Weakly Supervised Part-of-Speech (POS) Tagging without Disambiguation

Deyu Zhou, Zhikai Zhang, Min-Ling Zhang, School of Computer Science and Engineering, Southeast University, China
Yulan He, School of Engineering and Applied Science, Aston University, UK

Weakly supervised part-of-speech (POS) tagging aims to predict the POS tag for a given word in context by making use of partially annotated data instead of fully tagged corpora. As POS tagging is crucial for downstream natural language processing (NLP) tasks such as named entity recognition and information extraction, weakly supervised POS tagging is specifically attractive in languages where tagged corpora are mostly unavailable. In this paper, we propose a novel framework for weakly supervised POS tagging where no annotate corpora are available and the only supervision information comes from a dictionary of words where each of them is associated with a list of possible POS tags. Our approach is built upon error-correcting output codes (ECOC) is which each POS tag is assigned with a unique L-bit binary vector. For a total of O POS tags, we therefore have a coding matrix M of size $O \times L$ with value $\{1, -1\}$. Each column of the coding matrix M specifies a dichotomy over the tag space to learn a binary classifier. For each binary classifier, its training data is generated in the following way: a word will be considered as a positive training example only if the whole set of its possible tags falls into the positive dichotomy specified by the column coding; and similarly for negative training examples. Given a word, its POS tag is predicted by concatenating the predictive outputs of the L binary classifiers and choosing the one with the closest distance according to some measure. By incorporating the ECOC strategy, the set of all possible tags for each word is treated as an entirety without the need of performing disambiguation. Moreover, instead of manual feature engineering employed in most previous POS tagging approaches, features for training and testing in the proposed framework are automatically generated using neural language modeling. The proposed framework has been evaluated on three corpora for English, Italian and Malagasy POS tagging, achieving accuracies of 93.21%, 90.9% and 84.5% respectively, which shows a significant improvement compared to the state-of-the-art approaches.

1. INTRODUCTION

Part-of-speech (POS) tagging is to assign a POS tag to a word in text based on its context. It is crucial for downstream natural language processing (NLP) tasks such as named entity recognition [Finkel et al. 2005], syntactic parsing [Cer et al. 2010] and information extraction [Zhou et al. 2015]. Methods for POS tagging in general fall into two categories: rule based and machine learning based. Rule based approaches rely on manually designed rules while machine learning approaches require a large amount of annotated data for training.

In low-resource languages such as Malagasy, annotated data are mostly unavailable. It is thus attractive to explore weakly-supervised POS tagging approaches where the supervision information comes from other sources rather than the annotated data. As the ground-truth POS tag of a word in a sentence is not directly accessible, weakly-supervised approaches are more difficult to train compared to supervised approaches. One common way to address the problem of lack of annotated data is to make use of a dictionary of words with each one associated with a set of possible POS tags. The actual POS tag of a word in a sentence is considered as a latent variable which is identified via iterative refinement procedure. Thus, a typical setup for weakly-supervised

DOI: http://dx.doi.org/10.1145/0000000.0000000

Table I. An example of input and output of weakly supervised POS tagging. (PRP denotes personal pronoun, DT for determiner, JJ for adjective, VB for verb base form, CD for cardinal number and so on)

Dictionary (Each word is associated with a list of possible POS tags)	
you PRP; these DT; events NNS; took VBD; 35 CD; years NNS; ago IN RB; to IN JJ TO; place NN VB VBP; recognize VB VBP; that DT IN NN RB VBP WDT; have JJ VBD VBN VBP;...	
Input	**Output**
You have to recognize that these events took place 35 years .	You/PRP have/VBP to/TO recognize/VB that/IN these/DT events/NNS took/VBD place/NN 35/CD years/NNS ago/IN ./.

POS tagging is that given a dictionary of words with their possible POS tags, we aim to generate a correct POS tag sequence for any unannotated input sentence. This is illustrated in Table I.

Previous weakly-supervised POS tagging approaches are largely based on expectation maximization (EM) parameters estimation using hidden Markov models (HMMs) or conditional random fields (CRFs). For example, Merialdo [1994a] used maximum likelihood estimation (MLE) to train a trigram HMM. Banko and Moore [2004] modified the basic HMM structure to incorporate the context on both sides of the word to be tagged. Smith and Eisner [2005] proposed to train CRFs using contrastive estimation for POS tagging. It can be observed that most of the aforementioned approaches essentially perform disambiguation on a set of possible candidate POS tags for a word in a sentence. Although disambiguation presents as an intuitive and reasonable strategy to training weakly-supervised POS taggers, its effectiveness is largely affected by the possible errors introduced in the previous training iterations. That is, false positive tag(s) identified in early iterations will be propagated to the next iteration which makes it difficult for the model to identify the correct POS tag.

In this paper, we propose a novel framework for weakly supervised POS tagging without the need of disambiguating among a set of possible POS tags, built upon error-correcting output codes (ECOC) [Dietterich and Bakiri 1995], one of the multi-class learning techniques. A unique L-bit vector is assigned to each POS tag. For a total of O POS tags, a coding matrix M of size $O \times L$ can be constructed where each cell of M has a value of $\{1, -1\}$. Each column of M specifies a dichotomy over the tag space to learn a binary classifier. For example, given a set of POS tags {VB, DT, VBP, NN}, the column of M [-1,+1,-1,+1] separates the tag space into negative dichotomy {VB, VBP} and positive dichotomy{DT, NN}. The key adaptation lies in how the binary classifiers corresponding to the ECOC coding matrix M are built. For each column of the binary coding matrix, a binary classifier is built based on training examples derived from the dictionary of the words with their possible POS tags. Specifically, the word will be regarded as a positive or negative training example only if all its possible tags fall into the positive or negative dichotomy specified by the column coding. In this way, the set of possible tags is treated as an entirety without resorting to any disambiguation procedure. Moreover, the choice of features is a critical success factor for POS tagging. Most of the state-of-the-art POS tagging systems extract features based on the lexical context of the words to be tagged and their letter structures (e.g., presence of suffixes, capitalization and hyphenation). Obviously, such feature design needs domain knowledge and expertise. In this paper, features employed for weakly supervised POS tagging are generated based on neural language modelling without manual processing. The proposed approach has been evaluated on three corpora for English, Italian and Malagasy POS tagging, and shows a significant improvement in accuracy compared to the state-of-the-art approaches.

The main contributions of the paper are summarized below:

— We proposed a novel framework based on constrained ECOC for weakly supervised POS tagging. In such way, the set of a word's possible tags is treated as an entirety without resorting to any disambiguation procedure. It thus avoids the problem of iterative training based on disambiguation, which is commonly used for existing approaches to weakly supervised POS tagging.

- We developed a POS tagging system without human intervention. Features employed for POS tagging are generated automatically based on neural language modelling.
- We evaluated the proposed framework on three corpora for English, Italian and Malagasy POS tagging, and observed a significant improvement in accuracy compared to the state-of-the-art approaches.

