Weakly Supervised POS Tagging without Disambiguation

DEYU ZHOU, ZHIKAI ZHANG, and MIN-LING ZHANG, School of Computer Science and Engineering, Key Laboratory of Computer Network and Information Integration, Ministry of Education, Southeast University, Nanjing, China
YULAN HE, School of Engineering and Applied Science, Aston University, UK

Weakly supervised part-of-speech (POS) tagging is to learn to predict the POS tag for a given word in context by making use of partial annotated data instead of the fully tagged corpora. Weakly supervised POS tagging would benefit various natural language processing applications in such languages where tagged corpora are mostly unavailable.

In this article, we propose a novel framework for weakly supervised POS tagging based on a dictionary of words with their possible POS tags. In the constrained error-correcting output codes (ECOC)-based approach, a unique L-bit vector is assigned to each POS tag. The set of bitvectors is referred to as a coding matrix with value $\{1, -1\}$. Each column of the coding matrix 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: each pair of words and its possible POS tags are 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 in context, its POS tag is predicted by concatenating the predictive outputs of the L binary classifiers and choosing the tag 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% individually, which shows a significant improvement compared to the state-of-the-art approaches.

CCS Concepts: • **Computing methodologies** → **Natural language processing**; **Lexical semantics**;

Additional Key Words and Phrases: POS tagging, weakly supervised, error-correcting output codes, disambiguation

ACM Reference format:
Deyu Zhou, Zhikai Zhang, Min-Ling Zhang, and Yulan He. 2018. Weakly Supervised POS Tagging without Disambiguation. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.* 17, 4, Article 35 (July 2018), 19 pages.
https://doi.org/10.1145/3214707

This work was funded by the National Key Research and Development Program of China (2016YFC1306704), the National Natural Science Foundation of China (61772132, 61528302, 61573104), the Jiangsu Natural Science Funds (BK20161430) and Innovate UK (103652).

Authors' addresses: D. Zhou, Z. Zhang, and M.-L. Zhang, School of Computer Science and Engineering, Southeast University, No 2, Sipailou, Nanjing, 210096, China; emails: {d.zhou, 220151517, zhangml}@seu.edu.cn; Y. He, School of Engineering and Applied Science, Aston University, Birmingham, B4 7ET, UK; email: y.he@cantab.net.

https://doi.org/10.1145/3214707

1 INTRODUCTION

Part-of-speech (POS) tagging is to assign a particular part of speech to a word in a text based on its definition and context. It is crucial for further natural language processing (NLP) components such as named entity recognition [37], syntactic parsing [9], and event extraction [36]. Methods for POS tagging fall into two distinctive categories: rule-based and machine-learning-based. Rule-based approaches rely on manually designed rules while the performance of machine-learning-based approaches depends on the size and quality of the annotated corpora.

However, there are more than 50 low-density languages where both tagged corpora and language speakers are mostly unavailable [11]. Weakly supervised POS tagging might benefit NLP in such languages. In this article, weakly supervised POS tagging is to learn to predict POS tagging for a given word in context given a dictionary of words with their possible POS tags as shown in Table 1. However, it is difficult to conduct weakly supervised POS tagging since the ground-truth POS tag of the word in the sentence is hidden in its possible POS tags and is not directly accessible by the learning algorithm. One common way to learn from the dictionary of candidate POS tags is to regard the ground-truth tag as a latent variable that is identified via iterative refining procedure. Therefore, 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 [22] employs maximum likelihood estimation to train a trigram HMM. Following this way, some improvements are achieved by modifying the statistical model or employing better parameter estimation techniques. For example, Banko and Moore [3] modify the basic HMM structure to employ the context on both sides of the word to be tagged. In Reference [30], contrastive estimation is employed on CRF for POS tagging.

Regarding the ground-truth tag as latent variable, most of the approaches mentioned above are based on disambiguation. Although disambiguation presents as an intuitive and reasonable strategy to weakly supervised POS tagging, its effectiveness is largely affected by the false positive tag(s) within possible tags. For disambiguation in ground-truth tag identification, the identified tag refined in each iteration might turn out to be the false positive label instead of the ground truth one. Therefore, the negative influence brought by false positive tags will be more pronounced as the size of possible tags increases.

