Cross-lingual part-of-speech tagging using word embedding

Wei Yuan[1], Lei Wang[2], Xiao-Fei Sun[3], Wen-Wen Pan[4], Jia-Guo Lv[5]
[1, 2, 3, 4, 5]Zaozhuang University, Zaozhuang, Shandong, 277160, China
[2]China University of Mining and Technology, Xuzhou, Jiangsu, 221116, China
[1, 2]Corresponding author
E-mail: [1]*pcy8866@126.com*, [2]*look_00@163.com*, [3]*34740935@qq.com*, [4]*509974840@qq.com*, [5]*lvjiaguo2004@163.com*

(Received 11 August 2016; accepted 14 August 2016)

Abstract. As one of semi-supervised learning approach, cross-lingual projection leverages existing resources from a resource-rich language when building tools for resource-poor languages. In this paper we attempt to make use of word embedding with anchor based label propagation to improve the accuracy of a cross-lingual projection task: cross-lingual part-of-speech tagging under the graph-based framework. Our approach uses bilingual parallel corpora and labeled data from the resource-rich side assuming that there is no labeled data or only a few labeled data in resource-poor language. The results suggest the efficacy of our approach over traditional label propagation with lexical feature for projecting part-of-speech information across languages, and show that a few of labeled data help to enhance the effect a lot in cross-lingual task.

Keywords: part-of-speech (POS), statistical machine translation (SMT), support vector machine (SVM).

1. Introduction

Supervised learning has become a mainstream in the statistical learning domain, advanced in many applications. Supervised learning approaches rely on a large volume of labeled training data to build accurate model while obtaining labeled data usually requires a lot of labor and time. In many tasks of natural language processing, labeled data is scarce but unlabeled data is easy to obtain. There are only a few languages having a large number of labeled corpora due to different research effort and commitment. We consider cross-lingual projection as a practically motivated scenario, in which we want to leverage existing corpora from a resource-rich language (like English) when building annotations for resource-poor languages by transforming the annotations of sentences in one language into another one. There is an assumption that absolutely no labeled or only a few labeled training data is available for some languages of interest, and parallel data with a resource-rich language (like Chinese minority languages and Chinese) accessible. The scenarios for a large set of languages have been considered by a number of authors in the past [1-4] study related but different multilingual grammar and tag induction tasks, where it is assumed that no labeled data at all is available.

Our work focuses on one task of cross-lingual projection: part-of-speech (POS) tagging. In this paper we use graph based cross-lingual part-of-speech framework [4] and try to improve the accuracy of tagging under this framework in two ways. To make the projection practical, we rely on the twelve universal POS tags of Petrov et al. [5] (see Table 1). Syntactic universals is a well-studied concept in linguistics [6, 7], and was recently used in similar form by Naseem et al. [8] in multilingual grammar induction.

There are two main contributions of this paper: first, anchor graph based method is used to solve label propagation which achieved linearly time and space complexity compared to classical approach. Therefore, we can leverage more context information to build graph than ever; second, distributed representation (word embedding) as the features of context information is used to solve the data sparse problem compared to traditional one-hot representation. Besides, a graph with extra knowledge is built through training word embedding with external data.

2. Overview of the approach

The workflow of our approach is represented as Algorithm 1. Our goal is to build POS tag corpus for resource-poor language, assuming that we have POS taggers for resource-rich language and some parallel text between the two languages.

Algorithm 1. Bilingual POS induction based on anchor graph:

Input: Parallel corpora (t_i, s_i), $i = 1, \ldots, n$; unlabeled target language training data;

Output: POS tag $\mathrm{pos}(W_i)$ of target language T;

1) transfer specific pos tags to universal tags;

2) $A = \{a_{ij}\}$: word alignment with GIZA++ in two directions;

3) Filter the results of word alignments;

4) Compute target language POS tag distribution according to alignments;

5) Label the target word tag with the maximum probability: $\mathrm{pos}(W_i) = \mathrm{argmax}_y p_i(y)$.

For OOV and unaligned words there is no tag:

6) Construct graph;

7) Graph propagation with AGR algorithms.

On step 1 in the frame, we transfer the specific POS tags of the two languages into twelve universal tags. The twelve universal tags are listed in Table 1.