2. RELATED WORK

Supervised POS tagging has achieved very good results with per-token accuracies over 97% on the English Penn Treebank. However, there are more than 50 low-density languages where both tagged corpora and language speakers are mostly unavailable [Christodoulopoulos et al. 2010]. Some of them are even dead. Therefore, POS tagging without using any fully annotated corpora has attracted increasing interests. Generally, based on whether to use supervised information and where the supervised information comes from, there are three directions for handling the task: POS induction, where no prior knowledge is used; POS disambiguation, where a dictionary of words and their possible tags is assumed to be available; and prototype-driven approaches where a small set of prototypes for each POS tag is provided instead of a dictionary.

For fully unsupervised POS tagging or POS induction, many approaches casted the identification of POS tags as a knowledge-free clustering problem. Brown *et al.* [1992] proposed a n-gram model based on classes of words through optimizing the probability of the corpus $p(w_1|c_1)\prod_2^n p(w_i|c_i)p(c_i|c_{i-1})$ using some greedy hierarchical clustering. Following this way, Clark [2003] incorporated morphological information into clustering so that morphologically similar words are clustered together. Based on a standard trigram HMM, Goldwarter and Griffiths [2007] proposed a fully Bayesian approach which allowed the use of priors. A collapsed Gibbs sampler was used to inferring the hidden POS tags. Johnson [2007] also experimented with variational Bayesian EM apart from Gibbs sampling and his results showed that variational Bayesian converges faster than Gibbs sampling for POS tagging. Using the structure of a standard HMM, Berg-Kirkpatrick *et al.* [2010] turned each component multinomial of the HMM into a miniature logistic regression. By doing so, features can be easily added to standard generative models for unsupervised learning, without requiring complex new training methods. Different from the previous approaches, a graph clustering approach based on contextual similarity was proposed in [Biemann 2006] so that the number of POS tags (clusters) could be induced automatically. Based on the theory of prototypes, Abend *et al.* [2010] first clustered the most frequent words based on some morphological representations. They then defined landmark clusters which served as the cores of the induced POS categories and finally map the rest of the words to these categories. Kairit *et al.* [2014] presented an approach for inducing POS classes by combining morphological and distributional information in non-parametric Bayesian generative model based on distance-dependent Chinese restaurant process. As pointed out in [Christodoulopoulos et al. 2010], due to a lack of standard and informative evaluation techniques, it is difficult to compare the effectiveness of different clustering methods.

For weakly-supervised POS tagging, many researchers focused on POS disambiguation using tag dictionaries. Brill [1992] described a rule-based POS tagger, which captured the learned knowledge into a set of simple deterministic rules instead of a large table of statistics. He later proposed an unsupervised learning algorithm for automatically training a rule-based POS tagger [Brill 1995]. Considering POS tags as latent variables, there have been quite a few approaches relying on EM parameters estimation using HMMs or CRFs. For example, given a sentence $W = [w_1, w_2, ..., w_n\}$ and a sequence of tags $T = \{t_1, t_2, ..., t_n\}$ of the same length, a trigram model defined as $p(W,T) = \prod_{i=1}^{n} p(w_i|t_i)p(t_i|t_{i-2}, t_{i-1})$ was proposed in [Merialdo 1994b]. Following this way, some improvements were achieved by modifying the statistical models or employing better parameter estimation techniques. For example, Banko and Moore [2004] modified the basic HMM structure to incorporate the context on both sides of the word to be tagged. Smith and Eisner [2005] used contrastive estimation on CRFs for POS tagging. Toutanova *et al.* [2007] proposed a Bayesian model that extended latent Dirichlet allocation (LDA) and incorporated the intuition that words' distributions over tags are sparse. Naseem *et al.* [2009] proposed multilingual learning by combining cues from multiple languages in two ways: directly merging tag structures for a pair of languages into a single sequence, and incorporating multilingual context using latent variables. Markov Chain Monte Carlo sampling techniques were used for estimating the parameters of hierarchical Bayesian models. Ravi and Knight [2009] proposed using Integer Programming (IP) to search the smallest bi-gram POS tag set and used this set to constrain the training of EM. Their approach achieved an accuracy of 91.6% on the 24k English Penn Treebank test set, but could not handle larger datasets. For solving the deficiency of IP, Ravi *et al.* [2010] proposed a two-stage greedy minimization approach that run much faster while maintaining the performance of tagging. Yatbaz and Yuret [2010] chose unambiguous substitutes for each occurrence of an ambiguous word based on its context. Their approach achieved an accuracy of 92.25% using standard HMM model on the 24k test set. To further improve the performance, several heuristics were used in [Garrette and Baldridge 2012], which achieved an accuracy of 88.52% by using incomplete dictionary. Ravi *et al.* [2014] proposed a distributed minimum label cover which could parallelize the algorithm while preserving approximation guarantees. The approach achieved an accuracy of 91.4% on the 24k test set and 88.15% using incomplete dictionary.

Instead of using tag dictionaries, a few canonical examples of each POS tag could be used in prototype-driven learning [Haghighi and Klein 2006]. The provided prototype information could be propagated across a corpus using distributional similarity features in a log linear generative model. In a similar vein, a closed-class lexicon specifying possible tags was used to learn a disambiguation model for disambiguating the occurrences of words in context [Zhao and Marcus 2009].

Our work is similar to approaches to weakly-supervised learning using tag dictionaries since we also assume the availability of such a dictionary consisting of words with each associated with a list of possible POS tags. However, most previous approaches try to disambiguate the word's possible tags by identifying the ground-truth tag iteratively. This disambiguation is prone to be misled by the false positive tags within possible tags set. In this paper, we propose a novel approach for weakly supervised POS tagging. The set of possible tags is treated as an entirety without the need of disambiguation. From the perspective of machine learning, our approach falls into the partial label learning framework [Zhang 2014] in which each training instance is associated with a set of candidate labels, among which only one is correct. However, our problem setting here is different. The only supervision information we have is a POS tag dictionary which lists all possible POS tags for each word. The annotations of training instances need to be generated based on the POS tag dictionary. Moreover, the tag dictionary is equally applied to both the training and testing instances. Such constrains are applied in the test data using constrained ECOC.

Table II. Notations.

Symbol	*Description*
O	A list of distinct POS tags
D	A dictionary of words and their corresponding possible POS tags
U	An unannotated corpus consisting of sentences
G	A list of words and their corresponding word embeddings
L	ECOC codeword length
$\mathfrak{B}$	Binary learner used for ECOC training
thr	The threshold controlling the size of binary training set
T	The training data set

3. THE PROPOSED APPROACH

Assuming a full list of POS tags O and a dictionary of words and their corresponding possible POS tags D, we aim to predict the POS tag for a given word w in a sentence. Firstly, each word w in an unlabeled corpus U is converted into a feature vector based on neural language modeling. The word's feature vector together with its neighboring words' feature vectors form the word's context feature set. For each word w, its context feature set $\phi(w)$ and its corresponding possible POS tags A_w, which are retrieved from the dictionary D, form one training example in the training dataset T. After that, POS tagging is conducted following the encoding-decoding procedure. Table II lists notations used in this paper. The process of the proposed approach is illustrated in Figure 1 which consists of two main components, one is *Training Data Generation* and the other is *Training and Testing based on Constrained ECOC*. The details of each component are described as follows.

