Mutual Information Maximization for Simple and Accurate Part-Of-Speech Induction

Karl Stratos
Toyota Technological Institute at Chicago
stratos@ttic.edu

Abstract

We address part-of-speech (POS) induction by maximizing the mutual information between the induced label and its context. We focus on two training objectives that are amenable to stochastic gradient descent (SGD): a novel generalization of the classical Brown clustering objective and a recently proposed variational lower bound. While both objectives are subject to noise in gradient updates, we show through analysis and experiments that the variational lower bound is robust whereas the generalized Brown objective is vulnerable. We obtain strong performance on a multitude of datasets and languages with a simple architecture that encodes morphology and context.

1 Introduction

We consider information theoretic objectives for POS induction, an important unsupervised learning problem in computational linguistics (Christodoulopoulos et al., 2010). The idea is to make the induced label syntactically informative by maximizing its mutual information with respect to local context. Mutual information has long been a workhorse in the development of NLP techniques, for instance the classical Brown clustering algorithm (Brown et al., 1992). But its role in today's deep learning paradigm is less clear and a subject of active investigation (Belghazi et al., 2018; Oord et al., 2018).

We focus on fully differentiable objectives that can be plugged into an automatic differentiation system and efficiently optimized by SGD. Specifically, we investigate two training objectives. The first is a novel generalization of the Brown clustering objective obtained by relaxing the hard clustering constraint. The second is a recently proposed variational lower bound on mutual information (McAllester, 2018).

A main challenge in optimizing these objectives is the difficulty of stochastic optimization. Each objective involves entropy estimation which is a nonlinear function of all data and does not decompose over individual instances. This makes the gradients estimated on minibatches inconsistent with the true gradient estimated from the entire dataset. To our surprise, in practice we are able to optimize the variational objective effectively but not the generalized Brown objective. We analyze the estimated gradients and show that the inconsistency error is only logarithmic in the former but linear in the latter.

We validate our approach on POS induction by attaining strong performance on a multitude of datasets and languages. Our simple architecture that encodes morphology and context reaches up to 80.1 many-to-one accuracy on the 45-tag Penn WSJ dataset and achieves 4.7% absolute improvement to the previous best result on the universal treebank. Unlike previous works, our model does not rely on computationally expensive structured inference or hand-crafted features.

2 Background

2.1 Information Theory

Mutual information. Mutual information between two random variables measures the amount of information gained about one variable by observing the other. Unlike the Pearson correlation coefficient which only captures the degree of linear relationship, mutual information captures any nonlinear statistical dependencies (Kinney and Atwal, 2014).

Formally, the mutual information between discrete random variables X, Y with a joint distribution p is the KL divergence between the joint distribution $p(x, y)$ and the product distribution

Proceedings of NAACL-HLT 2019, pages 1095–1104
Minneapolis, Minnesota, June 2 - June 7, 2019.

$p(x)p(y)$ over X, Y:

$$I(X,Y) = \sum_{x,y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)}$$
$$= \mathop{\mathbf{E}}_{(x,y)\sim p}\left[\log \frac{p(x,y)}{p(x)p(y)}\right] \quad (1)$$

It is thus nonnegative and zero iff X and Y are independent. We assume that the marginals $p(x)$ and $p(y)$ are nonzero.

It is insightful to write mutual information in terms of entropy. The entropy of X is

$$H(X) = -\sum_{x} p(x) \log p(x) = \mathop{\mathbf{E}}_{x\sim p}\left[\log \frac{1}{p(x)}\right]$$

corresponding to the number of bits for encoding the behavior of X under p.[1] The entropy of X given the information that Y equals y is

$$H(X|Y=y) = -\sum_{x} p(x|y) \log p(x|y)$$

Taking expectation over Y yields the conditional entropy of X given Y:

$$H(X|Y) = \sum_{y} p(y) \left(-\sum_{x} p(x|y) \log p(x|y)\right)$$
$$= \mathop{\mathbf{E}}_{(x,y)\sim p}\left[\log \frac{1}{p(x|y)}\right] \quad (2)$$

By manipulating the terms in mutual information, we can write

$$I(X,Y) = \mathop{\mathbf{E}}_{x\sim p}\left[\log \frac{1}{p(x)}\right] - \mathop{\mathbf{E}}_{(x,y)\sim p}\left[\log \frac{1}{p(x|y)}\right]$$
$$= H(X) - H(X|Y)$$

which expresses the amount of information on X gained by observing Y. Switching X and Y shows that $I(X,Y) = H(Y) - H(Y|X)$.

Cross entropy. If p and q are full-support distributions over the same discrete set, the cross entropy between p and q is the asymmetric quantity

$$H(p,q) = -\sum_{x} p(x) \log q(x) = \mathop{\mathbf{E}}_{x\sim p}\left[\log \frac{1}{q(x)}\right]$$

corresponding to the number of bits for encoding the behavior of X under p by using q. It is revealing to write entropy in terms of KL divergence. Multiplying the term inside the log by $p(x)/p(x)$ we derive

$$H(p,q) = \mathop{\mathbf{E}}_{x\sim p}\left[\log \frac{1}{p(x)}\right] + \mathop{\mathbf{E}}_{x\sim p}\left[\log \frac{p(x)}{q(x)}\right]$$
$$= H(p) + D_{\mathrm{KL}}(p||q)$$

Thus $H(p,q) \geq H(p)$ with equality iff $p = q$.