In this article, we propose a novel strategy for weakly supervised POS tagging. It does not rely on disambiguating possible POS tags. In specific, error-correcting output codes (ECOC) [14], one of the famous multi-class learning techniques is adapted. A unique L-bit vector is assigned to each POS tag. The set of bitvectors is referred to as coding matrix and denoted as M with value $\{1, -1\}$. Each column of the coding matrix 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 $[-1, +1, -1, +1]^T$ separates the tag space into the negative dichotomy {VB, VBP} and the 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 M, the binary classifier is built based on binary 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 its possible tags entirely 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 address their tasks by exploring the lexical context of the words to be tagged and their letter structure (e.g., presence of suffixes, capitalization, and hyphenation). Obviously, such feature design needs domain knowledge and expertise. In this article, features employed for weakly supervised POS tagging are generated based on neural language modeling without manual intervene.

Table 1. An Example of Input and Output of Weakly Supervised POS Tagging (PRP Stands for Personal Pronoun, DT for Determiner, JJ for Adjective, VB for Verb Base Form, CD for Cardinal Number, and So On)

Dictionary of words with their 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;...	
Sentence	POS tagging
You have to recognize that these events took place 35 years ago.	You/PRP have/VBP to/TO recognize/VB that/IN these/DT events/NNS took/VBD place/NN 35/CD years/NNS ago/IN ./.

The main contributions of the article are summarized below:

—We proposed a novel framework based on constrained ECOC for weakly supervised POS tagging. In such a way, the set of a word's possible tags is treated as an entirety without resorting to any disambiguation procedure. It can easily avoid the disadvantage of disambiguation strategy, a common way for 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 modeling.
—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

Satisfactory results have been achieved in supervised POS tagging. The best taggers can obtain tagging 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 [11]. Some of them are even dead. Therefore, POS tagging without using the fully annotated corpora is vital, but full of challenge. An increasing number of researchers tackle this problem [34]. Generally, based on the way of using the annotated data, there are three directions for handling the task: POS induction, where no prior knowledge is employed, POS disambiguation, where a dictionary of words and their possible tags is used, and prototype-driven learning where a small set of prototypes for each POS tag is provided instead of a dictionary.

One the one hand, for fully unsupervised POS tagging, POS induction, some efforts have been made using the clustering techniques by casting the identification of POS tags into a knowledge-free clustering problem. Brown et al. [8] employ an n-gram model based on classes of words. It aims to optimize 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 [12] employs morphological information in the clustering so that morphologically similar words are clustered together. Based on a standard trigram HMM, Goldwarter and Griffiths [18] employ a fully Bayesian approach and the use of priors is allowed. A collapsed Gibbs sampler is employed to inferring the hidden tags. Instead of only using Gibbs sampling, both variational Bayesian EM and Gibbs sampling are employed in Reference [20] and experimental results show that variational Bayesian EM converges faster than Gibbs sampling for POS tagging. Using the structure of a standard HMM, Berg-Kirkpatrick et al. [4] assume that the distributions are logistic and each component multinomial of the HMM is turned into a miniature logistic regression. Therefore, 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 is proposed in Reference [5] so that the number of POS tags (clusters) is induced automatically. Clustering is conducted on the most frequent n words and low frequency words separately and the clusters are merged together. Based on the theory of prototypes, Abend et al. [1] first cluster words based on a fine morphological representation. The most frequent words are clustered using distributional representation and landmark clusters are defined serving as the cores of the induced POS categories. The rest of the words is mapped to these categories. Kairit et al. [29] present 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 Reference [11], due to a lack of standard and informative evaluation techniques, it is difficult to compare the effectiveness of different clustering methods.