Table 1. Universal POS tags

Universal tags	Name	Universal tags	Name	Universal tags	Name
Noun	Noun	Det	Determiner	Prt	Particle
Verb	Verb	Adp	Preposition	.	Punctuation
Adj	Adjective	Num	Quantifier	X	Others
Pron	Pronoun	Conj	Conjunction	Adv	Adverb

Suppose $S = \{S_1, S_2, \ldots S_l\}$ is source language sentence and $T = \{T_1, T_2, \ldots T_J\}$ is target language sentence, $A = \{a_{ij}\}$ is the word alignment between these two languages. Define the sentence alignment confidence $C(A|S,T)$ by the geometric mean of bidirectional word alignment posterior probability as follows:

$$C(A|S,T) = \sqrt{P_{s2t}(A|S,T)P_{t2s}(A|T,S)}, \quad P_{s2t}(A|S,T) = \frac{P(A,T|S)}{\sum_{A'} P(A',T|S)}. \tag{1}$$

$P_{s2t}(A|S,T)$ represents alignment probability given the source and target sentence and the direction is from the source to the target. $P_{t2s}(A|T,S)$ is on the contrary. The numerator is the probability of generating the alignment and the target sentence given the source sentence, as follows:

$$P(A,T|S) = \prod_{j=1}^{J} p(t_j|s_i, a_{ij} \in A). \tag{2}$$

The higher confidence is; the higher accuracy of alignments gets. We set the threshold 0.9 to filter sentences whose confidence is lower than the threshold.

After filtering alignments, the following steps can project the source language tags to target language tags through alignment results without any labeled data. First, tag the resource-rich side of the parallel text using supervised model; and then transfer the tags to the target side with the maximum probability of alignments. The POS distribution is computed as follows (step 4):

$$p_y(x) = \frac{\sum_{v_y} \#[u_i \leftrightarrow v_y]}{\sum_{y'} \sum_{v_{y'}} \#[u_i \leftrightarrow v_{y'}]}, \tag{3}$$

where $\#[u_i \leftrightarrow v_y]$ represents the count of target word u_i aligned to source word v whose POS tag is y.

3. Graph construction

In our graph-based learning approach we construct a graph whose vertices are labeled and unlabeled examples, and whose weighted edges encode the similarity degree of the examples they linked. An example of English graph is shown in Fig. 1. As we see, all the vertices are connected each other by weighted edges. The weight of each edge represents the similarity between the vertices. "[my]", "[I]", "[her]", "[his]" are all labeled as PRON while the label of "[their]" is unknown. Label propagation is used to propagate these tags inwards and results in the tag distributions for the unlabeled vertices. That is the purpose of constructing the graph.

Generally, POS tagging can be viewed as a sequence labeling problem while it can also be treated as a classification problem in this paper. Assuming there are N symbols for POS in a certain language and each word of a sentence is tagged as one symbol, then POS tagging is a multi-class classification problem. There are many multi-class machine learning approaches. We also use supervised support vector machine (SVM) in the comparison experiment

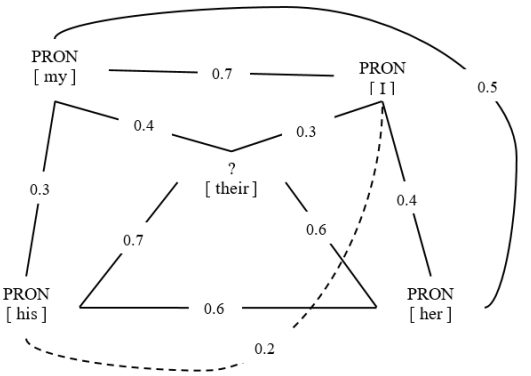

Fig. 1. An example of graph construction

3.1. Features for graph

In machine learning, each example is denoted by a set of features, the definition of feature template directly determines how much information to learn. Each word is viewed as an example whose attributes are described by N-gram of context information. Assuming context window of the sample is s, then the unigram context of word w can be represented as $\{w_{-s}, \ldots, w_0, \ldots, w_s\}$, the bigram of it corresponds to $\{w_{-s}w_{-s+1}, \ldots, w_{-1}w_0, w_0w_1, \ldots, w_{s-1}w_s\}$, the corresponding trigram is $\{w_{-s}w_{-s+1}w_{-s+2}, \ldots, w_{-1}w_0w_1, \ldots, w_{s-2}\, w_{s-1}w_s\}$ and so on. If w_0 denotes central word, then w_s denotes the word with distance s from right, and w_{-s} from left. Let us set context window $s = 2$, various features used are shown in Table 2.