2.2 Brown Clustering

Our primary inspiration comes from Brown clustering (Brown et al., 1992), a celebrated word clustering technique that had been greatly influential in unsupervised and semi-supervised NLP long before continuous representations based on neural networks were popularized. It finds a clustering $C : V \to [m]$ of the vocabulary V into m classes by optimizing the mutual information between the clusters of a random bigram (X,Y). Given a corpus of N words $(x_1 \dots x_N)$, it assumes a uniform distribution over consecutive word pairs (x_{i-1}, x_i) and optimizes the following empirical objective

$$\max_{C:V\to[m]} \sum_{c,c'\in[m]} \frac{\#(c,c')}{N} \log \left(\frac{\#(c,c')N}{\#(c)\#(c')}\right) \quad (3)$$

where $\#(c,c')$ denotes the number of occurrences of the cluster pair (c,c') under C. While this optimization is intractable, Brown et al. (1992) derive an effective heuristic that 1. initializes m most frequent words as singleton clusters and 2. repeatedly merges a pair of clusters that yields the smallest decrease in mutual information. The resulting clusters have been useful in many applications (Koo et al., 2008; Owoputi et al., 2013) and has remained a strong baseline for POS induction decades later (Christodoulopoulos et al., 2010). But the approach is tied to highly nontrivial combinatorial optimization tailored for the specific problem and difficult to scale/generalize.

3 Objectives

In the remainder of the paper, we assume discrete random variables $(X,Y) \in \mathcal{X} \times \mathcal{Y}$ with a joint distribution D that represent naturally co-occurring observations. In POS induction experiments, we will set D to be a context-word distribution where Y is a random word and X is the surrounding context of Y (thus $\mathcal{Y}$ is the vocabulary and $\mathcal{X}$ is the space of all possible contexts). Let m be the number of labels to induce. We introduce a pair of trainable classifiers that define conditional label distributions $p(z|x)$ and $q(z|y)$ for all $x \in \mathcal{X}$,

[1] The paper will always assume log base 2 to accommodate the bit interpretation.

$y \in \mathcal{Y}$, and $z \in [m]$. For instance, $p(\cdot|y)$ can be the output of a softmax layer on some transformation of y.

Our goal is to learn these classifiers without observing the latent variable z by optimizing an appropriate objective. For training data, we assume N iid samples $(x_1, y_1) \ldots (x_N, y_N) \sim D$.

3.1 Generalized Brown Objective

Our first attempt is to maximize the mutual information between the predictions of p and q. Intuitively, this encourages p and q to agree on some annotation scheme (up to a permutation of labels), modeling the dynamics of inter-annotator agreement (Artstein, 2017). It can be seen as a differentiable generalization of the Brown clustering objective. To this end, define

$$p(z) = \mathop{\mathbf{E}}_{x \sim D}[p(z|x)] \qquad \forall z \in [m]$$
$$q(z) = \mathop{\mathbf{E}}_{y \sim D}[q(z|y)] \qquad \forall z \in [m]$$

The mutual information between the predictions of p and q on a single sample (x, y) is then

$$J^{\text{mi}}_{x,y} = \sum_{z,z'} p(z|x)q(z'|y) \log \frac{p(z|x)q(z'|y)}{p(z)q(z')}$$

and the objective (to maximize) is

$$J^{\text{mi}} = \mathop{\mathbf{E}}_{(x,y) \sim D}\left[J^{\text{mi}}_{x,y}\right]$$

Note that this becomes exactly the original Brown objective (3) if (X, Y) is defined as a random bigram and p and q are tied and constrained to be a hard clustering.

Empirical objective. In practice, we work with the value of empirical mutual information $\widehat{J}^{\text{mi}}$ estimated from the training data:

$$\hat{p}(z) = \frac{1}{N}\sum_{i=1}^{N} p(z|x_i) \qquad \forall z \in [m]$$
$$\hat{q}(z) = \frac{1}{N}\sum_{i=1}^{N} q(z|y_i) \qquad \forall z \in [m]$$
$$\widehat{J}^{\text{mi}}_i = \sum_{z,z'} p(z|x_i)q(z'|y_i) \log \frac{p(z|x_i)q(z'|y_i)}{\hat{p}(z)\hat{q}(z')}$$
$$\widehat{J}^{\text{mi}} = \frac{1}{N}\sum_{i=1}^{N} \widehat{J}^{\text{mi}}_i \qquad (4)$$

Our task is to maximize (4) by taking gradient steps at random minibatches. Note, however, that the objective cannot be written as a sum of local objectives because we take log of the estimates $\hat{q}(z)$ and $\hat{p}(z)$ computed from all samples. This makes the stochastic gradient estimator biased (i.e., it does not match the gradient of (4) in expectation) and compromises the correctness of SGD. This bias is investigated more closely in Section 4.