On the other hand, for weakly supervised, many researchers focused on POS disambiguation using POS tag dictionaries. In Reference [6], a rule-based POS tagger is described, which captures the learned knowledge into a set of simple deterministic rules instead of a large table of statistics. In Reference [7], an unsupervised learning algorithm is proposed for automatically training a rule-based POS tagger. Regard the ground-truth tag as latent variable, previous weakly supervised POS tagging approaches are largely based on EM parameters estimation using HMMs or CRFs. For example, given a sentence $W = w_1 w_2 \ldots w_n$ and a sequence of tags $T = t_1 t_2 \ldots t_n$, of the same length, a triclass model defined as $p(W, T) = \prod_{i=1}^{n} p(w_i|t_i)p(t_i|t_{i-2}t_{i-1})$ is employed in Reference [23]. Following this way, some improvements are achieved by modifying the statistical model or employing better parameter estimation techniques. For example, Banko and Moore [3] modify the basic HMM structure to employ the context on both sides of the word to be tagged. In Reference [30], contrastive estimation is employed on CRF for POS tagging. In Reference [31], a Bayesian model is proposed that extends the latent Dirichlet allocation model and incorporates the intuition that words' distributions over tags are sparse. Integer programming (IP) is employed to search the smallest bi-gram POS tag set and this set was used to constrain the training of EM [26]. It achieves an accuracy of 91.6% on the 24k test set, but cannot handle large dataset. For solving the deficiency of IP, a two-stage greedy minimization approach is proposed in Reference [28] that runs much faster while maintaining the performance of tagging. To further improve the performance, several heuristics are employed in Reference [15]. Moreover, it works on incomplete dictionary and achieves an accuracy of 88.52%. In Reference [27], distributed minimum label cover is proposed, which can parallelize the algorithm while preserving approximation guarantees. It achieves an accuracy of 91.4% on the 24k test set and 88.15% using incomplete dictionary. In Reference [32], unambiguous substitutes are chosen for each occurrence of an ambiguous word based on its context. It achieves an accuracy of 92.25% using standard HMM model on standard 24k test set. In Reference [24], multilingual learning is employed 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 are employed for estimating the hierarchical Bayesian models.

Instead of using tag dictionaries, a few canonical examples of each POS tag are employed based on prototype-driven learning [19]. The provided prototype information is propagated across a corpus using distributional similarity features in a log linear generative model. Following this way, a closed-class lexicon specifying possible tags is required and a disambiguation model is learned for disambiguating the occurrences of words in context [35].

Our work is similar to the second way in the sense that we also focus on POS tagging using tag dictionaries. 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 tag sets. In this article, we propose a novel approach for weakly

Table 2. 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 dataset

supervised POS tagging without disambiguation. The set of possible tags is treated as an entirety without disambiguation. Moreover, instead of manual feature engineering employed in most previous weakly supervised POS tagging approaches, features for training and testing in the proposed framework are automatically generated using neural language modeling. The proposed approach was evaluated 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. From the perspective of machine learning, our approach falls into the partial label learning framework [33] 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. That is why we incorporate the Training Data Generation component in the proposed framework. Moreover, the tag dictionary is equally applied to both the training and testing instances. Such constrains are applied in the test data using the constrained ECOC.

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. First, 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, the encoding-decoding procedure is conducted. Table 2 lists notations used in this article. The architecture 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.

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 the 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 $B(B > 2)$ labels $y_1, y_2, \ldots, y_B$, one assigns a unique L-bit vector to each label y_i. It can be viewed as a unique coding for the label. The set of bitvectors is referred to as coding matrix and denoted as M. The coding matrix M can appear in different forms such as

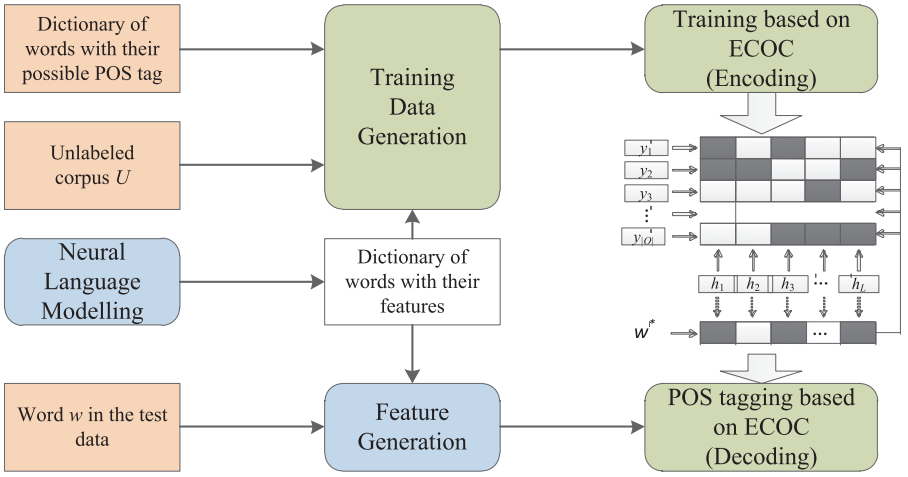

Fig. 1. The architecture of the proposed approach for weakly supervised POS tagging.

