbol{\gamma}_j$ from $\widehat{\mathbf{Q}}, \widehat{\mathbf{R}}$
 8: end for
 9: for each label $h \in \mathcal{H}$ do
10:     Estimate the mean parameters $\boldsymbol{\mu}_h$ from $\mathbf{\Gamma}$ (Eq. 24)
11:     Estimate the canonical parameters $\boldsymbol{\theta}_h$ from $\boldsymbol{\mu}_h$ by solving Eq. 25
12: end for
13: Estimate transitions $\mathbf{T}$ from $\mathcal{D}_L$
14: Return $\langle \mathbf{\Theta}, \mathbf{T} \rangle$
```

4.1 Estimation of Mean Parameters

First of all, we replace word probabilities by expectations over word features. We redefine the matrix $\mathbf{\Gamma} \in \mathbb{R}^{K \times W}$ as follows:

$$\Gamma_{h,j} := \frac{p(H=h) \times \mathbb{E}[\phi_j(X) \mid H=h]}{\mathbb{E}[\phi_j(X)]}. \qquad (17)$$

Note that, with one-hot word features, we have $\mathbb{E}[\phi_w(X) \mid H = h] = P(X = w \mid H = h)$, $\mathbb{E}[\phi_w(X)] = p(X = w)$, and therefore $\Gamma_{h,w} = p(H = h \mid X = w)$, so this can be regarded as a generalization of the framework in §3.4.

Second, we redefine the context-word moment matrix $\mathbf{Q}$ as the following matrix in $\mathbb{R}^{C \times W}$:

$$Q_{c,j} = \mathbb{E}\left[\psi_c(\boldsymbol{Z}) \times \phi_j(X)\right] / \mathbb{E}[\phi_j(X)]. \qquad (18)$$

Again, note that we recover the previous $\mathbf{Q}$ if we use one-hot word features. We then have the following generalization of Eq. 13:

$$\mathbb{E}\left[\psi_c(\boldsymbol{Z}) \times \phi_j(X)\right] / \mathbb{E}[\phi_j(X)] = \qquad (19)$$
$$\textstyle\sum_h \mathbb{E}\left[\psi_c(\boldsymbol{Z}) \mid H=h\right] \frac{P(H=h)\mathbb{E}[\phi_j(X) \mid H=h]}{\mathbb{E}[\phi_j(X)]},$$

[2] In preliminary experiments, the compositional likelihood method was not competitive with estimating the transition matrices directly from the labeled data, on the datasets described in §6; results are omitted due to lack of space. However, this may be a viable alternative if there is no labeled data and the anchors are extracted from gazetteers or a dictionary.

or, in matrix notation, $\mathbf{Q} = \mathbf{R}\boldsymbol{\Gamma}$.

Again, matrices $\mathbf{Q}$ and $\mathbf{R}$ can be estimated from data by collecting empirical feature expectations over unlabeled sequences of observations. For $\mathbf{R}$ use Eq. 12 with no change; for $\mathbf{Q}$ replace Eq. 5 by

$$\widehat{Q}_{c,j} = \sum\nolimits_{x,\boldsymbol{z}\in\mathcal{D}_U} \psi_c(\boldsymbol{z})\phi_j(x) \Big/ \sum\nolimits_{x,\boldsymbol{z}\in\mathcal{D}_U} \phi_j(x). \quad (20)$$

Let $\mathbf{q}_j \in \mathbb{R}^C$ and $\boldsymbol{\gamma}_j \in \mathbb{R}^K$ be columns of $\widehat{\mathbf{Q}}$ and $\widehat{\boldsymbol{\Gamma}}$, respectively. Note that we must have

$$\mathbf{1}^\top \boldsymbol{\gamma}_j = \sum_h \tfrac{P(H=h)\mathbb{E}[\phi_j(X)|H=h]}{\mathbb{E}[\phi_j(X)]} = 1, \qquad (21)$$

since $\mathbb{E}[\phi_j(X)] = \sum_h P(H = h)\mathbb{E}\left[\phi_j(X) \mid H = h\right]$. We rewrite the QP to minimize the squared difference for each dimension j independently:

$$\widehat{\boldsymbol{\gamma}}_j \;=\; \arg\min_{\boldsymbol{\gamma}_j} \;\; \left\|\mathbf{q}_j - \mathbf{R}\boldsymbol{\gamma}_j\right\|_2^2 \;\text{ s.t. }\; \mathbf{1}^\top\boldsymbol{\gamma}_j = 1. \qquad (22)$$

Note that, if $\boldsymbol{\phi}(x) \geq \mathbf{0}$ for all $x \in \mathcal{X}$, then we must have $\boldsymbol{\gamma}_j \geq \mathbf{0}$, and therefore we may impose this inequality as an additional constraint.