3.2 Variational Lower Bound

The second training objective we consider can be derived in a rather heuristic but helpful manner as follows. Since X, Y are always drawn together, if $q(z|y)$ is the target label distribution for the pair (x, y), then we can train $p(z|x)$ by minimizing the cross entropy between q and p over samples

$$H(q,p) = \mathop{\mathbf{E}}_{(x,y) \sim D}\left[-\sum_{z} q(z|y) \log p(z|x)\right]$$

which is minimized to zero at $p = q$. However, q is also untrained and needs to be trained along with p. Thus this loss alone admits trivial solutions such as setting $p(1|x) = p(1|y) = 1$ for all (x, y). This undesirable behavior can be prevented by simultaneously maximizing the entropy of q. Let Z denote a random label from q with distribution $q(z) = \mathop{\mathbf{E}}_{y \sim D}[q(z|y)]$ (thus Z is a function of q). The entropy of Z is

$$H(Z) = -\sum_{z} q(z) \log q(z)$$

Putting together, the objective (to maximize) is

$$J^{\text{var}} = H(Z) - H(q,p)$$

Variational interpretation. The reason this objective is named a variational lower bound is due to McAllester (2018) who shows the following. Consider the mutual information between Z and the raw signal X:

$$I(X,Z) = H(Z) - H(Z|X) \qquad (5)$$

Because Z is drawn from q conditioning on Y, which is co-distributed with X, we have a Markov chain $X \rightarrow Y \xrightarrow{q} Z$. Thus maximizing $I(X, Z)$ over the choice of q is a reasonable objective that enforces "predictive coding": the label predicted by q from Y must be as informative of X as possible. It can be seen as a special case of the objective underlying the information bottleneck method (Tishby et al., 2000).

So what is the problem with optimizing (5) directly? The problem is that the conditional entropy under the model

$$H(Z|X) = \mathop{\mathbf{E}}_{\substack{(x,y)\sim D \\ z\sim q(\cdot|y)}} \left[\log \frac{1}{\pi(z|x)} \right] \tag{6}$$

involves the posterior probability of z given x

$$\pi(z|x) = \frac{\sum_y D(x,y)q(z|y)}{\sum_{y,z} D(x,y)q(z|y)}$$

This *conditional* marginalization is generally intractable and cannot be approximated by sampling since the chance of seeing a particular x is small. However, we can introduce a variational distribution $p(z|x)$ to model $\pi(z|x)$. Plugging this into (6) we observe that

$$\mathop{\mathbf{E}}_{\substack{(x,y)\sim D \\ z\sim q(\cdot|y)}} \left[\log \frac{1}{p(z|x)} \right] = H(q,p)$$

Moreover,

$$\begin{aligned}
&\mathop{\mathbf{E}}_{\substack{(x,y)\sim D \\ z\sim q(\cdot|y)}} \left[\log \frac{1}{p(z|x)} \right] \\
&\quad = \mathop{\mathbf{E}}_{\substack{(x,y)\sim D \\ z\sim q(\cdot|y)}} \left[\log \frac{\pi(z|x)}{\pi(z|x)\, p(z|x)} \right] \\
&\quad = H(Z|X) + D_{\mathrm{KL}}(\pi||p)
\end{aligned}$$

Thus $H(q,p)$ is an upper bound on $H(Z|X)$ for any p with equality iff p matches the true posterior distribution π. This in turn means that $J^{\mathrm{var}} = H(Z) - H(q,p)$ is a lower bound on $I(X,Z)$, hence the name.

Empirical objective. As with the generalized Brown objective, the variational lower bound can be estimated from the training data as

$$\begin{aligned}
\widehat{H}(Z) &= -\sum_z \hat{q}(z) \log \hat{q}(z) \\
\widehat{H}(q,p) &= \frac{1}{N} \sum_{i=1}^{N} \left(-\sum_z q(z|y_i) \log p(z|x_i) \right) \\
\widehat{J}^{\mathrm{var}} &= \widehat{H}(Z) - \widehat{H}(q,p)
\end{aligned} \tag{7}$$

where $\hat{q}(z)$ is defined as in Section 3.1. Our task is again to maximize this empirical objective (7) by taking gradient steps at random minibatches. Once again, it cannot be written as a sum of local objectives because the entropy term involves log of $\hat{q}(z)$ computed from all samples. Thus it is not clear if stochastic optimization will be effective.

4 Analysis

The discussion of the two objectives in the previous section is incomplete because the stochastic gradient estimator is biased under both objectives. In this section, we formalize this issue and analyze the bias.