binary form [14] with value $\{+1, -1\}$ and ternary form [2] with value $\{+1, 0, -1\}$. In our proposed approach, $M \in \{+1, -1\}^{B \times L}$.

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 y_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 predicted to the class with the closest codeword according to some distance measure.

Generally, there are two popular binary coding schemes, the one-versus-rest scheme and dense random scheme, for choosing L. In the one-versus-rest scheme, each binary classifier is trained to discriminate one class against all the other classes. Obviously, the codeword is of length B, the number of classes. In the dense random scheme, Allwein et al. [2] suggested an optimal codeword length of $10 \log B$.

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 embeddings aim to capture the syntactic or semantic regularities among words such that words that are semantically similar to each other are placed in nearby locations in the embedding space. This characteristic is precisely what we want. Word embedding or word representation of each word is a real-value vector, usually with a dimension of between 50 and 300. We use neural language modeling [13] to learn word representations by discriminating the legitimate phrase from incorrect phrases.

Given a word sequence $p = (w_1, w_2, \ldots, w_d)$ 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, $\Re$ 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 dataset T = ∅;
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, φ(w);
    Retrieve from D all possible POS tags A_w for word w;
    Insert the pair (φ(w), A_w) into the training set T;
end for
T = {(φ(w_i), A_i)|1 ≤ i ≤ |U|}(w_i ∈ U, A_i ⊆ O);
```

To illustrate how the training data is generated, we present an example 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 "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.

3.3 Training and Testing Based on Constrained ECOC

In this section, we describe the 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 philosophy via an encoding-decoding procedure for multi-class classifier induction.

First, 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 that 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, 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 cannot be used in h_1, h_4.

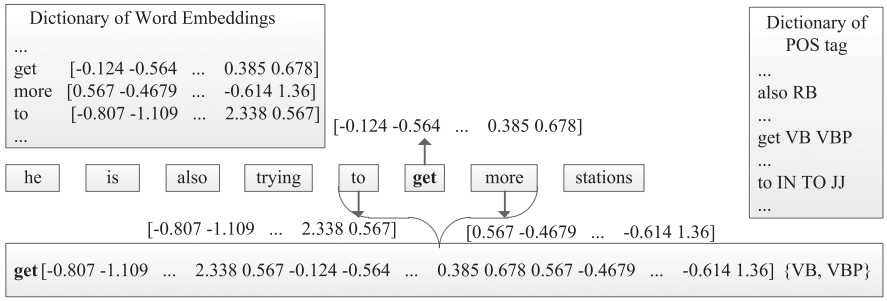

Fig. 2. An example of how the training data are generated.

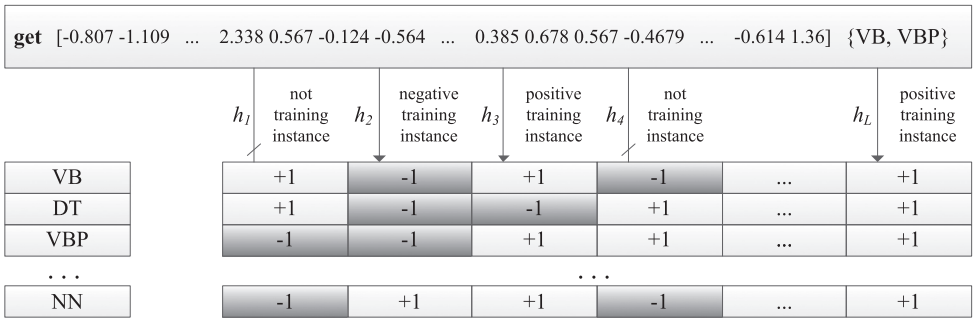

Fig. 3. An example of how the training instance is used in ECOC.

Table 3. The Definition of Different Decodings

Decoding	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 $\Delta(i_1, i_2) = \text{Hamming\_Dist}(y_{i1}, y_{i2})$ 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 codewords x and y, β_i is the number of miss-matches without considering the positions coded with 0.

Then, for any test word w^*, an L-bit codeword $h(\phi(w^*))$ is generated by concatenating the predictive outputs of the L binary classifiers: $h(\phi(w^*)) = [h_1(\phi(w^*)), h_2(\phi(w^*)), \ldots, h_L(\phi(w^*))]^{\mathrm{T}}$. After that, the tag whose codeword is closest to $h(\phi(w^*))$ is returned as the final prediction for w^*:

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

Here, the distance function $dist(,)$ can be implemented in various ways such as hamming distance [14] or Euclidean distance [25]. Table 3 lists the functions and their corresponding definitions employed in our approach.

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

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

The proposed approach based on constrained ECOC is summarized in Algorithm 2. As shown here, the proposed approach does not rely on any POS tag disambiguation strategy toward the candidate label set and is instead treated in an integrative manner. 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