Let $\bar{\boldsymbol{\gamma}} \in \mathbb{R}^K$ be the vector of state probabilities, with entries $\bar{\gamma}_h := p(H = h)$ for $h \in \mathcal{H}$. This vector can also be recovered from the unlabeled dataset and the set of anchors, by solving another QP that aggregates information for all words:

$$\bar{\boldsymbol{\gamma}} \;=\; \arg\min_{\bar{\boldsymbol{\gamma}}} \;\; \|\bar{\boldsymbol{q}} - \mathbf{R}\bar{\boldsymbol{\gamma}}\|_2^2 \;\text{ s.t. }\; \mathbf{1}^\top\bar{\boldsymbol{\gamma}} = 1,\; \bar{\boldsymbol{\gamma}} \geq \mathbf{0}. \qquad (23)$$

where $\bar{\boldsymbol{q}} := \widehat{\mathbb{E}}[\boldsymbol{\psi}(\boldsymbol{Z})] \in \mathbb{R}^C$ is the vector whose entries are defined in Eq. 6.

Let $\boldsymbol{\mu}_h := \mathbb{E}[\boldsymbol{\phi}(X) \mid H = h] \in \mathbb{R}^W$ be the mean parameters of the distribution for each state h. These parameters are computed by solving W independent QPs (Eq. 22), yielding the matrix $\boldsymbol{\Gamma}$ defined in Eq. 17, and then applying the formula:

$$\mu_{h,j} = \Gamma_{j,h} \times \mathbb{E}[\phi_j(X)] / \bar{\gamma}_h, \qquad (24)$$

with $\bar{\gamma}_h = p(H = h)$ estimated as in Eq. 23.

4.2 Estimation of Canonical Parameters

To compute a mapping from mean parameters $\boldsymbol{\mu}_h$ to canonical parameters $\boldsymbol{\theta}_h$, we use the well-known Fenchel-Legendre duality between the entropy and the log-partition function (Wainwright and Jordan, 2008). Namely, we need to solve the following convex optimization problem:

$$\widehat{\boldsymbol{\theta}}_h = \arg\max_{\boldsymbol{\theta}_h} \;\; \boldsymbol{\theta}_h^\top \boldsymbol{\mu}_h - \log Z(\boldsymbol{\theta}_h) + \epsilon\|\boldsymbol{\theta}_h\|, \quad (25)$$

where ϵ is a regularization constant.[3] In practice, this regularization is important, since it prevents $\boldsymbol{\theta}_h$ from growing unbounded whenever $\boldsymbol{\mu}_h$ falls outside the marginal polytope of possible mean parameters. We solve Eq. 25 with the limited-memory BFGS algorithm (Liu and Nocedal, 1989).

5 Method Improvements

In this section we detail three improvements to our moment-based method that had a practical impact.

Supervised Regularization. We add a supervised penalty term to Eq. 22 to keep the label posteriors $\boldsymbol{\gamma}_j$ close to the label posteriors estimated in the labeled set, $\boldsymbol{\gamma}'_j$, for every feature $j \in [W]$. The regularized least-squares problem becomes:

$$\begin{aligned} &\min_{\boldsymbol{\gamma}_j} \,(1-\lambda)\|\mathbf{q}_j - \mathbf{R}\boldsymbol{\gamma}_j\|^2 + \lambda\|\boldsymbol{\gamma}_j - \boldsymbol{\gamma}'_j\|^2 \\ &\text{s.t. }\; \mathbf{1}^\top\boldsymbol{\gamma}_j = 1. \end{aligned} \qquad (26)$$

CCA Projections. A one-hot feature representation of words and contexts has the disadvantage that it grows with the vocabulary size, making the moment matrix $\mathbf{Q}$ too sparse. The number of contextual features and words can grow rapidly on large text corpora. Similarly to Cohen and Collins (2014) and Dhillon et al. (2015), we use canonical correlation analysis (CCA) to reduce the dimension of these vectors. We use CCA to form low-dimensional projection matrices for features of words $\mathbf{P}_W \in \mathbb{R}^{W\times D}$ and features of contexts $\mathbf{P}_C \in \mathbb{R}^{C\times D}$, with $D \ll \min\{W, C\}$. We use these projections on the original feature vectors and replace the these vectors with their projections.