4.1 Setting

Let $B_1 \ldots B_K$ be a partition of the N training examples $(x_1, y_1) \ldots (x_N, y_N)$ into K (iid) minibatches. For simplicity, assume $|B_k| = M$ for all k and $N = MK$. We will adopt the same notation in Section 3 for $\hat{p}(z)$ and $\hat{q}(z)$ estimated from all N samples. Define analogous estimates based on the k-th minibatch by

$$\begin{aligned}
\hat{p}_k(z) &= \frac{1}{M} \sum_{x \in B_k} p(z|x) \qquad \forall z \in [m] \\
\hat{q}_k(z) &= \frac{1}{M} \sum_{y \in B_k} q(z|y) \qquad \forall z \in [m]
\end{aligned}$$

If l_N denotes an objective function computed from all N samples in the training data and l_k denotes the same objective computed from B_k, a condition on the correctness of SGD is that the gradient of l_k (with respect to model parameters) is consistent with the gradient of l_N on average:

$$\nabla l_N = \frac{1}{K} \sum_{k=1}^{K} \nabla l_k + \epsilon \tag{8}$$

where ϵ denotes the bias of the stochastic gradient estimator. In particular, any loss of the form $l_N = (1/K) \sum_k l_k$ that decomposes over independent minibatches (e.g., the cross-entropy loss for supervised classification) satisfies (8) with $\epsilon = 0$. The bias is nonzero for the unsupervised objectives considered in this work due to the issues discussed in Section 3.

4.2 Result

The following theorem precisely quantifies the bias for the empirical losses associated with the variational bound and the generalized Brown objectives. We only show the result with the gradient with respect to q, but the result with the gradient with respect to p is analogous and omitted for brevity.

Theorem 4.1. *Assume the setting in Section 4.1 and the gradient is taken with respect to the pa-*

rameters of q. *For* $l_N = -\widehat{J}^{\text{var}}$ *defined in* (7),

$$\epsilon = \frac{1}{K}\sum_{k=1}^{K}\sum_{z}\log\frac{\hat{q}(z)}{\hat{q}_k(z)}\nabla\hat{q}_k(z)$$

On the other hand, for $l_N = -\widehat{J}^{\text{mi}}$ *defined in* (4),

$$\epsilon = \frac{1}{N}\sum_{k=1}^{K}\sum_{z,z'}\Bigg(\epsilon_k(z,z')\nabla\hat{q}_k(z') + \log\frac{\hat{p}(z)\hat{q}(z')}{\hat{p}_k(z)\hat{q}_k(z')}\sum_{(x,y)\in B_k} p(z|x)\nabla q(z'|y)\Bigg)$$

where

$$\epsilon_k(z,z') = \frac{1}{K}\sum_{i=1}^{N}\frac{p(z|x_i)q(z'|y_i)}{\hat{q}(z')} - \sum_{(x,y)\in B_k}\frac{p(z|x)q(z'|y)}{\hat{q}_k(z')}$$

A proof can be found in the appendix. We see that both biases go to zero as $\hat{p}_k$ and $\hat{q}_k$ approach $\hat{p}$ and $\hat{q}$. However, the bias is logarithmic in the ratio $\hat{q}(z)/\hat{q}_k(z)$ for the variational lower bound but roughly linear in the difference between $\frac{1}{\hat{q}(z')}$ and $\frac{1}{\hat{q}_k(z')}$ for the generalized Brown objective. In this sense, the variational lower bound is exponentially more robust to noise in minibatch estimates than the generalized Brown objective. This is confirmed in experiments: we are able to optimize $\widehat{J}^{\text{var}}$ with minibatches as small as 80 examples but unable to optimize $\widehat{J}^{\text{mi}}$ unless minibatches are prohibitively large.

5 Experiments

We demonstrate the effectiveness of our training objectives on the task of POS induction. The goal of this task is to induce the correct POS tag for a given word in context (Merialdo, 1994). As typical in unsupervised tasks, evaluating the quality of induced labels is challenging; see Christodoulopoulos et al. (2010) for an in-depth discussion. To avoid complications, we follow a standard practice (Berg-Kirkpatrick et al., 2010; Ammar et al., 2014; Lin et al., 2015; Stratos et al., 2016) and adopt the following setting for all compared methods.

- We use many-to-one accuracy as a primary evaluation metric. That is, we map each induced label to the most frequently coinciding ground-truth POS tag in the annotated data and report the resulting accuracy. We also use the V-measure (Rosenberg and Hirschberg, 2007) when comparing with CRF autoencoders to be consistent with reported results (Ammar et al., 2014; Lin et al., 2015).
- We use the number of ground-truth POS tags as the value of m (i.e., number of labels to induce). This is a data-dependent quantity, for instance 45 in the Penn WSJ and 12 in the universal treebank. Fixing the number of tags this way obviates many evaluation issues.
- Model-specific hyperparameters are tuned on the English Penn WSJ dataset. This configuration is then fixed and used for all other datasets: 10 languages in the universal treebank[2] and 7 languages from CoNLL-X and CoNLL 2007.

5.1 Setting

We set D to be a uniform distribution over context-word pairs in the training corpus. Given N samples $(x_1, y_1) \ldots (x_N, y_N) \sim D$, we optimize the variational objective (7) or the generalized Brown objective (4) by taking gradient steps at random minibatches. This gives us conditional label distributions $p(z|x)$ and $q(z|y)$ for all contexts x, words y, and labels z. At test time, we use

$$z^* = \arg\max_z q(z|y)$$

as the induced label of word y. We experimented with different inference methods such as taking $\arg\max_z p(z|x)q(z|y)$ but did not find it helpful.