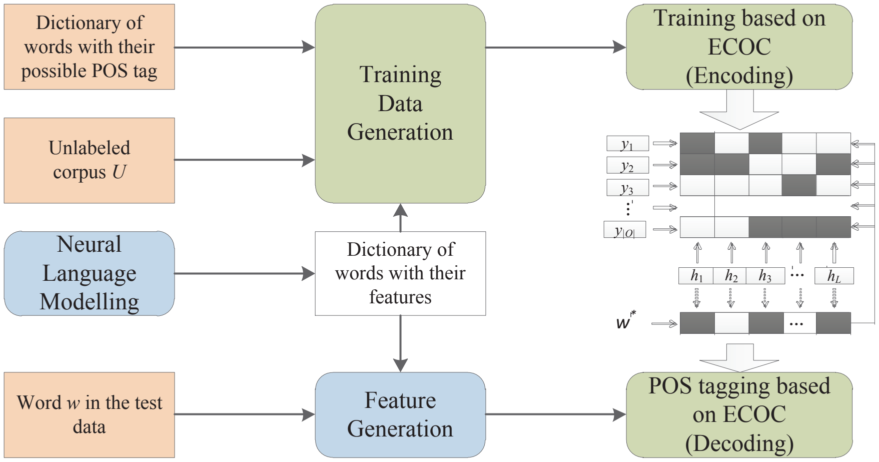

Dictionary of words with their possible POS tag
Unlabeled corpus U
Neural Language Modelling
Word w in the test data
Training Data Generation
Dictionary of words with their features
Feature Generation
Training based on ECOC (Encoding)
y_1
y_2
y_3
$y_{|O|}$
h_1
h_2
h_3
...
h_L
w^*
...
POS tagging based on ECOC (Decoding)

Fig. 1. The proposed approach for weakly supervised POS tagging.

3.1. Error Correcting Output Codes (ECOC)

As the proposed approach for POS tagging is based on ECOC, we give a brief introduction to ECOC. In machine learning, multi-class classification problem is the problem of classifying instances into one of more than two classes. ECOC is a widely applied strategy for multi-class classification that enhances the generalization ability of binary classifiers.

Assuming there are $O (O > 2)$ labels $y_1, y_2, ..., y_O$, one assigns a unique L-bit vector to each label. It can be viewed as a unique coding for the label. In general, $L > O$. The set of bit-vectors is referred to as coding matrix and denoted as M with value $\{1, -1\}$. Then, the ECOC method can be separated into two steps: encoding and decoding. In the encoding step, a binary classifier is learned for each column of the coding matrix M which specifies a dichotomy over the label space. Therefore, each column corresponds to a binary classifier which separates the set of classes into two meta-classes. The instance x which belongs to the class i is considered as a positive instance for the j^{th} classifier if and only if $M_{i,j} = 1$ and is a negative instance if and only if $M_{i,j} = -1$.

In the decoding step, the codeword of an unlabeled test instance is generated by concatenating the predictive outputs of the L binary classifiers. The instance is assigned to the class with the closest codeword according to some distance measure.

3.2. Training Data Generation

In this section, we describe how to generate training data based on word embeddings, which is shown in Algorithm 1. Word embedding or word representation of each word is a real-value vector usually with a dimension of between 50 and 300. Word embeddings aim to capture the syntactic or semantic regularities among words such that words which are semantically similar to each other are placed in nearby locations in the embedding space. This characteristic is precisely what we want. We use neural language modeling [Collobert et al. 2011] to learn word representations by discriminating the legitimate phrase from incorrect phrases.

Given a word sequence $p = (w_1, w_2, ..., w_d)$ with window size d, the goal of the model is to discriminate the sequence of words p (the correct phrase) from a random sequence of words p^r. Thus, the objective of the model is to minimize the ranking loss with respect to parameters θ:

$$\sum_{p \in \mathfrak{p}} \sum_{r \in \mathfrak{R}} \max(0, 1 - f_\theta(p) + f_\theta(p^r)), \tag{1}$$

where $\mathfrak{p}$ is the set of all possible text sequences with d words coming from the corpus U, $\mathfrak{R}$ is the dictionary of words, p^r denotes the sequence of words obtained by replacing the central word of p by the word r and $f_\theta(p)$ is the ranking score of p. Therefore, the dataset for learning the language model can be constructed by considering all the word sequences in the corpus. Positive examples are the word sequences from the corpus, while negative examples are the same word sequence with the central word replaced by a random one.

Algorithm 1 Training Data Generation.

```
Input:  $O$, $D$, $U$, $G$
Output: $T$
Initialize the training data set $T = \varnothing$;
for each word $w$ in each sentence of $U$ do
    Retrieve from $G$ the word embeddings of $w$, and its previous and next word;
    Concatenate the retrieved vectors to form the feature of $w$, $\phi(w)$;
    Retrieve from $D$ all possible POS tags $A_w$ for word $w$;
    Insert the pair $(\phi(w), A_w)$ into the training set $T$;
end for
$T = \{(\phi(w_i), A_i) | 1 \leq i \leq |U|\} (w_i \in U, A_i \subseteq O)$;
```

To illustrate how the training data is generated, we present an example as shown in Figure 2. Given a sentence “He is also trying to get more stations.” from unannotated corpus U, we want to generate a $(\phi(w), A_w)$ pair for the word "get". The feature set $\phi(w)$ of word "get" is generated by concatenating word embeddings of "to", "get" and "more" retrieved from the dictionary of word embedding, the output of neural language modeling. The candidate POS tags of the word "get" are VB and VBP retrieved from the dictionary of POS tags.

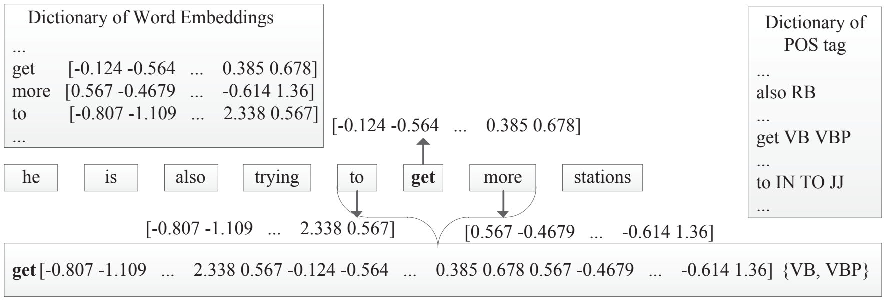

Dictionary of Word Embeddings
...
get [-0.124 -0.564 ... 0.385 0.678]
more [0.567 -0.4679 ... -0.614 1.36]
to [-0.807 -1.109 ... 2.338 0.567]
...
Dictionary of POS tag
...
also RB
...
get VB VBP
...
to IN TO JJ
...
[-0.124 -0.564 ... 0.385 0.678]
he
is
also
trying
to
get
more
stations
[-0.807 -1.109 ... 2.338 0.567]
[0.567 -0.4679 ... -0.614 1.36]
get [-0.807 -1.109 ... 2.338 0.567 -0.124 -0.564 ... 0.385 0.678 0.567 -0.4679 ... -0.614 1.36] {VB, VBP}

Fig. 2. An example of how the training data is generated.

