Weakly Supervised POS Tagging without Disambiguation

Deyu Zhou, Haiyang Xu, 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 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. As POS tagging is crucial for further natural language processing (NLP) applications such as named entity recognition and information extraction, weakly supervised POS tagging would benefit various NLP applications in such languages where tagged corpora are mostly unavailable. In this paper, we propose a novel approach 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 binary classifier is learned for each POS tag. For each classifier, its training data is generated in the following way: each 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 in ECOC; and similarly for negative training examples. Therefore, 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 two corpora for English and Italian POS tagging, achieving accuracies of 93.21% and 90.9% individually, which shows a significant improvement compared to the state-of-the-art approaches.

1. INTRODUCTION

Due to the lack of enough annotated corpora, weakly supervised learning has become a hot topic in natural language processing (NLP) domain in recent years. In this paper, weakly supervised part-of-speech (POS) tagging is to learn to predict POS tag for a given word in context given a dictionary of words with their possible POS tags as shown in Table I. As POS tagging is crucial for further NLP applications such as named entity recognition [Zhou et al. 2014] and information extraction [Zhou et al. 2015], weakly supervised POS tagging might benefit NLP in such languages where both tagged corpora and language annotators are mostly unavailable.

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 latent variable which 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 [1994] employed maximum likelihood estimation (MLE) to train a trigram HMM. Following this way, some improvements were made by modifying the statistical model or employing better parameter estimation techniques. For example, Banko and Moore [2004] modified the basic HMM structure to employ context on both sides of the word to be tagged. Training is conducted on CRF using contrastive estimation for POS tagging [Smith and Eisner 2005].

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	
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 .	You/PRP have/VBP to/TO recognize/VB that/IN these/DT events/NNS took/VBD place/NN 35/CD years/NNS ago/IN ./.

Clustering can also be employed for weakly supervised POS tagging by casting the identification of syntactic classes as a knowledge-free clustering problem. Distributional clustering and dimensionality reduction techniques are typically applied [Toutanova and Johnson 2008]. Unfortunately, due to a lack of standard and informative evaluation techniques, it is difficult to compare the effectiveness of different clustering methods.

Most of the approaches mentioned above are disambiguation based. 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 by 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 paper, we incorporate a novel strategy for weakly supervised POS tagging. It does not rely on disambiguating possible POS tags. In specific, error-correcting output codes (ECOC) [Dietterich and Bakiri 1995], one of the famous multi-class learning techniques, is adapted and a binary classifier is learned for each POS tag. The key adaptation lies in how the binary classifiers corresponding to the ECOC coding matrix are built. For each column of the binary coding matrix, one binary classifier is built based on binary training examples derived from the POS tag dictionary. Specifically, any word will be regarded 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 this way, the set of possible tags is treated as an entirety without resorting to the 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, these feature design needs domain knowledge and expertise. In this paper, features employed for weakly supervised POS tagging are generated without manual intervene.

The main contributions of the paper are summarized below:

— We proposed a novel approach based on ECOC for weakly supervised POS tagging. In this way, the set of 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.
— We evaluated the proposed approach on two corpora for English and Italian 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 Part-of-Speech tagging. The best taggers can obtain tagging accuracies over 97% on the English Penn Treebank. However, tagging without labeled data still is full of challenge. Recently, more and more researchers pay attention to POS tagging without using the fully annotated corpora. There are mainly two directions to handle this problem.

On one hand, some efforts have been made on unsupervised POS tagging using the clustering techniques. Omri *et al.* [2010] first identify landmark clusters of words, then utilize morphological and distributional representations computed in a fully unsupervised manner. Kairit *et al.* [2014] 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. However, evaluating these systems proves difficult due to the lack of mapping from cluster labels to gold standard tags.