5.2 Definition of (X, Y)

Let V denote the vocabulary. We assume an integer $H \geq 1$ that specifies the width of local context. Given random word $y \in V$, we set $x \in V^{2H}$ to be an ordered list of H left and H right words of y. For example, with $H = 2$, a typical context-target pair $(x, y) \sim D$ may look like

$$x = (\text{“had”}, \text{“these”}, \text{“in”}, \text{“my”})$$
$$y = \text{“keys”}$$

We find this simple fixed-window definition of observed variables to be the best inductive bias for POS induction. The correct label can be inferred from either x or y in many cases: in the above example, we can infer that the correct POS tag is plural noun by looking at the target or the context.

[2]https://github.com/ryanmcd/uni-dep-tb

5.3 Architecture

We use the following simple architecture to parameterize the label distribution $p(\cdot|x)$ conditioned on context $x \in V^H$ and the label distribution $q(\cdot|y)$ conditioned on word $y \in V$.

Context architecture. The parameters of p are word embeddings $e_w \in \mathbb{R}^d$ for all $w \in V$ and matrices $W_j \in \mathbb{R}^{m \times d}$ for all $j = 1 \dots 2H$. Given $2H$ ordered contextual words $x = (w_j)_{j=1}^{2H}$, we define

$$p(\cdot|x) = \text{softmax}\left(\sum_{j=1}^{2H} W_j e_{w_j}\right)$$

Word architecture. The parameters of q are the same word embeddings $e_w \in \mathbb{R}^d$ shared with p, character embeddings $e_c \in \mathbb{R}^{d/2}$ for all distinct characters c, two single-layer LSTMs with input/output dimension $d/2$, and matrices $W_c, W_w \in \mathbb{R}^{m \times d}$. Given the word y with character sequence $c_1 \dots c_T$, we define

$$\begin{aligned}(f_1 \dots f_T) &= \text{LSTM}_f(e_{c_1} \dots e_{c_T}) \\ (b_1 \dots b_T) &= \text{LSTM}_b(e_{c_T} \dots e_{c_1}) \\ q(\cdot|y) &= \text{softmax}\left(W_c \begin{bmatrix} f_T \\ b_T \end{bmatrix} + W_w e_y\right)\end{aligned}$$

The overall architecture is illustrated in Figure 1. Our hyperparameters are the embedding dimension $d = 200$, the context width $H = 2$, the learning rate of the Adam optimizer $r = 0.001$, and the minibatch size $B = 80$.[3] Their values are tuned on the 45-tag Penn WSJ dataset to maximize accuracy.

5.4 Baselines

We focus on comparing with the following models which are some of the strongest baselines in the literature we are aware of. Berg-Kirkpatrick et al. (2010) extend a standard hidden Markov Model (HMM) to incorporate linguistic features. Stratos et al. (2016) develop a factorization-based algorithm for learning a constrained HMM. Ammar et al. (2014) propose a CRF autoencoder that reconstructs words from a structured label sequence. Lin et al. (2015) extend Ammar et al. (2014) by switching a categorical reconstruction distribution with a Gaussian distribution. In addition to these

[3]An implementation is available at: https://github.com/karlstratos/mmi-tagger.

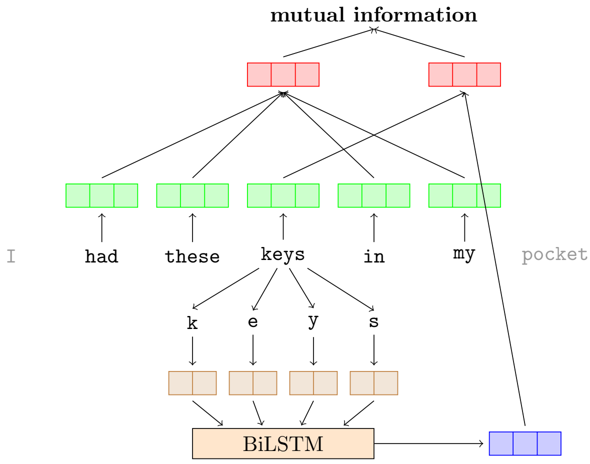

Figure 1: Architecture illustrated on the example text "had these keys in my" with target Y = "keys".

baselines, we also report results with Brown clustering (Brown et al., 1992), the Baum-Welch algorithm (Baum and Petrie, 1966), and k-means clustering of 300-dimensional GloVe vectors (Pennington et al., 2014).

5.5 Results

The 45-tag Penn WSJ dataset. The 45-tag Penn WSJ dataset is a corpus of around one million words each tagged with one of $m = 45$ tags. It is used to optimize hyperparameter values for all compared methods. Table 1 shows the average accuracy over 10 random restarts with the best hyperparameter configurations; standard deviation is given in parentheses (except for deterministic methods Stratos et al. (2016) and Brown clustering).

Our model trained with the variational objective (7) outperforms all baselines.[4] We also observe that our model trained with the generalized Brown objective (4) does not work. We have found that unless the minibatch size is as large as 10,000 the gradient steps do not effectively increase the true data-wide mutual information (4). This supports our bias analysis in Section 4. While it may be possible to develop techniques to resolve the difficulty, for instance keeping a moving average of estimates to stabilize estimation, we leave this as future work and focus on the variational objective in the remainder of the paper.