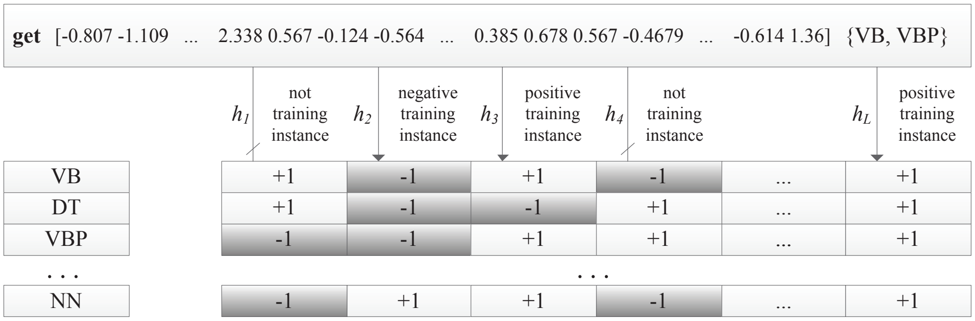
3.3. Training and Testing based on Constrained ECOC

In this section, we describe our proposed approach based on constrained ECOC for solving the weakly supervised POS tagging problem, which does not rely on disambiguating possible tags. Constrained ECOC follows the binary decomposition strategy via an encoding-decoding procedure for multi-class classifier induction.

Firstly, in the encoding phase, a $|O| \times L$ binary coding matrix $M \in \{+1, -1\}^{|O| \times L}$ is needed where $|O|$ is the number of distinct POS tags. Each row of the coding matrix $M(j, :)$ represents an L-bit codeword for one tag y_j (See the right half of Figure 1). Each column of the coding matrix $M(:, l)$ specifies a dichotomy over the tag space y with $y_l^+ = \{y_j | M(j, l) = +1, 1 \le j \le |O|\}$ and $y_l^- = \{y_j | M(j, l) = -1, 1 \le j \le |O|\}$. Then, one binary classifier is built for each column by treating training examples from y_l^+ as positive ones and those from y_l^- as negative ones. For each training instance, $(\phi(w_i), A_i)$, where $\phi(w_i)$ is the feature vector of the word w_i and A_i is its possible POS tags which are retrieved from the dictionary D. The possible tag set A_i associated with w_i is regarded as an entirety. The training instance $(\phi(w_i), A_i)$ will be used as a positive (or negative) training example only if A_i entirely falls into y_l^+ (or y_l^-) to build the binary classifier h_l. Otherwise, $(\phi(w_i), A_i)$ will not be used in the training process of h_l.

An example of how the training instance is used is illustrated in Figure 3. For the training instance "[-0.807 -1.109 ... 2.338 0.567 -0.124 -0.564 ... 0.385 0.678 0.567 -0.4679 ... -0.614 1.36], {VB, VBP}" which is generated in Figure 2, it can be used as a positive training example for h_3 and h_L as {VB, VBP} entirely falls into y_3^+ and y_L^+. Similarly, it can be used as a negative training example for h_2 as {VB, VBP} entirely falls into y_2^-. It can not be used for h_1 and h_4.

Then, for any test word w^*, an L-bit codeword $h(\phi(w^*))$ is generated by concatenating the predicted outputs of the L binary classifiers: $h(\phi(w^*)) =$


get [-0.807 -1.109 ... 2.338 0.567 -0.124 -0.564 ... 0.385 0.678 0.567 -0.4679 ... -0.614 1.36] {VB, VBP}
h_1
not training instance
h_2
negative training instance
h_3
positive training instance
h_4
not training instance
h_L
positive training instance

VB	+1	-1	+1	-1	...	+1
DT	+1	-1	-1	+1	...	+1
VBP	-1	-1	+1	+1	...	+1
. . .			. . .			
NN	-1	+1	+1	-1	...	+1

Fig. 3. An example of how the training instance is used in ECOC.

$[h_1(\phi(w^*)), h_2(\phi(w^*)), \cdots, h_L(\phi(w^*))]^{\mathrm{T}}$. After that, the tag whose codeword is closest to $h(\phi(w^*))$ is returned as the final prediction for w^*:

$$g(\phi(w^*)) = \underset{\substack{y_j \\ 1 \leq j \leq |O|}}{\arg\min} \; dist(h(\phi(w^*)), M(j,:)) \tag{2}$$

Here, the distance function $dist(,)$ can be implemented in various ways such as hamming distance [Dietterich and Bakiri 1995] or Euclidean distance [Pujol et al. 2008]. Table III lists various distance measurement functions and their corresponding definitions.

Table III. Definitions of different distance measurement functions.

Distance Measurement	**Definition**
Euclidean	$\sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$
Attenuated Euclidean	$\sqrt{\sum_{i=1}^{n} \lvert y_i \rvert \lvert x_i \rvert (x_i - y_i)^2}$
Hamming	$\sum_{i=1}^{n}(1 - sign(x_i \cdot y_i))/2$
Inverse Hamming	$\max(\Delta^{-1}H^T)$, where $\Delta(i_1, i_2)$ = Hamming_Dist(y_{i1}, y_{i2}) and H is the vector of Hamming decoding values of the x for each y_i.
Laplacian	$(\alpha_i + 1)/(\alpha_i + \beta_i + \lvert O \rvert)$, where α_i is the number of matched positions between the codeword x and y, β_i is the number of missmatches without considering the positions coded with 0.

As for a test word w^*, its candidate POS tags A_{w^*} can be found in the dictionary D. The final prediction for w^*, $g(\phi(w^*))$ must be in its candidate POS tags. To apply such constrains, Equation 2 is modified as

$$g(\phi(w^*)) = \underset{\substack{y_j \\ 1 \leq j \leq |O| \\ y_j \in A_{w^*}}}{\arg\min} \; dist(h(\phi(w^*)), M(j,:)) \tag{3}$$

The proposed approach based on constrained ECOC is summarized in Algorithm 2. It can be seen that the proposed approach does not rely on any POS tag disambiguation on the candidate label set for any word. The procedure is conceptually simple and amenable to different choices of the binary learner $\mathfrak{B}$, similar to the standard ECOC mechanism. Furthermore, as reported in the next section, the performance of the proposed approach is highly competitive against the state-of-the-art weakly supervised POS tagging approaches.