Input: L, $\mathfrak{B}$, thr, T, w^* (the test word in a given sentence)
Output: The predicted POS tag for w^*

Encoding:
$l = 0$;
do
  Randomly generate a $|O|$-bit column coding $v = [v_1, v_2, \ldots, v_{|O|}]^{\mathrm{T}} \in \{-1, +1\}^{|O|}$;
  Dichotomize the tag space according to v: $y_v^+ = \{y_j | v_j = +1, 1 \le j \le |O|\}$, $y_v^- = y \backslash y_v^+$;
  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 $|T_v| \ge thr$ **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

For English POS tagging, we evaluate the proposed approach on Penn Treebank III (PTB) [21]. Following the same experimental setup as in References [15, 27, 28], we construct a dictionary D from the entire *Wall Street Journal* data in PTB. There are 45 distinct POS tags in PTB such as

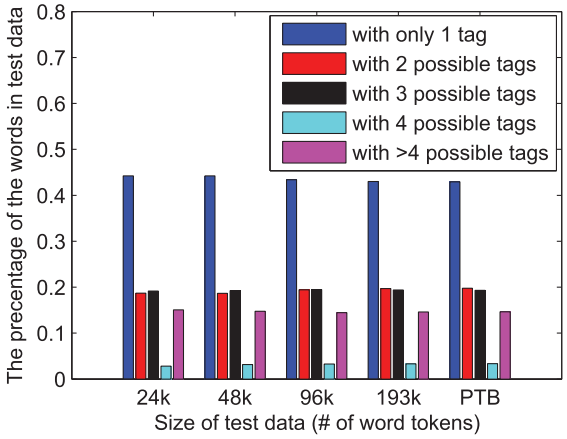

Fig. 4. Distribution of words with different number of possible tags on 24k test set.

PRP, DT, CD, IN mentioned in Table 1, 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 the similar setup in previous methods [26, 32], 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.

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 a 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 beautiful/JJ wine/NN ...", "the" is wrongly tagged as "CD." To remove the noisy tags, we correct the tag dictionary using the similar way in Reference [17].

As mentioned before, word embeddings are trained using neural language models [13]. Instead of training by ourselves, we download the word embeddings from the website,[1] which were trained on the entire English Wikipedia (November 2007 version). 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 appearing in the learnt word embeddings, we use various morphological features to assign the word embeddings of the similar words to these words. 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 [38]. The binary learner $\mathfrak{B}$ is chosen to be support vector machines, using the implementation of Libsvm [10]. The thresholding parameter thr is set to $\frac{1}{10}|U|$.

[1] ronan.collobert.com/senna/.

Table 4. Performance Comparison of Weakly Supervised POS Tagging on Different Test Sets (– Represents That No Result Was Reported on the Test Set for This Method)

	Tagging Accuracy				
Methods	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 %**

4.2 Baseline Construction

To evaluate the efficiency of the proposed framework for weakly supervised POS tagging, we choose the following approaches as the baseline 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 [26]: Using IP to search the smallest bi-gram POS tag set and using this set to constrain the training of EM.
(3) MIN-GREEDY [28]: Minimizing grammar size using the two-step greedy method.
(4) DMLC+EM [27]: An extension of MIN-GREEDY with a fast, greedy algorithm with formal approximation.
(5) RD [32]: 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 4 shows the performance comparison results of weakly supervised 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.

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 dataset, the performance of the proposed approach decreases slightly. It might attribute to, that for a larger test dataset, there is a big chance that some words in the test data might have not been well learned in training data. Therefore, the performance of the proposed approach on a larger test dataset is slightly worse than on a small test dataset. 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 a different number of possible tags, one (unambiguous), two, three, four, and more than four. As shown in Figure 5, the accuracy of POS tagging on words with only one POS tag is 100%. For words with two to four possible tags, the POS tagging accuracy of our approach is fairly stable. We observe that the accuracy on words with two possible tags is less than 90% but the accuracy on words with three 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