Table 2 shows ablation experiments on our best

[4] We remark that Tran et al. (2016) report a single number 79.1 with a neuralized HMM. We also note that the concurrent work by He et al. (2018) obtains 80.8 by using word embeddings carefully pretrained on one billion words.

Method	Accuracy
Variational $\widehat{J}^{\text{var}}$ (7)	**78.1** (±0.8)
Generalized Brown $\widehat{J}^{\text{mi}}$ (4)	48.8 (±0.9)
Berg-Kirkpatrick et al. (2010)	74.9 (±1.5)
Stratos et al. (2016)	67.7
Brown et al. (1992)	65.6
Baum-Welch	62.6 (±1.1)
k-MEANS	32.6 (±0.7)

Table 1: Many-to-one accuracy on the 45-tag Penn WSJ with the best hyperparameter configurations. The average accuracy over 10 random restarts is reported and the standard deviation is given in parentheses (except for deterministic methods).

Configuration	Accuracy
Best	80.1
$H = 3$	75.9
$H = 1$	75.9
Sentence-level batching	72.4
GloVe initialization	67.6
No character encoding	65.6

Table 2: Ablation of the best model on Penn WSJ.

model (accuracy 80.1) to better understand the sources of its strong performance. Context size $H = 2$ is a sizable improvement over $H = 3$ or $H = 1$. Random sampling is significantly more effective than sentence-level batching (i.e., each minibatch is the set of context-word pairs within a single sentence as done in McAllester (2018)). GloVe initialization of word embeddings e_w is harmful. As expected for POS tagging, morphological modeling with LSTMs gives the largest improvement.

While it may be surprising that GloVe initialization is harmful, it is well known that pretrained word embeddings do not necessarily capture syntactic relationships (as evident in the poor performance of k-means clustering). Consider the top ten nearest neighbors of the word "made" under GloVe embeddings (840B.300d, within PTB vocab) shown in Table 3. The neighbors are clearly not in the same syntactic category. The embeddings can be made more syntactic by controlling the context window. But we found it much more effective (and simpler) to start from randomly initialized embeddings and let the objective induce appropriate representations.

Cosine Similarity	Nearest Neighbor
0.7426	making
0.7113	make
0.6851	that
0.6613	they
0.6584	been
0.6574	would
0.6533	brought
0.6521	had
0.6514	came
0.6494	but
0.6486	even

Table 3: Nearest neighbors of "made" under GloVe embeddings (840B.300d, within PTB vocab).

The 12-tag universal treebank. The universal treebank v2.0 is a corpus in ten languages tagged with $m = 12$ universal POS tags (McDonald et al., 2013). We use this corpus to be compatible with existing results. Table 4 shows results on the dataset, using the same setting in the experiments on the Penn WSJ dataset. Our model significantly outperforms the previous state-of-the-art, achieving an absolute gain of 4.7 over Stratos et al. (2016) in average accuracy.

Comparison with CRF autoencoders. Table 5 shows a direct comparison with CRF autoencoders (Ammar et al., 2014; Lin et al., 2015) in many-to-one accuracy and the V-measure. We compare against their reported numbers by running our model once on the same datasets using the same setting in the experiments on the Penn WSJ dataset. The data consists of the training portion of CoNLL-X and CoNLL 2007 labeled with 12 universal tags. Our model is competitive with all baselines.

6 Related Work

Information theory, in particular mutual information, has played a prominent role in NLP (Church and Hanks, 1990; Brown et al., 1992). It has intimate connections to the representation learning capabilities of neural networks (Tishby and Zaslavsky, 2015) and underlies many celebrated modern approaches to unsupervised learning such as generative adversarial networks (GANs) (Goodfellow et al., 2014).

There is a recent burst of effort in learning continuous representations by optimizing various lower bounds on mutual information (Belghazi et al., 2018; Oord et al., 2018; Hjelm et al., 2018). These representations are typically eval-

Method	de	en	es	fr	id	it	ja	ko	pt-br	sv	Mean
Variational $\widehat{\mathcal{J}}^{\text{var}}$ (7)	(±1.5) **75.4**	(±1.7) **73.1**	(±1.0) 73.1	(±2.9) 70.4	(±1.5) **73.6**	(±3.3) **67.4**	(±0.4) 77.9	(±1.2) **65.6**	(±2.3) **70.7**	(±1.5) **67.1**	**71.4**
Stratos et al.	63.4	71.4	**74.3**	**71.9**	67.3	60.2	69.4	61.8	65.8	61.0	66.7
Berg-Kirkpatrick et al.	(±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	63.2
Brown et al.	60.0	62.9	67.4	66.4	59.3	66.1	60.3	47.5	67.4	61.9	61.9
Baum-Welch	(±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	54.0

Table 4: Many-to-one accuracy on the 12-tag universal treebank dataset. We use the same setting in Table 1. All models use a fixed hyperparameter configuration optimized on the 45-tag Penn WSJ.