Algorithm 2 Training and Testing based on constrained ECOC.

```
Inputs: $L$, $\mathfrak{B}$, $threshold$, $T$, $w^*$ (the test word in a given sentence)
Outputs: The predicted POS tag for $w^*$
Encoding:
$l = 0$;
do
    Randomly generate a $|O|$-bit column coding $v = [v_1, v_2, \cdots, v_{|O|}]^{\mathrm{T}} \in \{-1, +1\}^{|O|}$;
    Dichotomize the tag space according to $v$: $y_v^+ = \{y_j | v_j = +1, 1 \leq j \leq |O|\}, y_v^- = y \backslash y_v^+$;
    Initialize the binary training set $T_v = \varnothing$;
    for each word $w_i$ appeared in $U$ do
        if $A_i \subseteq y_v^+$ then
            add $((\phi(w_i), A_{w_i}), +1)$ to $T_v$
        end if
        if $A_i \subseteq y_v^-$ then
            add $((\phi(w_i), A_{w_i}), -1)$ to $T_v$
        end if
    end for
    if $|T_v| \geq threshold$ then
        $l = l + 1$;
        Set the $l$-th column of the coding matrix $M$ to $v$;
        Build the binary classifier $h_l$ by invoking $\mathfrak{B}$ on $T_v$;
    end if
while $l < L$
Decoding:
Generate $\phi(w^*)$, the feature of $w^*$, based on Algorithm 1;
Generate codeword $h(\phi(w^*))$ by querying binary classifiers' outputs;
Return $y^* = g(x^*)$ according to Equation 3.
```

4. EXPERIMENTS

4.1. Setup

We evaluate English POS tagging on Penn Treebank III (PTB) [Marcus et al. 1993]. Following the same experimental setup as in [Garrette and Baldridge 2012; Ravi et al. 2010; Ravi et al. 2014], we construct a dictionary D from the entire Wall Street Journal data in PTB. There are 45 distinct POS tags in PTB such as PRP, DT, CD, IN mentioned in Table I, which form O. The dictionary contains 48,461 words and 56,602 word/tag pairs. We also build an unannotated corpus U by choosing the first 50,000 tokens of PTB. Following a similar setup in previous methods [Ravi and Knight 2009; Yatbaz and Yuret 2010], we construct a standard test data by collecting 24,115 word tokens from PTB. In the 24k test set, there are 5,175 distinct words with 8,162 word/tag pairs found in the dictionary D.

In order to fairly compare the proposed approach with the state-of-the-art approaches, we also build larger datasets with different number of word tokens ranging from 48k, 96k and 193k to the entire PTB in addition to the standard 24k dataset. Figure 4 shows the percentage of words with different number of possible POS tags on different test sets. It can be observed that the unambiguous words (with one POS tag only) approximately account for less than 45% of all words while more than 70% of ambiguous words are with no more than four possible POS tags.

The dictionary D derived from the entire PTB is quite noisy due to the tagging errors. For example, in the tagged sentence "... the/CD 1982/CD Salon/NNP is/VBZ a/DT

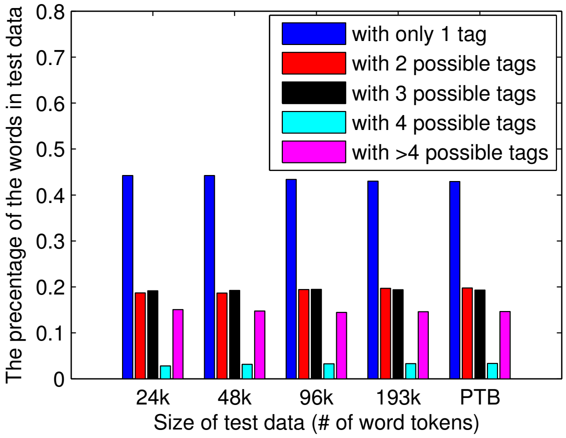


Fig. 4. Distribution of words with different number of possible tags on 24k test set.

beautiful/JJ wine/NN ...", "the" is wrongly tagged as "CD". To remove the noisy tags, we correct the tag dictionary following a similar strategy as in [Goldberg et al. 2008].

As mentioned before, word embeddings can be trained using neural language models [Collobert et al. 2011]. We use the word embeddings which were trained on the entire English Wikipedia (November 2007 version)[1]. To represent the context features of a target word, we concatenate the word embedding of the first left word, the target word and first right word to form a 150-dimensional vector of $[w_{i-1}, w_i, w_{i+1}]$ and use it as the feature vector of the target word. For words not appeared in the pre-trained word embeddings, we assign the word embeddings of other words to them following some simple morphological rules. The most frequent 20 suffixes are chosen to handle unknown words such as "tion","ness", "ment" and so on. For example, if the suffix of a word w is "ing", we randomly select a word with "ing" and assign its word embedding to w. For a hyphenated word, we assign the word embedding of the latter part to this word.

The codeword length L is set to $\lceil 10 \log_2(|O|) \rceil$, as is typically set in ECOC-based approaches [Zhou 2012]. The binary learner $\mathfrak{B}$ is chosen to be Support Vector Machines (SVMs) using the implementation of Libsvm [Chang and Lin 2011]. The thresholding parameter $threshold$ is set to $\frac{1}{10}|U|$.

4.2. Baselines Construction

To evaluate the efficiency of the proposed framework for weakly supervised POS tagging, we choose the following approaches as the baselines and compare the performance on the standard test data (24k tokens) as well as larger test data (48k, 96k, 193k and the entire PTB) for POS tagging.

(1) HMM: Training a bigram HMM model using an EM algorithm.
(2) IP+EM [Ravi and Knight 2009]: Using IP to search the smallest bi-gram POS tag set and using this set to constrain the training of EM.
(3) MIN-GREEDY [Ravi et al. 2010]: Minimizing grammar size using the two-step greedy method.
(4) DMLC+EM [Ravi et al. 2014]: An extension of MIN-GREEDY with a fast, greedy algorithm with formal approximation.

[1] ronan.collobert.com/senna/

(5) RD [Yatbaz and Yuret 2010]: Unambiguous substitutes are chosen for each occurrence of an ambiguous word based on its context using a standard HMM model with a filtered dictionary.

4.3. Overall Results

Table IV shows the performance comparison results of unsupervised POS tagging on different test sets. Here, Laplacian decoding is used to implement the distance function between two binary codewords. Other distance metrics have also been evaluated and the details will be elaborated in Section 4.4.

Table IV. Performance comparison of weakly supervised POS tagging on different test sets. (– represents that no result was reported on the test set for this method).

Methods	Tagging Accuracy				
	24k	48k	96k	193k	PTB
HMM	81.7%	81.4%	82.8%	82.0%	82.3%
IP+EM	91.6%	89.3%	89.5%	91.6%	–
MIN-GREEDY	91.6%	88.9%	89.4%	89.1%	87.1%
DMLC+EM	91.4%	–	–	–	87.5%
RD	92.25%	92.47%	–	–	–
Our approach	**93.21%**	**93.15%**	**93.01%**	**92.77%**	**92.63** %

It can be observed that our approach achieves the best performance on the 24k data, with an accuracy of 93.21%. With the increasing size of the test set, the performance of the proposed approach decreases slightly. It might attribute to the fact that with a larger test set, there is an increased likelihood that some words in the test set might have not been well learned in training data. Therefore, the performance of the proposed approach on larger test sets is slightly worse than that on smaller test sets. Nevertheless, our approach outperforms all the baselines on all the test sets with the improvements ranging from 0.68% to 11.51% on accuracy. Overall, we see superior performance achieved by our proposed approach.