Table 2. Various features used to represent a word

N gram	Feature
Unigram	$w_{-2}, w_{-1}, w_0, w_1, w_2$
Bigram	$w_{-2}w_{-1}, w_{-1}w_0, w_0w_1, w_1w_2$
Trigram	$w_{-2}w_{-1}w_0, w_{-1}w_0w_1, w_0w_1w_2$

3.2. Lexical feature

Lexical feature uses sparse vector to denote a word. At first three lexicons are built by

extracting from the corpus for unigram, bigram and trigram respectively. The order of words in these vocabularies is independent. Assuming the sizes of lexicons are $|V_1|$, $|V_2|$, $|V_3|$ for unigram, bigram and trigram respectively, total size is $5|V_1| + 4|V_2| + 3|V_3|$ if using all the features in Table 2. As lexical feature is sparse, we use sparse vector to denote it whose index is the order in the lexicon and value is 0/1 representing whether it is in the context of the word. Generally, ignore the terms with feature value 0.

3.3. Word embedding feature

Word embedding uses distributed representation. The feature of word embedding does not rely on the lexicon. The size of features is related to feature category and dimensions of word embedding itself. Now word embedding can be trained by many deep learning algorithms using task corpus or external corpora [10, 11]. The more external corpora we use, the more knowledge we have whose latent semantic and syntactic information can help improve task effect.

Currently, training algorithm for word embedding is relatively mature while the approaches for training phrase embedding have not been widely recognized. Therefore, we use combination of word embedding (unigram) as the feature representation of bigram and trigram context feature. Assuming the word embedding of word w_i is $w_i(e)$, dimension size of word embedding is m, then the embedding of bigram $w_i w_{i+1}$ is $w_i(e){:}\, w_{i+1}(e)$, colon means "joint", dimension size of bigram embedding is 2 m. It is in the similar fashion for trigram. This way can also keep the position information.

4. Anchor graph label prediction

Label propagation is used to transfer the labels to the adjacent untagged vertices first, and then to all of the untagged vertices. Labels are propagated out according to the degree of similarity between the two samples. In this procedure the labeled data will not change while the labels of unlabeled data update.

As traditional label propagation consumes much time and space, the context information is limited when building a graph. To make use more context information and improve the efficiency we adopt anchor based label propagation algorithm [12]. The anchor based label propagation (Algorithm 2) includes 3 stages: 1) K-means clustering. The time complexity of K-means algorithm is $O(mn)$; 2) Compute the mapping matrix Z (data to anchor) whose time complexity is $O(kmn)$; 3) Compute the soft label matrix A for anchors whose time complexity is $O(m^3 + m^2 n)$. K is small number for nearest points, m the number of anchors which is a fixed, n the number of all points, and $k \ll m \ll n$. Therefore, the total time complexity of anchor based label prediction is $O(m^2 n)$.

Compared to traditional label propagation, the algorithm of anchor based label propagation reduces the time complexity from $O(kn^2)$ to $O(m^2 n)$, $m \ll n$. We will use traditional label propagation as a baseline to compare their experimental results.

Algorithm 2. Anchor based label propagation:

Input: Dataset D (labeled data L and unlabeled data U, label set C;

Output: labels for unlabeled data.

1) Select m cluster centers using K-means algorithm as anchor points

2) Compute the mapping matrix Z (data x to anchor u_i) as follows:

$$\boldsymbol{z}(x) = \frac{\left[\delta_1 \exp\left(-\frac{\mathcal{D}^2(x,u_1)}{t}\right), \ldots, \delta_m \exp\left(-\frac{\mathcal{D}^2(x,u_m)}{t}\right)\right]^T}{\sum_{j=1}^{m} \delta_j \exp\left(-\frac{\mathcal{D}^2(x,u_j)}{t}\right)},$$

where $\delta \in \{0, 1\}$, $\mathcal{D}$ is distance function defined by the user.