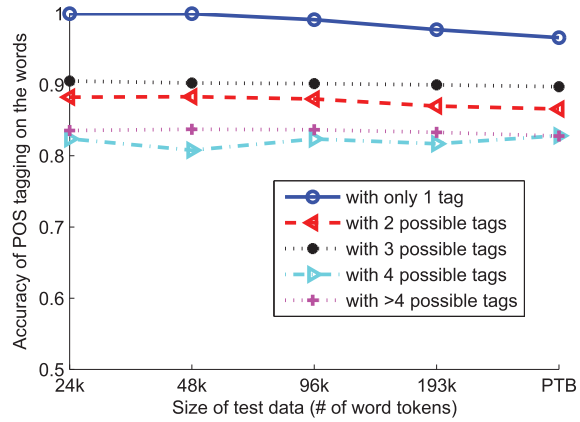

Fig. 5. Accuracy of words with different number of possible tags on 24k test set.

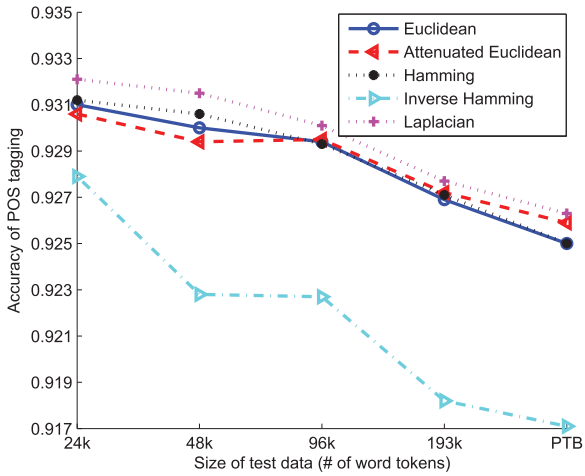

Fig. 6. Performance comparison of unsupervised POS tagging using different decodings on different test sets.

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 four possible tags is lower than the accuracy on words with > 4 possible tags. It might attribute to the insufficient training data for the words with four possible tags as shown in Figure 4.

4.4 The Impact of Different Decoding Functions

As described in Section 3.3, various distance functions can be used to decode the codewords of target word w. To investigate the impact of decoding, we conducted experiments using various distance functions on different sizes of test sets with 50k train set. The performance of POS tagging with different distance measures are presented in Figure 6 while the definitions of different decodings are presented in Table 3.

Table 5. 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%**

Table 6. Performance Comparison with an Incomplete Dictionary (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%**

4.5 The Impact of Difference Sizes of Unannotated Corpus U

In this subsection, we investigate how the POS tagging performance changes with different sizes of U. It can be observed from Table 5 that for some big test datasets such as 193k, PTB, the performance of the proposed approach increases gradually and then converges with more unannotated data, just as we expected. Generally, for weakly supervised approaches, their performance will increase and then converge with more unannotated data. However, for some small test datasets, the performance of the proposed approach fluctuates slightly. It might be explained by that the small test dataset has the big chance of sharing different distribution with the training data. The evaluation on the small test dataset is not complete and stable.

4.6 The Impact of Dictionary D

In reality, it might be difficult to build a complete dictionary consisting of all possible words 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 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 6. 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 [15] learned taggers based on HMM from incomplete tag dictionaries. It improves MIN-REEDY algorithm [17] with several intuitive heuristics and achieves 88.52% in accuracy. As far as we know,

Table 7. 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

Table 8. 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	92.90%
Our approach	**95.40%**

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 that are unseen in the POS tag dictionary D might still exist in the learned word embeddings from Wikipedia.

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 45-tag set and conduct experiments on the standard 24k test data, following a similar experimental setup as in previous approaches [15, 27, 28]. The details of the reduction of POS Tag are presented in Table 7.

Table 8 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 achieves an accuracy of 95.4%, outperforming most baselines, except IP+EM where our approach is only 1.4% lower.

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 effect of incorporating of such constrain, we conducted experiments on different test sets with and without such constrain. It can be observed from Figure 7 that the performance

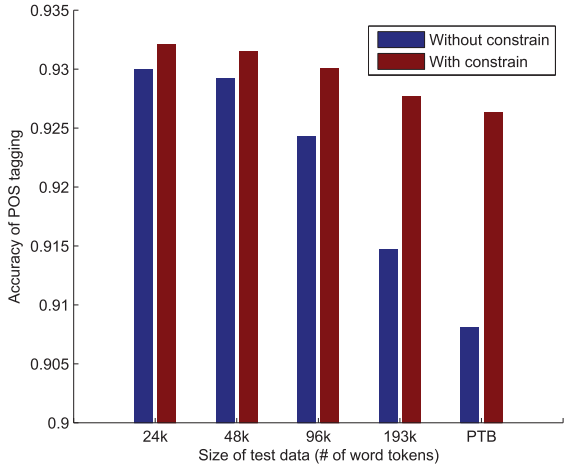

Fig. 7. Performance comparison of the proposed approach with or without the constrain on different test sets.

Table 9. Performance Comparison of the Proposed Approach with or without Using the Word Embedding

Feature	Accuracy
Using Word Embedding Features	**92.63**%
Using Manually Constructed Features	92.45%

of the proposed model with the constrain outperforming the one without the constrain. It further verifies the effectiveness of incorporating such constrain.