To investigate the degree of disambiguation achieved by our proposed approach, we analyze the accuracy of POS tagging on words with different number of possible tags, 1 (unambiguous), 2, 3, 4 and more than 4. As shown in Figure 5, the accuracy of POS tagging on words with only one POS tag is 100%. For words with 2 to 4 possible tags, the POS tagging accuracy of our approach is fairly stable. We observe that the accuracy on words with 2 possible tags is less than 90% but the accuracy on words with 3 possible tags is around 90%. This is somewhat contrary to our prior belief. By further analyzing the results, we found that a majority of words with two possible POS tags are those tagged with either (VB, VBP) or (VBD, VBN). Since VB and VBP co-occur quite often in the dictionary D and similarly for VBD and VBN, these two pairs of tags are difficult to be disambiguated by our approach. It can be observed that the accuracy of POS tagging on words with 4 possible tags is lower than the accuracy on words with > 4 possible tags. It might due to insufficient training data for the words with 4 possible tags as has been previously shown in Figure 4.

4.4. The Impact of Different Distance Functions

As described in Section 3.3, various distance functions can be used to decode the codewords of the target word w. To investigate the impact of decoding, we conducted experiments using different distance functions on different sizes of test sets with the 50k train set. The performance of POS tagging with different distance measures as have been previously described in Table III are presented in Figure 6. It can be observed that

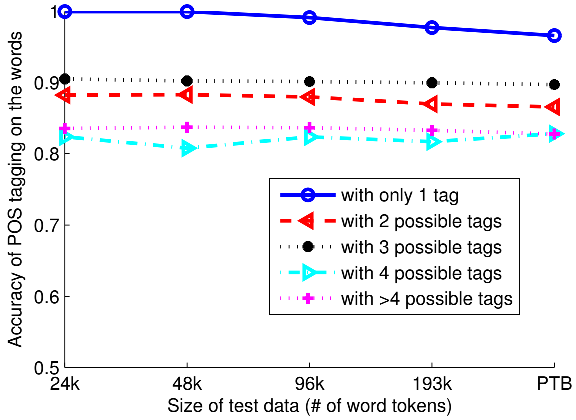


Fig. 5. Accuracy of words with different number of possible tags on test sets with varying sizes.

Laplacian performs the best while *Inverse Hamming* gives the worst results across all test sets. Other distance functions generate very similar results.

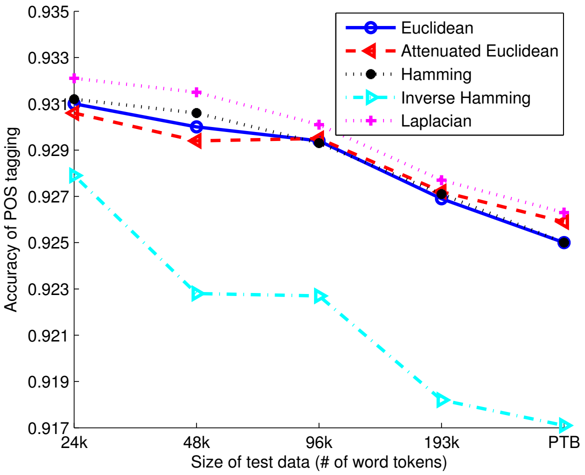


Fig. 6. Performance comparison using different distance functions on test sets with varying sizes.

4.5. The Impact of Difference Sizes of Unannotated Corpus U

In this subsection, we investigate how the POS tagging performance changes with different sizes of the unannotated data U. It can be observed from Table V that for some bigger test sets such as 193k and PTB, the performance of the proposed approach increases gradually and then converges with more un-annotated data. This is inline with what we expected for weakly-supervised training. However, for smaller test sets, the performance of the proposed approach fluctuates slightly. This could be due to the fact that the distribution of tags in smaller test sets might be slightly different from that in the training set. Hence, adding more unannotated data would not necessarily lead to increased accuracy.

Table V. Performance comparison of the proposed approach trained on U with different sizes.

Size of U	Tagging Accuracy				
	24k	48k	96k	193k	PTB
50k	**93.21%**	**93.15%**	93.01%	92.77%	92.63%
100K	93.10%	93.10%	**93.18%**	93.05%	92.87%
150k	93.20%	93.09%	93.17%	**93.11%**	**92.91%**
200K	93.09%	93.02%	93.09%	93.04%	**92.91%**

4.6. The impact of Dictionary D

In reality, it might be difficult to build a complete dictionary consisting of all possible words each with a correct set of POS tags. Therefore, it will be interesting to see how the proposed framework performs when provided with an incomplete dictionary, meaning that some words in the test data cannot be found in the dictionary.

We build a dictionary derived from section $00-15$ in PTB. It consists of $39,087$ words and $45,331$ word/tag entries. We use section 16 as raw data and perform final evaluation on the sections $22-24$. We use the raw corpus along with the unlabeled test data to train the proposed model. Unknown words are allowed to have all possible tags.

We compare the performance of our approach with several baselines in Table VI. The “Random” baseline simply chooses a tag randomly from the tag dictionary and gives an accuracy of 63.53%. “EM” uses the standard EM algorithm and achieves an accuracy of 69.20%. The “Type+HMM” system [Garrette and Baldridge 2012] learned taggers based on HMM from incomplete tag dictionaries. It improves MIN-REEDY algorithm [Goldberg et al. 2008] with several intuitive heuristics and achieves 88.52% in accuracy. As far as we know, it is the best score reported for this task in the literature. Our proposed approach gives an accuracy of 91.52%, outperforming all the baselines including the state-of-the-art approach, Type+HMM. One possible reason is that our proposed approach constructed features from word embeddings. Thus words in the test data which are unseen in the POS tag dictionary D might still exist in the learned word embeddings from Wikipedia.

Table VI. Performance comparison with an incomplete dictionary. The dictionary is derived from section $00-15$ and test data is from section $22-24$ of PTB.

Method	Accuracy (%)
Random	63.53
EM	69.20
DMLC+EM	88.11
Type+HMM	88.52
Our approach	**91.52**

4.7. The Impact of POS Tag Space

To evaluate the performance of our proposed framework with a coarse grained dictionary, we use a reduced tag set of 17 tags instead of the full 45-tag set and conduct experiments on the standard 24k test data, following a similar experimental setup as in previous approaches [Garrette and Baldridge 2012; Ravi et al. 2010; Ravi et al. 2014]. The details of the reduction of POS Tag are presented in Table VII.