On the other hand, many researchers focused on weakly supervised POS tagging using tag dictionaries [Merialdo 1994], where unlabeled word sequences and a dictionary of possible tags for each word are given. A Bayesian approach that integrates over all possible parameter values is employed and has the standard structure of 3-gram HMM [Goldwater and Griffiths 2007]. Integer programming (IP) is employed to search the smallest bi-gram POS tag set and uses this set to constrain the training of EM [Ravi and Knight 2009]. It achieves an accuracy of 91.6% on the 24k test set, but can not handle large dataset. For solving the deficiency of IP, a two-stage greedy minimization approach is proposed in [Ravi et al. 2010] that runs much faster while maintaining the performance of tagging. To further improve the performance, several heuristics are employed in [Garrette and Baldridge 2012]. Moreover, it works on incomplete dictionary and achieves an accuracy of 88.52%. In [Ravi et al. 2014], 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 [Yatbaz and Yuret 2010], 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.

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 tags set. In this paper, we propose a novel approach for weakly supervised POS tagging without disambiguation. The set of possible tags is treated as an entirety without disambiguation. Moreover, instead of manually 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 two corpora for English and Italian 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 [Zhang 2014] in which each training example 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. This equally applies to both the training and testing instances. 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.

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

Table II lists notations used in this paper. 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. 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 coding-decoding procedure. 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 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 Error Correcting Output Codes (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. To begin, one assigns a unique L-bit vector to each label. One can view the ith bitvector as a unique coding for label i. The set of bitvectors is referred as decomposition matrix (coding matrix) and denoted as M with value $\{1, -1\}$. Then, the ECOC method can be separated into two steps: encoding and decoding. The purpose of the encoding step is to design M. Each row of the coding matrix M is called codeword to represent each class and each column of the coding matrix M specifies a dichotomy over the label space to learn a binary classifier. 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 sample is generated by concatenating the predictive outputs of the L binary classifiers. The sample is predicted 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 that are semantically similar to each other are placed in nearby locations in the embedding space. This characteristic is precisely what we want, because the key

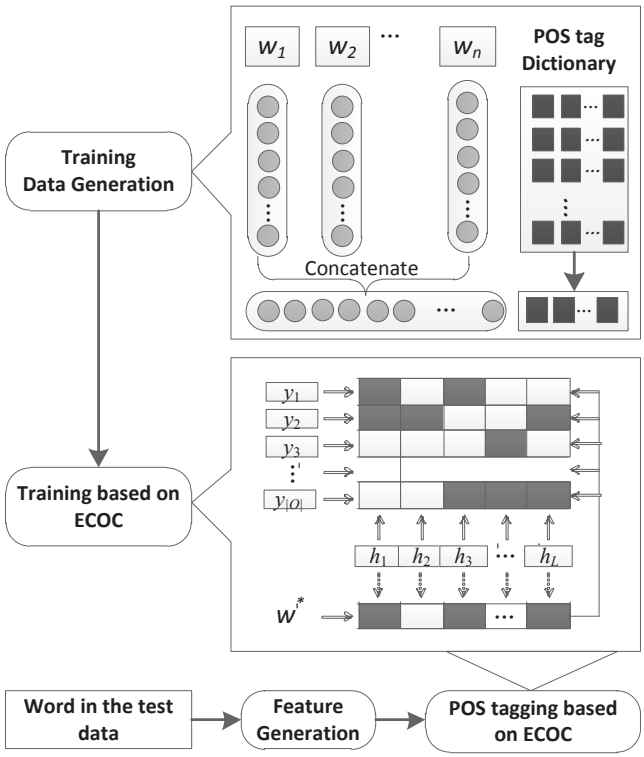

Fig. 1. The proposed approach for weakly supervised POS tagging.

to one-class classification is semantic similarity measurement. As such, word embeddings could be explored for one-class classification.