3) Compute the soft label matrix A for anchors as follows $A^* = (Z_l{}^T Z_l + \gamma\tilde{L})^{-1} Z_l{}^T Y$, $\tilde{L} = Z^T Z - (Z^T Z)\Lambda^{-1}(Z^T Z)$.

4) Label the unlabeled data as following:

$$\hat{y}_i = \arg\max_{j\in\{1,\ldots,c\}} \frac{Z_i a_j}{\lambda_j}, \quad i = l+1, \ldots, n.$$

In Algorithm 2, the formula of Step 2 gives the value of each element in the matrix Z, where δ_i is a weight and t is an adjusting parameter. Step 3 provides a global optimal calculation of A, where $\tilde{L}$ is a Laplacian matrix, Y is a class indicator matrix and $\gamma > 0$ as a regularization parameter. And in Step 4, $\hat{y}_i$ will indicate the class for unlabeled data x_i, where λ_j is a normalization parameter.

5. Experiment

5.1. Comparison for feature setting

At first we compare the effect of the lexical feature and the word embedding feature in two ways on small data set. The data comes from Chinese Treebank (CTB7) of which the first 1000 sentences are labeled data, sentences from 19526 to 21435 are test set and the others are unlabeled data. We evaluate it by POS tag accuracy. We train our word embedding that has 64 dimensions on self-mined Chinese corpora using RNNLM. When extracting lexical features, the low frequency words in the vocabularies are filtered. We conduct supervised and semi-supervised experiments. The supervised algorithm is the support vector machine (SVM) using Libsvm toolkit with RBF kernel function and the default setting. Semi-supervised algorithm is the anchor graph label propagation (AGLP) setting with anchor number 1000 and kNN number 3. We use the cluster centers obtained from sofia-kmeans toolkit as anchors.

Table 3. POS tagging accuracies for different features

	Lexical	Add	Combine
SVM	61.25	68.704	69.457
AGLP	56.32	64.44	66.39

Table 3 shows the results of the experiments. As expected the word embedding feature performs much better whether in supervised setting or in semi-supervised setting. There are three reasons: 1) the lexical feature is sparse which harms machine learning models while the word embedding feature is dense and smooth which will be appropriate for computing similarity in graph-based model; 2) word embedding benefits from the extra knowledge in external corpora and is more accurate; 3) when extracting the lexical feature low frequency words are filtered and information losed while the word embedding feature keep all the information and reduce dimensionality naturally.

5.2. POS tagging experiment

The parallel data for English to Chinese projection comes from LDC2003E14 (FBIS, 239335 pair sentences). First the English side of the parallel data is tagged by Stanford POS tagger that has the stat-of-art performance, and then the language-specific tags is transferred to the universal tags for evaluation. We use the alignments produced by GIZA++ [9]. Test set includes sentences from 19526 to 21435 in Chinese Penn Treebank (CTB7). We use the first 500 sentences in CTB7 as a few labeled data. To involve additional knowledge, we train our word embedding which has 64 dimensions on Chinese Wiki data using RNNLM.

The data for Chinese to Tibetan projection comes from CWMT2013 (109381 pair sentences)

whose first 100000 pair sentences in training set are parallel corpus and the develop set is test set. We use the last 500 sentences in the training set of CWMT2013 as a few labeled data. The Chinese side of the parallel data is tagged and the result transferred into universal tags in the same way. And then we run the alignment procedure and filter it as above. Tibetan word embedding is trained on all Tibetan sentences in CWMT2013 using RNNLM too.

We take traditional label propagation based approach implemented in-lab as the baseline. The baseline algorithm constructs graph with lexical feature (only unigram) which extracting top frequency words as feature word and reducing dimension to 75 using Singular Value Decomposition (SVD). The kNN number sets 5 and context window is 2.

There is directly projecting POS from source language to target language and for untagged words labeling the most frequency tags in source side. We denote it by DP(U). The direct projections are used to initialize the graph and anchor graph label propagation run on the graph. We use word embedding organized by "combine" way as feature for graph construction. We denote it by AGLP(U). In addition to the direct projection result we also use a few labeled data to initialize the graph. We denote it by AGLP(L).