Table VIII summarizes the previously reported results on coarse grained POS tagging. “BH-MM” is a fully Bayesian approach that uses sparse POS priors and achieves an accuracy of 87.3%, “CE” is based on the HMM model using contrastive estimation method and achieves an accuracy of 88.7%. It can be observed that our approach

Table VII. The reduced tag set with 17 tags.

Reduced Tag	Treebank tag
ADJ	CD JJ JJR JJS PRP$
ADV	RB RBR RBS
DET	DT PDT
INPUNC	,:LS SYM UH
LPUNC	“ -LRB
N	EX FW NN NNP NNPS NNS PRP
RPUNC	” -RRB-
W	WDT WP$ WP WRB
V	MD VBD VBP VB VBZ

achieves an accuracy of 95.4%, outperforming most baselines, except “IP+EM” where our approach is only 1.4% lower.

Table VIII. Performance comparison of the proposed framework with the baseline approaches using 17-tagset on the standard 24k test data.

Method	Accuracy
BH-MM	87.3%
CE	88.7%
IP+EM	96.8%
RD	92.9%
Our approach	**95.4%**

4.8. The Impact of Constrained ECOC

As mentioned in Section 3.3, the final prediction for w^*, $g(\phi(w^*))$ must be in its candidate POS tags. Therefore, a constrain is applied in Equation 2 for predicting the POS tag. To investigate the impact of using constrained ECOC, we conducted experiments on different test sets with or without such a constrain. It can be observed from Figure 4.8 that the performance of the proposed model with the constrain outperforms the one without. It further verifies the effectiveness of incorporating such a constrain.

4.9. The Impact of Features Used

To find out whether the accuracy gain of the proposed method is due to the incorporation of word embeddings, we compare the performance of the proposed approach with or without using word embeddings. When not using word embeddings, we use the manually designed features instead, such as POS induction features (e.g., whether contains digit, hyphen) and word alignment features (e.g., prefix, suffix and stemming), following the same set of features as previously used in [Ravi et al. 2010]. Experimental results are presented in Table IX. The size of U is set to 50k and the whole PTB is used as the test set. It can be observed that the proposed approach achieved similar performance with or without using word embeddings. Nevertheless, using word embeddings avoids expensive feature engineering.

Table IX. Performance comparison of the proposed approach with or without using word embeddings.

Features	Accuracy
Word Embeddings	**92.63**%
Manually Constructed Features	92.45%

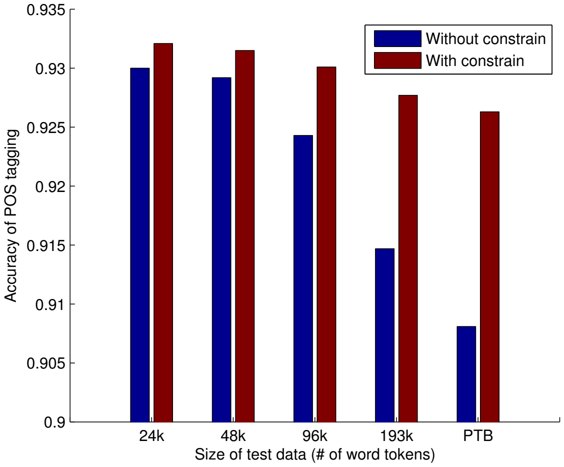


Fig. 7. Performance comparison of the proposed approach with or without the constrain on different test sets.

4.10. Experimental Results of POS Tagging on Italian and Malagasy

To explore whether the proposed approach is effective only for some specific language such as English, we conduct experiments on two other languages, Italian and Malagasy.

For Italian language, the CCG-TUT corpus[2] is used for evaluating Italian POS tagging. There are 90 distinct POS tags in CCG-TUT, which form O. The dictionary contains 8,177 words and 8,733 word/tag pairs. The unannotated corpus U was constructed using 42,100 tokens in CCG-TUT. A standard test set was constructed by collecting 21,878 word tokens from CCG-TUT. In the test set, there are 3,838 distinct words with 4,078 word/tag pairs found in the dictionary D. We download 64-dimensional word embeddings from the website[3] which were trained on over 14 million sentences extracted from the Italian Wikipedia with the window size set to 11. To represent the context features of a target word, we concatenate the word embedding of the first left word, the target word and the first right word to form a 192-dimensional vector of $[w_{i-1}, w_i, w_{i+1}]$ and use it as the feature vector of the target word.

For Malagasy language, the dataset[4] used in [Garrette and Baldridge 2013] is employed for evaluating Malagasy POS tagging. There are 44 distinct POS tags in the dataset. The dictionary contains 64,934 words and 67,256 word/tag pairs. The held-out test set contains 1,602 words and 1,683 word/tag pairs(5303 tokens). To generate Malagasy word embeddings, we download the whole Malagasy Wikipedia [5] and extract 290k sentences extracted from the corpus for generating 128-dimensional word embeddings using word2vec[6].

Table X shows the experimental results of the proposed approach and some baseline approaches on Italian and Malagasy POS tagging. It can be observed that our proposed approach achieves an accuracy of 90.9% on Italian and an accuracy of 84.5% on

[2] www.di.unito.it/~tutreeb/CCG-TUT
[3] tanl.di.unipi.it/embeddings/overview.html
[4] github.com/dhgarrette/low-resource-pos-tagging-2013
[5] We use the dump file “mgwiki-20161201-pages-articles-multistream.xml.bz2”
[6] code.google.com/p/word2vec/

Malagasy, which are better than all the baselines. It shows that our proposed approach works well across different languages.

Table X. Performance comparison of the proposed approach for Italian and Malagasy POS tagging.

Italian		Malagasy	
Method	Accuracy	Method	Accuracy
EM	83.4%	[Garrette and Baldridge 2013]	80.7%
IP	88.0%	DMLC+EM	81.1%
MIN-GREEDY	88.0%		
Our approach	**90.9%**	Our Approach	**84.5%**

5. CONCLUSIONS AND FUTURE WORK

In this paper, we propose a novel approach based on constrained ECOC for weakly supervised POS tagging. It does not require an iterative training procedure for POS tag disambiguation. Any word will be treated as a positive or negative training example only if its possible tags entirely falls into the positive or negative dichotomy specified by the column coding in ECOC. In this way, the set of possible tags of each word is treated as an entirety without resorting to any disambiguation procedure. Moreover, features employed for POS tagging are generated without manual intervention. We have evaluated the proposed approach on three corpora for English, Italian and Malagasy POS tagging, and observed a significant improvement in accuracy compared to the state-of-the-art approaches. In the future, we will investigate other ways to generate the coding matrix for possible performance improvement. Also, we will explore other disambiguation-free approaches for weakly supervised POS tagging.

Acknowledgments

This work was funded by the National Natural Science Foundation of China (61528302), the Jiangsu Natural Science Funds (BK20161430), and the Collaborative Innovation Centre of Wireless Communications Technology.

REFERENCES

Omri Abend, Roi Reichart, and Ari Rappoport. 2010. Improved unsupervised POS induction through prototype discovery. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 1298–1307.