Metric	Method	Arabic	Basque	Danish	Greek	Hungarian	Italian	Turkish	Mean
M2O	Variational $\widehat{\mathcal{J}}^{\text{var}}$ (7)	**74.3**	**70.4**	**71.7**	**66.1**	**61.2**	**67.4**	**64.2**	**67.9**
	Ammar et al.	69.1	68.1	60.9	63.5	57.1	60.4	60.4	62.8
	Berg-Kirkpatrick et al.	66.8	66.2	60.0	60.2	56.8	64.1	62.0	62.3
	Baum-Welch	49.7	44.9	42.4	39.2	45.2	39.3	52.7	44.7
VM	Variational $\widehat{\mathcal{J}}^{\text{var}}$ (7)	**56.9**	43.6	**56.0**	**56.3**	47.9	**53.3**	38.5	**50.4**
	Lin et al.	50.5	**51.7**	51.3	50.0	**55.9**	46.3	**43.1**	49.8
	Ammar et al.	49.1	41.1	46.1	49.1	41.1	43.1	35.0	43.5
	Berg-Kirkpatrick et al.	33.8	33.4	41.1	40.9	39.0	46.6	31.6	38.8
	Baum-Welch	15.3	8.2	11.1	9.6	10.1	9.9	11.6	10.8

Table 5: Comparison with the reported results with CRF autoencoders in many-to-one accuracy (M2O) and the V-measure (VM).

uated on extrinsic tasks as features. In contrast, we learn discrete representations by optimizing a novel generalization of the Brown clustering objective (Brown et al., 1992) and a variational lower bound on mutual information proposed by McAllester (2018). We focus on intrinsic evaluation of these representations on POS induction. Extrinsic evaluation of these representations in downstream tasks is an important future direction.

The issue of biased stochastic gradient estimators is a common challenge in unsupervised learning (e.g., see Wang et al., 2015). This arises mainly because the objective involves a nonlinear transformation of all samples in a training dataset, for instance the whitening constraints in deep canonical correlation analysis (CCA) (Andrew et al., 2013). In this work, the problem arises because of entropy. This issue is not considered in the original work of McAllester (2018) and the error analysis we present in Section 4 is novel. Our finding is that the feasibility of stochastic optimization greatly depends on the size of the bias in gradient estimates, as we are able to effectively optimize the variational objective while not the generalized Brown objective.

Our POS induction system has some practical advantages over previous approaches. Many rely on computationally expensive structured inference or pre-optimized features (or both). For instance, Tran et al. (2016) need to calculate forward/backward messages and is limited to truncated sequences by memory constraints. Berg-Kirkpatrick et al. (2010) rely on extensively hand-engineered linguistic features. Ammar et al. (2014), Lin et al. (2015), and He et al. (2018) rely on carefully pretrained lexical representations like Brown clusters and word embeddings. In contrast, the model presented in this work requires no expensive structured computation or feature engineering and uses word/character embeddings trained from scratch. It is easy to implement using a standard neural network library and outperforms these previous works in many cases.

Acknowledgments

The author thanks David McAllester for many insightful discussions, and Sam Wiseman for helpful comments. The Titan Xp used for this research was donated by the NVIDIA Corporation.

References

Waleed Ammar, Chris Dyer, and Noah A Smith. 2014. Conditional random field autoencoders for unsupervised structured prediction. In *Advances in Neural Information Processing Systems*, pages 3311–3319.

Galen Andrew, Raman Arora, Jeff Bilmes, and Karen Livescu. 2013. Deep canonical correlation analysis.

In *Proceedings of the 30th International Conference on Machine Learning*, pages 1247–1255.

Ron Artstein. 2017. Inter-annotator agreement. In *Handbook of linguistic annotation*, pages 297–313. Springer.

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.

Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeshwar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and Devon Hjelm. 2018. Mutual information neural estimation. In *Proceedings of the 35th International Conference on Machine Learning*, pages 531–540.

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.

Kenneth Ward Church and Patrick Hanks. 1990. Word association norms, mutual information, and lexicography. *Computational linguistics*, 16(1):22–29.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680.

Junxian He, Graham Neubig, and Taylor Berg-Kirkpatrick. 2018. Unsupervised learning of syntactic structure with invertible neural projections. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1292–1302.

R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Adam Trischler, and Yoshua Bengio. 2018. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*.

Justin B Kinney and Gurinder S Atwal. 2014. Equitability, mutual information, and the maximal information coefficient. *Proceedings of the National Academy of Sciences*, page 201309933.

Terry Koo, Xavier Carreras, and Michael Collins. 2008. Simple semi-supervised dependency parsing. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Chu-Cheng Lin, Waleed Ammar, Chris Dyer, and Lori Levin. 2015. Unsupervised pos induction with word embeddings. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1311–1316, Denver, Colorado. Association for Computational Linguistics.

David McAllester. 2018. Information theoretic co-training. *arXiv preprint arXiv:1802.07572*.

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.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

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. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the Empiricial Methods in Natural Language Processing*, volume 12.

Andrew Rosenberg and Julia Hirschberg. 2007. V-measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*.

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

Naftali Tishby, Fernando C Pereira, and William Bialek. 2000. The information bottleneck method. *arXiv preprint physics/0004057*.

Naftali Tishby and Noga Zaslavsky. 2015. Deep learning and the information bottleneck principle. In *Information Theory Workshop (ITW), 2015 IEEE*, pages 1–5. IEEE.