Selecting Anchors. We collect counts of each word-label pair, and select up to 500 anchors with high conditional probability on the anchoring state $\widehat{p}(h \mid w)$. We tuned the probability threshold to

[3] As shown by Xiaojin Zhu (1999) and Yasemin Altun (2006), this regularization is equivalent, in the dual, to a "soft" constraint $\|\mathbb{E}_{\boldsymbol{\theta}_h}[\boldsymbol{\phi}(X) \mid H = h] - \boldsymbol{\mu}_h\|_2 \leq \epsilon$, as opposed to a strict equality.

select the anchors on the validation set, using steps of 0.1 in the unit interval, and making sure that all tags have at least one anchor. We also considered a frequency threshold, constraining anchors to occur more than 500 times in the unlabeled corpus, and four times in the labeled corpus. Note that past work used a single anchor word per state (i.e., $|\mathcal{A}(h)| = 1$). We found that much better results are obtained when $|\mathcal{A}(h)| \gg 1$, as choosing more anchors increases the number of samples used to estimate the context-label moment matrix $\widehat{\mathbf{R}}$, reducing noise.

6 Experiments

We evaluated our method on two tasks: POS tagging of Twitter text (in English), and POS tagging for a low-resource language (Malagasy). For all the experiments, we used the universal POS tagset (Petrov et al., 2012), which consists of $K = 12$ tags. We compared our method against supervised baselines (HMM and FHMM), which use the labeled data only, and two semi-supervised baselines that exploit the unlabeled data: self-training and EM. For the Twitter experiments, we also evaluated a stacked architecture in which we derived features from our model's predictions to improve a state-of-the-art POS tagger (MEMM).[4]

6.1 Twitter POS Tagging

For the Twitter experiment, we used the *Oct27* dataset of Gimpel et al. (2011), with the provided partitions (1,000 tweets for training and 328 for validation), and tested on the *Daily547* dataset (547 tweets). Anchor words were selected from the training partition as described in §5. We used 2.7M unlabeled tweets (O'Connor et al., 2010) to train the semi-supervised methods, filtering the English tweets as in Lui and Baldwin (2012), tokenizing them as in Owoputi et al. (2013), and normalizing at-mentions, URLs, and emoticons.

We used as word features $\phi(X)$ the word iself, as well as binary features for capitalization, titles, and digits (Berg-Kirkpatrick et al., 2010), the word shape, and the Unicode class of each character. Similarly to Owoputi et al. (2013), we also used suffixes and prefixes (up to length 3), and Twitter-specific features: whether the word starts with @, #, or *http://*. As contextual features $\psi(Z)$, we derive analogous features for the preceding and following words, before reducing dimensionality with CCA. We collect feature expectations for words and contexts that occur more than 20 times in the unlabeled corpus. We tuned hyperparameters on the development set: the supervised interpolation coefficient in Eq. 26, $\lambda \in \{0, 0.1, \ldots, \underline{1.0}\}$, and, for all systems, the regularization coefficient $\epsilon \in \{\underline{0.0001}, 0.001, 0.01, 0.1, 1, 10\}$. (Underlines indicate selected values.) The former controls how much we rely on the supervised vs. unsupervised estimates. For $\lambda = 1.0$ we used supervised estimates only for words that occur in the labeled corpus, all the remaining words rely solely on unsupervised estimates.