Michele Banko and Robert C. Moore. 2004. Part of Speech Tagging in Context. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING '04)*. Association for Computational Linguistics, Stroudsburg, PA, USA.

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 (HLT '10)*. Association for Computational Linguistics, Stroudsburg, PA, USA, 582–590. http://dl.acm.org/citation.cfm?id=1857999.1858082

Chris Biemann. 2006. Unsupervised part-of-speech tagging employing efficient graph clustering. In *Proceedings of the 21st international conference on computational linguistics and 44th annual meeting of the association for computational linguistics: student research workshop*. Association for Computational Linguistics, 7–12.

Eric Brill. 1992. A simple rule-based part of speech tagger. In *Proceedings of the workshop on Speech and Natural Language*. Association for Computational Linguistics, 112–116.

Eric Brill. 1995. Unsupervised learning of disambiguation rules for part of speech tagging. In *Proceedings of the third workshop on very large corpora*, Vol. 30. Somerset, New Jersey: Association for Computational Linguistics, 1–13.

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. *Comput. Linguist.* 18, 4 (Dec. 1992), 467–479. http://dl.acm.org/citation.cfm?id=176313.176316

Daniel M. Cer, Marie Catherine De Marneffe, Daniel Jurafsky, and Christopher D. Manning. 2010. Parsing to Stanford Dependencies: Trade-offs between Speed and Accuracy. In *International Conference on Language Resources and Evaluation, Lrec 2010, 17-23 May 2010, Valletta, Malta*.

Chih-Chung Chang and Chih-Jen Lin. 2011. LIBSVM: A Library for Support Vector Machines. *ACM Trans. Intell. Syst. Technol.* 2, 3, Article 27 (May 2011), 27 pages.

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 (EMNLP '10)*. Association for Computational Linguistics, Stroudsburg, PA, USA, 575–584. http://dl.acm.org/citation.cfm?id=1870658.1870714

Alexander Clark. 2003. Combining Distributional and Morphological Information for Part of Speech Induction. In *Proceedings of the Tenth Conference on European Chapter of the Association for Computational Linguistics (EACL '03)*. 59–66.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural Language Processing (Almost) from Scratch. *Journal of Machine Learning Research* 12 (Nov. 2011), 2493–2537.

Thomas G. Dietterich and Ghulum Bakiri. 1995. Solving Multiclass Learning Problems via Error-correcting Output Codes. *Journal of Artificial Intelligence Research* 2, 1 (Jan. 1995), 263–286.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics (ACL '05)*. Association for Computational Linguistics, Stroudsburg, PA, USA, 363–370. DOI:http://dx.doi.org/10.3115/1219840.1219885

Dan Garrette and Jason Baldridge. 2012. Type-supervised hidden markov models for part-of-speech tagging with incomplete tag dictionaries. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Association for Computational Linguistics, 821–831.

Dan Garrette and Jason Baldridge. 2013. Learning a Part-of-Speech Tagger from Two Hours of Annotation. In *HLT-NAACL*. Citeseer, 138–147.

Yoav Goldberg, Meni Adler, and Michael Elhadad. 2008. EM Can Find Pretty Good HMM POS-Taggers (When Given a Good Start).. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. 746–754.

Sharon Goldwater and Tom Griffiths. 2007. A fully Bayesian approach to unsupervised part-of-speech tagging. In *ACL 2007, Proceedings of the Meeting of the Association for Computational Linguistics, June 23-30, 2007, Prague, Czech Republic*. 744–751.

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 (HLT-NAACL '06)*. Association for Computational Linguistics, Stroudsburg, PA, USA, 320–327. DOI:http://dx.doi.org/10.3115/1220835.1220876

Mark Johnson. 2007. Why Doesn't EM Find Good HMM POS-Taggers?. In *EMNLP-CoNLL 2007, Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, June 28-30, 2007, Prague, Czech Republic*. 296–305.

Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a Large Annotated Corpus of English: The Penn Treebank. *Computational Linguistics* 19, 2 (June 1993), 313–330.

Bernard Merialdo. 1994a. Tagging English Text with a Probabilistic Model. *Computational Linguistics* 20, 2 (1994), 155–171.

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

Tahira Naseem, Benjamin Snyder, Jacob Eisenstein, and Regina Barzilay. 2009. Multilingual Part-of-speech Tagging: Two Unsupervised Approaches. *J. Artif. Int. Res.* 36, 1 (Sept. 2009), 341–385. http://dl.acm.org/citation.cfm?id=1734953.1734961

Oriol Pujol, Sergio Escalera, and Petia Radeva. 2008. An Incremental Node Embedding Technique for Error Correcting Output Codes. *Pattern Recognition* 41, 2 (Feb. 2008), 713–725.

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

Sujith Ravi, Sergei Vassilivitskii, and Vibhor Rastogi. 2014. Parallel Algorithms for Unsupervised Tagging. *Transactions of the Association for Computational Linguistics* 2 (2014), 105–118.

Sujith Ravi, Ashish Vaswani, Kevin Knight, and David Chiang. 2010. Fast, greedy model minimization for unsupervised tagging. In *Proceedings of the 23rd International Conference on Computational Linguistics*. Association for Computational Linguistics, 940–948.

Kairit Sirts, Jacob Eisenstein, Micha Elsner, and Sharon Goldwater. 2014. POS induction with distributional and morphological information using a distance-dependent Chinese restaurant process. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Baltimore, Maryland, 265–271.

Noah A. Smith and Jason Eisner. 2005. Contrastive Estimation: Training Log-linear Models on Unlabeled Data. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics (ACL '05)*. Association for Computational Linguistics, Stroudsburg, PA, USA, 354–362.

Kristina Toutanova, Mark Johnson, and others. 2007. A Bayesian LDA-based model for semi-supervised part-of-speech tagging.. In *NIPS*. 1521–1528.

Mehmet Ali Yatbaz and Deniz Yuret. 2010. Unsupervised part of speech tagging using unambiguous substitutes from a statistical language model. In *Proceedings of the 23rd International Conference on Computational Linguistics*. Association for Computational Linguistics, 1391–1398.

Minling Zhang. 2014. Disambiguation-free partial label learning. In *Proceedings of the 14th SIAM International Conference on Data Mining (SDM'14)*. 37–45.

Qiuye Zhao and Mitch Marcus. 2009. A Simple Unsupervised Learner for POS Disambiguation Rules Given Only a Minimal Lexicon. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2 - Volume 2 (EMNLP '09)*. Association for Computational Linguistics, Stroudsburg, PA, USA, 688–697. http://dl.acm.org/citation.cfm?id=1699571.1699602

Deyu Zhou, Liangyu Chen, and Yulan He. 2015. An Unsupervised Framework of Exploring Events on Twitter: filtering, Extraction and Categorization. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI-15)*. 2468–2474.

Zhi-Hua Zhou. 2012. *Ensemble Methods: Foundations and Algorithms* (1st ed.). Chapman & Hall/CRC.