Augmented Parsing of Unknown Word by Graph-based Semi-supervised Learning

Qiuping Huang Derek F. Wong Lidia S. Chao Xiaodong Zeng Liangye He
NLP^{2}CT Laboratory / Department of Computer and Information Science University of Macau
Macau S.A.R., China
michellehuang718@gmail.com,{derekfw,lidiasc}@umac.mo,
nlp2ct.samuel@gmail.com,wutianshui0515@gmail.com

Abstract

This paper presents a novel method using graph-based semi-supervised learning (SSL) to improve the syntax parsing of unknown words. Different from conventional approaches that uses hand-crafted rules, rich morphological features, or a character-based model to handle unknown words, this method is based on a graph-based label propagation technique. It gives greater improvement on grammars trained on a smaller amount of labeled data and a large amount of unlabeled one. A transductiv[1] graph-based SSL method is employed to propagate POS and derive the emission distributions from labeled data to unlabeled one. The derived distributions are incorporated into the parsing process. The proposed method effectively augments the original supervised parsing model by contributing 2.28% and 1.72% absolute improvement on the accuracy of POS tagging and syntax parsing for Penn Chinese Treebank respectively.

1 Introduction

Parsing is an important and fundamental task in natural language processing. In the past years, many researches focusing on building high quality parsers for English (Charniak, 2000; Collins, 2003; Charniak and Johnson, 2005; Petrov et al., 2006) and these parsers obtain the state-of-the-art performance up to 92% accuracy. Recently, Chinese parsing has received more and more attention, and several researchers attempt to develop accurate parsers for Chinese (Klein and Manning, 2003; Charniak and Johnson, 2005; Petrov and Klein, 2007). Inspired from their works, Huang et al., (2012) design a head propagation table to improve the parsing performance with a factored model. Nevertheless, as pointed out in (Harper and Huang, 2009), the improved performance around 84% F-measure that still falls far short of performance on English. This leaves a large space for the further improvement of Chinese parsing.

As far as we known, there is a large portion of fixed errors stemming from unknown words in Chinese parsing. Therefore, a robust parser must have a mechanism of processing unknown words, where it discovers the POS tag and features information about unknown words during parsing. A number of researches design hand-crafted rules or make use of rich morphological features to handle them. It is well known that Chinese words tend to have greater POS tag ambiguities than English and the morphological properties of Chinese words are complicated to be predicted of POS type for unknown words. For this reason, Harper and Huang (2007) present a character-based model to handle Chinese unknown words. Similar to their work, He et al., (2012) propose a more effective method. They mainly use an exponential function to represent the distance between the head character and other characters in an unknown word and use the geometric average to estimate the emission probability of it. However, in this paper, we focus on using a graph-based label

[1]Transductive learning is used to contrast inductive learning. A learner is transductive if it only works on the labeled and unlabeled training data, and cannot handle unseen data.

27th Pacific Asia Conference on Language, Information, and Computation pages 474－482

propagation method to deal with unknown words. Graph-based label propagation methods have made a remarkable improvement in several natural language processing tasks, e.g. knowledge acquisition (Talukar et al., 2008), Chinese word segmentation and POS tagging (Zeng et al., 2013) and etc. As far as we known, this study is the first attempt at applying graph-based label propagation to resolve the problem of unknown word, which is mainly used to propagate POS tag and derive the emission probabilities to the large amount of unlabeled data by utilizing the limited resource (e.g. POS information from the labeled data, i.e. Penn Chinese Treebank and lexical emission probability learned by the PCFG-LA model). Then the derived unlabeled information generated by graph-based knowledge will be incorporated into the parser. In fact, this method explores a new way to exploit the use of unlabeled data to strengthen the supervised model in parsing.

This paper is structured as follows. Section 2 reviews the background, including the lexical model in the Berkeley PCFG-LA model and the graph-based label propagation methods. Section 3 presents the details of our proposed model based on graph-based semi-supervised learning approach and compares with other unknown word recognition models. Experiments setup and result analysis are reported in section 4. The last section draws the conclusion and future work.

2 Background

2.1 Lexical Model in Berkeley Parser

The Berkeley parser (Petrov et al., 2006; Petrov and Klein, 2007) is an efficient and effective parser that introduces latent annotations to learn high accurate context-free grammars (CFG) directly from a Treebank. Nevertheless, the lexical model of grammar is not well designed to effectively handle the out-of-vocabulary (OOV) words (aka unknown words) universally and the OOV model of Berkeley parser has proved to be more suitable for English in (Huang and Harper, 2009; Attia et al., 2010). The built-in treatment to unseen words of Berkeley parser can be concluded as: utilizing the estimation of rare words[2] to reflect the appearance likelihood of OOV words.

In order to get the more refine and accurate grammar, Petrov et al., (2006) developed a simple split-merge-smooth training procedure. In order to counteract over-fitting problem, they introduced a linear smoothing method to smooth the lexical emission probabilities:

$$\overline{P} = \frac{1}{|t|}\sum_x P_\theta(w|t_x) \quad (1)$$

$$P_\theta(w|t_x) \leftarrow \varepsilon\overline{P} + (1-\varepsilon)P_\theta(w|t_x) \quad (2)$$

where $|t|$ denotes the number of latent tags from t and t_x means a set of latent subcategories $\{t_x|x = 1, \dots, |t|\}$. In Equation (1), θ is the model parameters which can be optimized by EM-algorithm. In Equation (2), ε is a smoothing parameter.