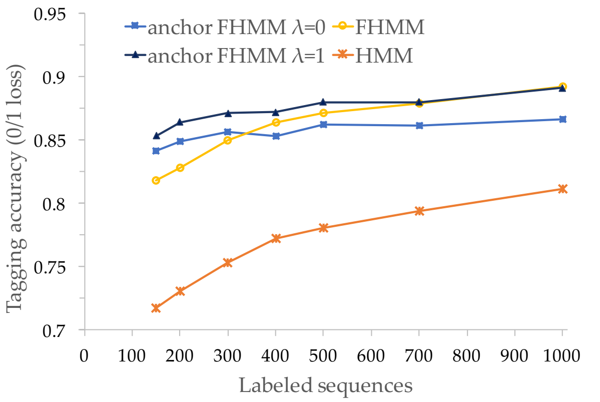

Figure 2: POS tagging accuracy in the Twitter data versus the number of labeled training sequences.

Varying supervision. Figure 2 compares the learning curves of our anchor-word method for the FHMM with the supervised baselines. We show the performance of the anchor methods without interpolation ($\lambda = 0$), and with supervised interpolation coefficient ($\lambda = 1$). When the amount of supervision is small, our method with and without interpolation outperforms all the supervised baselines. This improvement is gradually attenuated when more labeled sequences are used, with the supervised FHMM catching up when the full labeled dataset is used. The best model $\lambda = 1.0$ relies on supervised estimates for words that occur in the labeled corpus, and on anchor estimates for words that occur only in the unlabeled corpus. The unregular-

[4] http://www.ark.cs.cmu.edu/TweetNLP/

	HMM		FHMM	
Models / #sequences	150	1000	150	1000
Supervised baseline				
HMM	71.7	81.1	81.8	89.1
Semi-supervised baselines				
EM	77.2	83.1	81.8	89.1
self-training	78.2	86.1	83.4	**89.4**
Anchor Models				
anchors, $\lambda = 0.0$	83.0	85.5	84.1	86.7
anchors, $\lambda = 1.0$	**84.3**	**88.0**	**85.3**	89.1

Table 1: Tagging accuracies on Twitter. Shown are the supervised and semi-supervised baselines, and our moment-based method, trained with 150 training labeled sequences, and the full labeled corpus (1000 sequences).

ized model $\lambda = 0.0$ relies solely on unsupervised estimates given the set of anchors.

Semi-supervised comparison. Next, we compare our method to two other semi-supervised baselines, using both HMMs and FHMMs: EM and self-training. EM requires decoding and counting in multiple passes over the full unlabeled corpus. We initialized the parameters with the supervised estimates, and selected the iteration with the best accuracy on the development set.[5] The self-training baseline uses the supervised system to tag the unlabeled data, and then retrains on all the data.

Results are shown in Table 1. We observe that, for small amounts of labeled data (150 tweets), our method outperforms all the supervised and semi-supervised baselines, yielding accuracies 6.1 points above the best semi-supervised baseline for a simple HMM, and 1.9 points above for the FHMM. With more labeled data (1,000 instances), our method outperforms all the baselines for the HMM, but not with the more sophisticated FHMM, in which our accuracies are 0.3 points below the self-training method.[6] These results suggest that our method is more effective when the amount of labeled data is small.

	150	1000
MEMM (same+clusters)	89.57	**93.36**
MEMM (same+clusters+posteriors)	**91.14**	93.18
MEMM (all+clusters)	91.55	**94.17**
MEMM (all+clusters+posteriors)	**92.06**	94.11

Table 2: Tagging accuracy for the MEMM POS tagger of Owoputi et al. (2013) with additional features from our model's posteriors.

Stacking features. We also evaluated a stacked architecture in which we use our model's predictions as an additional feature to improve the state-of-the-art Twitter POS tagger of Owoputi et al. (2013). This system is based on a semi-supervised discriminative model with Brown cluster features (Brown et al., 1992). We provide results using their full set of features (*all*), and using the same set of features in our anchor model (*same*). We compare tagging accuracy on a model with these features plus Brown clusters (*+clusters*) against a model that also incorporates the posteriors from the anchor method as an additional feature in the MEMM (*+clusters+posteriors*). The results in Table 2 show that using our model's posteriors are beneficial in the small labeled case, but not if the entire labeled data is used.