We use neural language modeling [Collobert et al. 2011] to learn word representations by discriminating the legitimate phrase from incorrect phrases. Given a sequence of words $p = (w_1, w_2, ..., w_d)$ 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 function 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 window of words obtained by replacing the central word of p by the word r and $f_\theta(p)$ is the 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

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

3.3. Training and Testing based on Constrained ECOC

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

Firstly, in the coding 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 class y_j (See the lower half of Figure 1). Each column of the coding matrix $M(:,l)$ specifies a dichotomy over the tag space y with $y_l^+ = \{y_j|M(j,l) = +1, 1 \leq j \leq |O|\}$ and $y_l^- = \{y_j|M(j,l) = -1, 1 \leq j \leq |O|\}$. 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.

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

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

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 the functions and their corresponding definitions employed in our approach.

Table III. The definition of Different decodings.

Decoding	Definition
Euclidean	$\sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$
Attenuated Euclidean	$\sqrt{\sum_{i=1}^{n}\|y_i\|x_i\|(x_i - y_i)^2}$
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	$(\alpha_i + 1)/(\alpha_i + \beta_i + \|O\|)$, 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, the equation 2 is modified as

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

Algorithm 2 Training and Testing based on constrained ECOC.

Inputs: L, $\mathfrak{B}$, thr, 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 $v = [v_1, v_2, \cdots, v_{|O|}]^{\mathrm{T}}$;
  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 $|T_v| \geq thr$ **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.

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 which often runs in an iterative 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.

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 4.1 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.

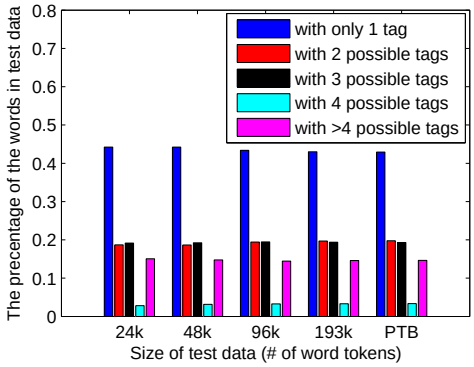

Fig. 2. Distribution of words with different number of possible tags on 24k test set.

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 [Goldberg et al. 2008].

As mentioned before, word embeddings are trained using neural language models [Collobert et al. 2011]. The training of 50-dimensional word embeddings from Wikipedia took about 7 weeks. 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 [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 thr is set to $\frac{1}{10}|U|$.

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 [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.
(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%	
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 an accuracy of 93.21% on the 24k data, which is the best performance reported on the dataset to the best of our knowledge. With the increasing size of the test data, the performance of our approach decreases slightly. According to our analysis, the size of train set is limited, so the train procedure does not cover all words of dictionary, which leads to the performance of larger test set little worse than smaller test set. 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 3, 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 attribute to the insufficient training data for the words with 4 possible tags as shown in Figure 4.1.

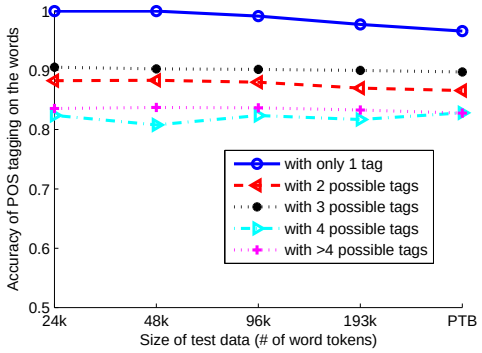

Fig. 3. Accuracy of words with different number of possible tags on 24k test set.

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 on different sizes of test set with 50k train set. The performance of POS tagging with different distance measures are presented in Figure 4 while the definitions of different decodings are presented in Table III.