4.9 The Impact of Features Used

To find out whether the accuracy gain of the proposed method is due to incorporating the word embedding features, we compare the performance of the proposed approach with or without using the word embedding features. When not using word embedding features, we employ the manually designed features instead, such as POS induction features (e.g., whether containing digit, hyphen) and word alignment features (e.g., prefix, suffix, and stemming), following the same way as Reference [28]. The experimental results are presented in Table 9. The size of U is set to 50k and the whole PTB is used as a test dataset. It can be observed that the proposed approach achieved the similar performance with or without using the word embedding. It further verifies that the performance gain achieved by the POS tagging system might attribute to the proposed constrained ECOC-based approach.

4.10 Experimental Results 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 employed 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.

[2]www.di.unito.it/~tutreeb/CCG-TUT.

Table 10. Comparison of the Performance of the Proposed Framework for Italian and Malagasy POS Tagging

Italian		Malagasy	
Method	Accuracy	Method	Accuracy
EM	83.4%	Reference [16]	80.7%
IP	88.0%	DMLC+EM	81.1%
MIN-GREEDY	88.0%		
Our approach	**90.9%**	Our Approach	**84.5%**

Table 11. The Performance of the Proposed Approach Versus L on Malagasy Dataset

L	15	25	35	45	55	65	75	85	95
Accuracy	77.79%	80.18%	81.95%	81.67%	**84.50%**	80.05%	84.33%	84.69%	85.39%

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 take concatenated 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 used it as the feature vector of the target word.

For the Malagasy language, the dateset used in Reference [16][4] 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 unannotated corpus U contains 20,000 word tokens, among which, 8,674 tokens are from the training set of the dateset in Reference [16] and the others are from Malagasy Wikipedia. The held-out test set contains 1,602 words and 1,683 word/tag pairs (5303 tokens). To generate Malagasy word embeddings, we downloaded the whole Malagasy Wikipedia.[5] Two-hundred ninety-thousand sentences extracted from the corpus were employed for generating 128-dimensional word embeddings using word2vec.[6]

Table 10 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 Malagasy, which are better than all the baselines. It further validates the effectiveness of our proposed approach regardless of the language.

We also conduct an experiment by changing L on Malagasy dataset. Experimental results are presented in Table 11. It can be observed that choosing $10 \log |O| = 55$ ($|O| = 44$, the number of distinct POS tags in Malagasy dataset) almost achieves the best performance.

5 CONCLUSIONS AND FUTURE WORK

In this article, 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 fall into the positive or negative dichotomy specified by the column coding in ECOC. In this way, the

[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.

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

We would like to thank the anonymous reviewers for their valuable comments and suggestions.

REFERENCES

[1] 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.

[2] Erin L. Allwein, Robert E. Schapire, and Yoram Singer. 2000. Reducing multiclass to binary: A unifying approach for margin classifiers. In *Proceedings of the 17th International Conference on Machine Learning (ICML'00)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, 9–16. http://dl.acm.org/citation.cfm?id=645529.658120.

[3] 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.

[4] 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, 582–590. http://dl.acm.org/citation.cfm?id=1857999.1858082.

[5] 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.

[6] 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.