Runtime comparison. The training time of anchor FHMM is 3.8h (hours), for self-training HMM 10.3h, for EM HMM 14.9h and for Twitter MEMM (all+clusters) 42h. As such, the anchor method is much more efficient than all the baselines because it requires a single pass over the corpus to collect the moment statistics, followed by the QPs, without the need to decode the unlabeled data. EM and the Brown clustering method (the latter used to extract features for the Twitter MEMM) require several passes over the data; and the self-training method involves decoding the full unlabeled corpus, which is expensive when the corpus is large. Our analysis adds to previous evidence that spectral methods are more scalable than learning algorithms that require inference (Parikh et al., 2012; Cohen et al., 2013).

6.2 Malagasy POS Tagging

For the Malagasy experiment, we used the small labeled dataset from Garrette et al. (2013), which consists of 176 sentences and 4,230 tokens. We also make use of their tag dictionaries with 2,773 types

[5] The FHMM with EM did not perform better than the supervised baseline, so we consider the initial value as the best accuracy under this model.

[6] According to a word-level paired Kolmogorov-Smirnov test, for the FHMM with 1,000 tweets, the self-training method outperforms the other methods with statistical significance at $p < 0.01$; and for the FHMM with 150 tweets the anchor-based and self-training methods outperform the other baselines with the same p-value. Our best HMM outperforms the other baselines at a significance level of $p < 0.01$ for 150 and 1000 sequences.

Models	Accuracies
supervised FHMM	90.5
EM FHMM	90.5
self-training FHMM	88.7
anchors FHMM (token), λ=1.0	89.4
anchors FHMM (type+token), λ=1.0	**90.9**

Table 3: Tagging accuracies for the Malagasy dataset.

and 23 tags, and their unlabeled data (43.6K sequences, 777K tokens). We converted all the original POS tags to universal tags using the mapping proposed in Garrette et al. (2013).

Table 3 compares our method with semi-supervised EM and self-training, for the FHMM.We tested two supervision settings: token only, and type+token annotations, analogous to Garrette et al. (2013). The anchor method outperformed the baselines when both type and token annotations were used to build the set of anchor words.[7]

7 Conclusion

We proposed an efficient semi-supervised sequence labeling method using a generative log-linear model. We use contextual information from a set of *anchor* observations to disambiguate state, and build a weakly supervised method from this set. Our method outperforms other supervised and semi-supervised methods, with small supervision in POS-tagging for Malagasy, a scarcely annotated language, and for Twitter. Our anchor method is most competitive for learning with large amounts of unlabeled data, under weak supervision, while training an order of magnitude faster than any of the baselines.

Acknowledgments

Support for this research was provided by Fundação para a Ciência e Tecnologia (FCT) through the CMU Portugal Program under grant SFRH/BD/52015/2012. This work has also been partially supported by the European Union under H2020 project SUMMA, grant 688139, and by FCT, through contracts UID/EEA/50008/2013, through the LearnBig project (PTDC/EEI-SII/7092/2014), and the GoLocal project (grant CMUPERI/TIC/0046/2014).

[7] Note that the accuracies are not directly comparable to Garrette et al. (2013), who use a different tag set. However, our supervised baseline trained on those tags is already superior to the best semi-supervised system in Garrette et al. (2013), as we get 86.9% against the 81.2% reported in Garrette et al. (2013) using their tagset.

References

Sanjeev Arora, Rong Ge, Yoni Halpern, David Mimno, David Sontag Ankur Moitra, Yichen Wu, and Michael Zhu. 2013. A practical algorithm for topic modeling with provable guarantees. In *Proc. of International Conference of Machine Learning*.

Raphaël Bailly, Xavier Carreras, Franco M. Luque, and Ariadna Quattoni. 2013. Unsupervised spectral learning of WCFG as low-rank matrix completion. In *Proc. of Empirical Methods in Natural Language Processing*, pages 624–635.

Borja Balle and Mehryar Mohri. 2012. Spectral learning of general weighted automata via constrained matrix completion. In *Advances in Neural Information Processing Systems*, pages 2168–2176.

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

Peter F. Brown, Peter V. de Souza, 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.

Arun T. Chaganty and Percy Liang. 2014. Estimating latent-variable graphical models using moments and likelihoods. In *Proc. of International Conference on Machine Learning*.