Table 4. POS tagging accuracies for various algorithms and tasks, as well as our approach

	En-Chs	Chs-Ti
Baseline	62.57	57.28
DP(U)	61.33	56.94
AGLP(U)	63.66	58.14
AGLP(L)	82.59	69.37

Table 4 shows complete results of different algorithms in two directions. We get similar results in the two tasks. In no labeled data setting our approach (AGLP) gives a little of improvement upon baseline which proves the advantage of our method. There are three reasons: first, we use more context information to construct the graph that making the vertices representation more accurate while anchor based label propagation reduce the time and space complexity significantly; second, the baseline filters the low frequency feature words which losing part of context information but word embedding feature keeps all context information and includes more knowledge from external data; finally, our approach solves the data sparse problem by using word embedding feature, therefore similarity computing is more accurate. Meanwhile we found that the baseline approach does not give a large improvement upon direct projection DP(U) method. It is reasonable that label propagation trusts the directly projected results and do not change the initial labels of labeled data. Therefore, the accuracy of the direct projection limited the promotion space. It is worthy note that when add a few of labeled data the accuracy improves 19.93 % and 11.23 % respectively in two language pairs. Such a result indicates a few of labeled data have a significant impact on the task and help to enhance the effect a lot.

6. Conclusions

In this paper we utilized word embedding with anchor based label propagation to improve the accuracy of cross-lingual part-of-speech tagging under the graph based framework. Since we are interested in applying our approach to resource-poor languages whose labeled data is scarce, we conduct our approach into two settings: no labeled data and only a few labeled data. Our results outperform the baseline method that based on traditional label propagation with lexical feature, and the results also indicate the word embedding feature is suitable for graph based model. The results have shown the efficacy of our approach for projecting POS information across languages. Besides, our results suggest that a few of labeled data help enhance the effect a lot in the cross-lingual task.

Acknowledgement

This work was funded by the Research on the Problem of Maximizing the Impact of Competition Environment in Social Networks (No. J15LN81).

References

[1] **Han W. L., Li Z. G., Yuan L., Xu W. Y.** Regional patterns and vulnerability analysis of Chinese web passwords. IEEE Transactions on Information Forensics and Security, Vol. 11, Issue 2, 2016, p. 258-272.

[2] **Koller O., Forster J., Ney H.** Continuous sign language recognition: towards large vocabulary statistical recognition systems handling multiple signers. Computer Vision and Image Understanding, Vol. 141, 2015, p. 108-125.

[3] **Stoilos G., Venetis T., Stamou G.** A Fuzzy extension to the OWL 2 RL ontology language. Computer Journal, Vol. 58, Issue 11, 2015, p. 2956-2971.

[4] **Ben Mohamed M. A., Mallat S., Nandi M. A., Zrigui M.** Exploring the potential of schemes in building NLP tools for Arabic language. International Arab Journal of Information Technology, Vol. 12, Issue 6, 2015, p. 566-573.

[5] **Das D., Petrov S.** Unsupervised part-of-speech tagging with bilingual graph-based projections. Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies – Volume 1. Association for Computational Linguistics, 2011, p. 600-609.

[6] **Petrov Slav, Das Dipanjan, Mcdonald Ryan** A universal part-of-speech tagset. ArXiv:1104.2086, 2011.

[7] **Carnie Andrew** Syntax: A Generative Introduction (Introducing Linguistics). Blackwell Publishing, 2002.

[8] **Newmeyer Frederick J.** Possible and Probable Languages: A Generative Perspective on Linguistic Typology. Oxford University Press, 2005.

[9] **Naseem Tahira, Chen Harr, Barzilay, Regina, Johnson Mark** Using universal linguistic knowledge to guide grammar induction. Proceedings of EMNLP, 2010.

[10] **Och F. J., Ney H.** Giza++: Training of Statistical Translation Models. 2000.

[11] **Bengio Yoshua, Ducharme Rejean, Vincent Pascal, Jauvin Christian** A neural probabilistic language model. Journal of Machine Learning Research (JMLR), Vol. 3, 2003, p. 1137-1155.

[12] **Phi T. P., et al.** Naming persons in news video with label propagation. IEEE International Conference on Multimedia and Expo, 2010, p. 1528-1533.