[7] Eric Brill. 1995. Unsupervised learning of disambiguation rules for part of speech tagging. In *Proceedings of the 3rd Workshop on Very Large Corpora*, Vol. 30. Somerset, New Jersey: Association for Computational Linguistics, 1–13.

[8] 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 (Dec. 1992), 467–479. http://dl.acm.org/citation.cfm?id=176313.176316.

[9] 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*.

[10] Chih-Chung Chang and Chih-Jen Lin. 2011. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology* 2, 3, Article 27 (May 2011), 27 pages.

[11] 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, 575–584. http://dl.acm.org/citation.cfm?id=1870658.1870714.

[12] Alexander Clark. 2003. Combining distributional and morphological information for part of speech induction. In *Proceedings of the 10th Conference on European Chapter of the Association for Computational Linguistics (EACL'03)*. 59–66.

[13] 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.

[14] 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.

[15] 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.

[16] Dan Garrette and Jason Baldridge. 2013. Learning a part-of-speech tagger from two hours of annotation. In *HLT-NAACL*. Citeseer, 138–147.

[17] 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.

[18] 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.

[19] 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, 320–327. DOI: http://dx.doi.org/10.3115/1220835.1220876

[20] Mark Johnson. 2007. Why doesn't EM find good HMM POS-taggers? In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL'07), June 28–30, 2007, Prague, Czech Republic*. 296–305.

[21] 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.

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

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

[24] Tahira Naseem, Benjamin Snyder, Jacob Eisenstein, and Regina Barzilay. 2009. Multilingual part-of-speech tagging: Two unsupervised approaches. *Journal of Artificial Intelligence Research* 36, 1 (Sept. 2009), 341–385. http://dl.acm.org/citation.cfm?id=1734953.1734961.

[25] 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.

[26] 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.

[27] Sujith Ravi, Sergei Vassilivitskii, and Vibhor Rastogi. 2014. Parallel algorithms for unsupervised tagging. *Transactions of the Association for Computational Linguistics* 2 (2014), 105–118.

[28] 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.

[29] 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.

[30] 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, 354–362.

[31] Kristina Toutanova, Mark Johnson, et al. 2007. A Bayesian LDA-based model for semi-supervised part-of-speech tagging. *Advances in Neural Information Processing Systems*. 1521–1528.

[32] 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.

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

[34] Meishan Zhang, Yue Zhang, Wanxiang Che, and Ting Liu. 2014. Type-supervised domain adaptation for joint segmentation and POS-tagging. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, Stroudsburg, PA, 588–597.

[35] 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 (EMNLP'09)*. Association for Computational Linguistics, Stroudsburg, PA, 688–697. http://dl.acm.org/citation.cfm?id=1699571.1699602

[36] Deyu Zhou, Liangyu Chen, and Yulan He. 2015. An unsupervised framework of exploring events on twitter: Filtering, extraction and categorization. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI'15)*. 2468–2474.
[37] Deyu Zhou, Dayou Zhong, and Yulan He. 2014. Event trigger identification for biomedical events extraction using domain knowledge. *Bioinformatics* 30, 11 (2014), 1587.
[38] Zhi-Hua Zhou. 2012. *Ensemble Methods: Foundations and Algorithms* (1st ed.). Chapman & Hall/CRC.

Received May 2017; revised March 2018; accepted May 2018