Shay B. Cohen and Michael Collins. 2014. A provably correct learning algorithm for latent-variable PCFGs. In *Proc. of Association for Computational Linguistics*.

Shay B. Cohen, Karl Stratos, Michael Collins, Dean P. Foster, and Lyle Ungar. 2013. Experiments with spectral learning of latent-variable PCFGs. In *Proc. of North American Association of Computational Linguistics*.

Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407.

Paramveer S. Dhillon, Dean P. Foster, and Lyle H. Ungar. 2015. Eigenwords: Spectral word embeddings. *Journal of Machine Learning Research*, 16:3035–3078.

Dan Garrette, Jason Mielens, and Jason Baldridge. 2013. Real-world semi-supervised learning of POS-taggers for low-resource languages. In *Proc. of Association for Computational Linguistics*.

Gimpel, Schneider, O'Connor, Das, Mills, Eisenstein, Heilman, Yogatama, Flanigan, and Smith. 2011. Part-of-speech tagging for twitter: Annotation, features, and experiments. In *Proc. of Association of 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.

Thomas K. Landauer, Peter W. Foltz, and Darrell Laham. 1998. An introduction to latent semantic analysis. *Discourse Processes 25*, pages 259–284.

Omer Levy, Yoav Goldberg, and Ido Dagan. 2015. Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics*, 3:211–225.

Dong Liu and Jorge Nocedal. 1989. On the limited memory bfgs method for large scale optimization. *Mathematical Programming*, 45:503–528.

Marco Lui and Timothy Baldwin. 2012. langid.py: An off-the-shelf language identification tool. In *Proc. of Association of Computational Linguistics System Demonstrations*, pages 25–30.

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

Thang Nguyen, Jordan Boyd-Graber, Jeff Lund, Kevin Seppi, and Eric Ringger. 2015. Is your anchor going up or down? Fast and accurate supervised topic models. In *Proc. of North American Association for Computational Linguistics*.

Brendan O'Connor, Michel Krieger, and David Ahn. 2010. TweetMotif: Exploratory search and topic summarization for Twitter. In *Proc. of AAAI Conference on Weblogs and Social Media*.

Dominique Osborne, Shashi Narayan, and Shay B. Cohen. 2016. Encoding prior knowledge with eigenword embeddings. *Transactions of the Association of Computational Linguistics*.

Olutobi Owoputi, Brendan O'Connor, Chris Dyer, Kevin Gimpel, Nathan Schneider, and Noah A Smith. 2013. Improved part-of-speech tagging for online conversational text with word clusters. In *Proc. of North American Association for Computational Linguistics*.

Ankur P. Parikh, Lee Song, Mariya Ishteva, Gabi Teodoru, and Eric P. Xing. 2012. A spectral algorithm for latent junction trees. In *Proc. of Uncertainty in Artificial Intelligence*.

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

Ariadna Quattoni, Borja Balle, Xavier Carreras, and Amir Globerson. 2014. Spectral regularization for max-margin sequence tagging. In *Proc. of International Conference of Machine Learning*, pages 1710–1718.

Hinrich Schütze. 1998. Automatic word sense discrimination. *Computational Linguistics*, 24(1):97–123.

Noah A. Smith and Jason Eisner. 2005. Contrastive estimation: Training log-linear models on unlabeled data. In *Proc. of Association for Computational Linguistics*, pages 354–362.

Karl Stratos, Alexander M. Rush, Shay B. Cohen, and Michael Collins. 2013. Spectral learning of refinement hmms. In *Proc. of Computational Natural Language Learning*.

Karl Stratos, Michael Collins, and Daniel J. Hsu. 2016. Unsupervised part-of-speech tagging with anchor hidden markov models. *Transactions of the Association for Computational Linguistics*, 4:245–257.

Martin J. Wainwright and Michael I. Jordan. 2008. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(2):1–305.

Roni Rosenfeld Xiaojin Zhu, Stanley F. Chen. 1999. Linguistic features for whole sentence maximum entropy language models. In *European Conference on Speech Communication and Technology*.

Alexander J. Smola Yasemin Altun. 2006. Unifying divergence minimization and statistical inference via convex duality. In *Proc. of Conference on Learning Theory*.

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*.