Ke M. Tran, Yonatan Bisk, Ashish Vaswani, Daniel Marcu, and Kevin Knight. 2016. Unsupervised neural hidden markov models. In *Proceedings of the Workshop on Structured Prediction for NLP*, pages 63–71. Association for Computational Linguistics.

Weiran Wang, Raman Arora, Karen Livescu, and Nathan Srebro. 2015. Stochastic optimization for deep cca via nonlinear orthogonal iterations. In *Communication, Control, and Computing (Allerton), 2015 53rd Annual Allerton Conference on*, pages 688–695. IEEE.

A Proof of Theorem 4.1

We first analyze the variational loss

$$\widehat{H}(q,p) - \widehat{H}(Z)$$

Note that the cross entropy term decomposes over samples and causes no bias. Thus we focus on the negative entropy term

$$-\widehat{H}(Z) = \sum_z \hat{q}(z) \log \hat{q}(z)$$

whose gradient with respect to q is

$$\begin{aligned}&\sum_z (1 + \log \hat{q}(z)) \nabla \hat{q}(z) \\ &= \frac{1}{K} \sum_{k=1}^{K} \sum_z (1 + \log \hat{q}(z)) \nabla \hat{q}_k(z) \qquad (9)\end{aligned}$$

where we expand $\nabla \hat{q}(z)$ by the identity

$$\nabla \hat{q}(z) = \frac{1}{N} \sum_{i=1}^{N} \nabla q(z|y_i) = \frac{1}{K} \sum_{k=1}^{K} \nabla \hat{q}_k(z) \tag{10}$$

In contrast, the gradient of the negative entropy term averaged over minibatches is

$$\frac{1}{K} \sum_{k=1}^{K} \sum_z (1 + \log \hat{q}_k(z)) \nabla \hat{q}_k(z) \tag{11}$$

Hence the difference between (9) and (11) is

$$\frac{1}{K} \sum_{k=1}^{K} \sum_z \log \frac{\hat{q}(z)}{\hat{q}_k(z)} \nabla \hat{q}_k(z)$$

This shows the first result. Now we analyze the generalized Brown loss

$$\frac{1}{N} \sum_{i=1}^{N} \sum_{z,z'} p(z|x_i) q(z'|y_i) \log \frac{\hat{p}(z)\hat{q}(z')}{p(z|x_i)q(z'|y_i)}$$

When we expand the log fraction, we see that the denominator decomposes over samples and causes no bias. Thus we focus on the numerator term

$$\frac{1}{N} \sum_{z,z'} \log \left(\hat{p}(z)\hat{q}(z')\right) \sum_{i=1}^{N} p(z|x_i) q(z'|y_i)$$

By the product rule, its gradient with respect to q is a sum of two terms. The first term is (using (10) again)

$$\frac{1}{N} \sum_{k=1}^{K} \sum_{z,z'} \left(\frac{1}{K} \sum_{i=1}^{N} \frac{p(z|x_i)q(z'|y_i)}{\hat{q}(z')} \right) \nabla \hat{q}_k(z') \tag{12}$$

The second term is (as a sum over batches)

$$\frac{1}{N} \sum_{k=1}^{K} \sum_{z,z'} \log \left(\hat{p}(z)\hat{q}(z')\right) \sum_{(x,y)\in B_k} p(z|x) \nabla q(z'|y) \tag{13}$$

In contrast, the numerator term estimated as an average over minibatches is

$$\frac{1}{N} \sum_{k=1}^{K} \sum_{z,z'} \log \left(\hat{p}_k(z)\hat{q}_k(z')\right) \sum_{(x,y)\in B_k} p(z|x) q(z'|y)$$

and the two terms of its gradient with respect to q (corresponding to (12) and (13)) are

$$\frac{1}{N} \sum_{k=1}^{K} \sum_{z,z'} \left(\sum_{(x,y)\in B_k} \frac{p(z|x)q(z'|y)}{\hat{q}_k(z')} \right) \nabla \hat{q}_k(z') \tag{14}$$

$$\frac{1}{N} \sum_{k=1}^{K} \sum_{z,z'} \log \left(\hat{p}_k(z)\hat{q}_k(z')\right) \sum_{(x,y)\in B_k} p(z|x_i) \nabla q(z'|y) \tag{15}$$

Thus the difference between (12) and (14) is

$$\frac{1}{N} \sum_{k=1}^{K} \sum_{z,z'} \epsilon_k(z,z') \nabla \hat{q}_k(z')$$

where

$$\begin{aligned}\epsilon_k(z,z') = &\frac{1}{K} \sum_{i=1}^{N} \frac{p(z|x_i)q(z'|y_i)}{\hat{q}(z')} \\ &- \sum_{(x,y)\in B_k} \frac{p(z|x)q(z'|y)}{\hat{q}_k(z')}\end{aligned}$$

The difference between (13) and (15) is

$$\frac{1}{N} \sum_{k=1}^{K} \sum_{z,z'} \log \frac{\hat{p}(z)\hat{q}(z')}{\hat{p}_k(z)\hat{q}_k(z')} \sum_{(x,y)\in B_k} p(z|x) \nabla q(z'|y)$$

Adding these differences gives the second result.