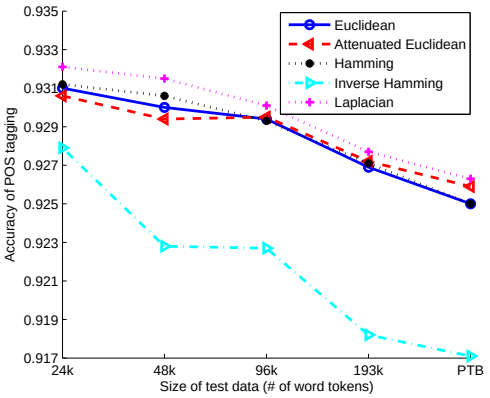

Fig. 4. Performance comparison of unsupervised POS tagging using different decodings on different test sets.

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 V that in general, on a larger test set, the better performance is given by a larger unannotated training set.

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 used a reduced tag set of 17 tags instead of the 45-tag set and evaluated on the standard 24k test data, following a similar experimental setup as in previous approaches. 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	92.90%
Our approach	**95.40%**

4.8. The Impact of Constrained ECOC

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

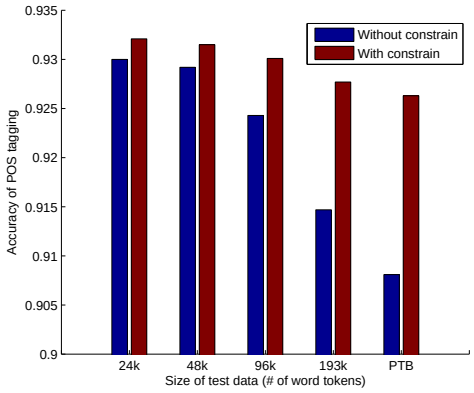

Fig. 5. Performance comparison of the proposed approach with or without constrain on different test sets.

4.9. Evaluation Results of Italian POS Tagging

The proposed model is independent of specific language. To demonstrate this, we use the CCG-TUT corpus [1] 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. To generate Italian word embeddings, we trained word2vec [2] from 14 million sentences extracted from the Italian Wikipedia with the window size set to 11 and got 64-dimensional word embeddings. 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.

Table IX shows the results of Italian POS tagging. It can be observed that our proposed approach achieves an accuracy of 90.9%, which is better than all the baselines. It further validates the effectiveness of our proposed approach in a language other than English.

Table IX. Comparison of the performance of the proposed framework on the CCG-TUT corpus for Italian POS tagging.

Method	Accuracy
EM	83.4%
IP	88.0%
MIN-GREEDY	88.0%
Our approach	**90.9%**

To investigate the impact of decoding, the performance of Italian POS Tagging with different distance measures are presented in Table X. It can be observed that Inverse Hamming achieves the best results.

Table X. Performance comparison of weakly supervised Italian POS tagging using different decodings on the test set.

Decoding	Accuracy
Inverse Hamming	**90.9%**
Euclidean	90.0%
Attenuated Euclidean	90.0%
Hamming	90.0%
Laplacian	89.9%

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,

[1] www.di.unito.it/~tutreeb/CCG-TUT
[2] code.google.com/p/word2vec/

features employed for POS tagging are generated without manual intervention. We have evaluated the proposed approach on the Penn Treebank and CCG-TUT corpus for English and Italian POS tagging, and observed a significant improvement in accuracy compared to the state-of-the-art approaches. In future, 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), the Innovate UK under the grant number 101779 and the Collaborative Innovation Center 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.

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.

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.

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.

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 *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*. Association for Computational Linguistics, Prague, Czech Republic, 744–751.

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

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 and Mark Johnson. 2008. A Bayesian LDA-based model for semi-supervised part-of-speech tagging. In *Advances in Neural Information Processing Systems 20*, J.C. Platt, D. Koller, Y. Singer, and S.T. Roweis (Eds.). Curran Associates, Inc., 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.

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.

Deyu Zhou, Dayou Zhong, and Yulan He. 2014. Event trigger identification for biomedical events extraction using domain knowledge. *Bioinformatics* 30, 11 (2014), 1587–1594. DOI:http://dx.doi.org/10.1093/bioinformatics/btu061

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