Since the lexical model can only generate words observed in the training data, a separate module is needed to handle the OOV words that appear in the test sentences. There are two ways to estimate an OOV word *w* based on a specific latent tag t_x. One is assigning the probability of generating rare words in the training data by t_x: $P_\theta(rare|t_x)$; another is, suggested by the Berkeley parser as *Sophisticated Lexicon*, to calculate the emission probability through analysing the morphological features of the OOV words. In the Berkeley parser, English words are classified into a set of signatures based on the presence of characters, especially on a list of inherent suffixes (e.g., *-ed, -ing*), then the estimation of w/t_xpair is:

$$P_\theta(w|t_x) \propto P_\theta(s|t_x) \quad (3)$$

where s is the OOV signature for w and $P_\theta(s|t_x)$ is computed by $e_{t_x,s}/e_{t_x}$.

Nevertheless, the features applied to Chinese word are simpler than English. Only the last character of word will be taken into account in estimating emission probabilities of rare word. Before applying such model, OOV words will be checked if they belong to temporal noun (NT)[3], cardinal number (CD)[4], ordinal number (OD)[5] or proper noun (NR)[6] preferentially.

2.2 Graph-based Label Propagation

Graph-based label propagation, a critical subclass of semi-supervised learning (SSL), has

[2]In the newest version of Berkeley parser, words with frequent less than 10 will be regarded as rare words acquiescently.

[3]By checking if the word contains characters like "年" (year), "月" (month), or "日""号"(day).

[4]By checking if the word contains character of number.

[5]By checking if the word contains character, such as "第".

[6]By checking if the word contains character, such as " • "

Algorithm 1: Words Label Propagation Algorithm

Input:

- $_l = \{w_i\}_{i=1}^{l}$: labeled texts
- $_u = \{w_i\}_{i=l+1}^{l+u}$: unlabeled texts
- $E_l = \{P_\theta(w_i, t_i)\}_{i=1\ldots l}$: emission probabilities trained by Berkeley parser

Run:

1. $\{G\}$ = construct_POSTagGraph (T_l,T_u)
2. $\{Q\}$ = propagate_POSTagProbability ($\{G\}$,E_l)
3. $\{D_l, D_u\}$ = propagate_POSTag ($\{Q\}$, E_l,T_u)
4. For $i = 1, 2, \ldots, N$
5. $\{g_i\}$ = construct_latentGraph (D_l, D_u)
6. $\{q_i\}$ = propagate_latentTagProbability ($\{g_i\}$)
7. E_u = combine ($\{q_i\}_{i=1}^{N}$)

Output:

$E_u = \{P_\theta(w_i, t_i)\}_{i=1\ldots u}$: emission probabilities of unknown words

End

been widely used and shown to outperform other SSL methods (Chapelle et al., 2006). Most of these algorithms are transductive in nature, so they cannot be used to predict an unseen test example in the future (Belkin et al., 2006). Typically, graph-based label propagation algorithms are run in two main steps: graph construction and label propagation. The graph construction provides a natural way to represent data in a variety of target domains. One constructs a graph whose vertices consist of labeled and unlabeled data. Pairs of vertices are connected by weighted edges which encode the degree to which they are expected to have the same label (Zhu et al., 2003). The great importance of graph construction methods leads to a number of graph construction algorithms in the past years. Popular graph construction methods include *k*-nearest neighbors (*k*-NN), *e*-neighborhood, and local reconstruction. In this paper , the *k*-NN method is used to construct the graph. Besides, label propagation operates on the constructed graph. Its primary objective is to propagate labels from a few labeled vertices to the entire graph by optimizing a loss function based on the constraints or properties derived from the graph, e.g. smoothness (Zhu et al., 2003; Subramanya and Bilmes, 2008; Talukdar and Crammer, 2009) or sparsity (Das and Smith, 2012). State-of-the-art label propagation algorithms include LP-ZGL (Zhu et al., 2003), Adsorption (Baluja et al., 2008), MAD (Talukdarand and Crammer, 2009) and Sparse Inducing Penalties (Das and Smith, 2012). The Sparse Inducing Penalties algorithm is used in this study.

3 The Proposed Approach

The emphasis of this paper is on presenting a method to recognize Chinese unknown words by using two different kinds of data sources, e.g. labeled texts and unlabeled texts, to construct a specific similarity graph. In essence, this problem can be treated as incorporating gainful information, e.g. prior knowledge or label constraints, of unlabeled data into the supervised model. In our approach, we employ a transductive graph-based label propagation method to achieve such gainful information, e.g. label distributions are inferred from a similarity graph constructed over labeled and unlabeled data. Then, the derived label distributions are regarded as "soft evidence" to augment the parsing of Chinese unknown words based on a new learning objective function. The algorithm contains the following two stages (see Algorithm 1). Firstly, given labeled data and unlabeled data, i.e. $_l = \{w_i\}_{i=1}^{l}$ with l labeled words and $_u = \{w_i\}_{i=l+1}^{l+u}$ with u unlabeled words, a specific similarity graph $\{G\}$ representing T_l and T_u is constructed (POS tag graph). In this stage, we construct one graph over all of labeled data and unlabeled data and propagate one POS tag for each unlabeled word (see section 3.1). Secondly, probabilities of latent tag $P_\theta(w|t_x)$ are estimated subsequently. In this application, we will generate N graphs. Where N stands for the number of POS types, each graph is aimed at propagating latent tag for the unlabeled words in their most probable POS tag, which can be determined from the graph in first stage (see section 3.2).

Feature	Example
Trigram + Context	我非常开心
Trigram	非常开
Left Context	我非
Right Context	开心
Center Word	常
Left Word + Right Word	非开
Left Word + Right Context	非开心
Left Context + Right Word	我非开

Table 1: Features employed to measure the similarity between two vertices, in a given text example "我非常开心" (I am very happy), where the trigram is "非常开".

3.1 Assigning POS Tags to Unlabeled Words

In this stage (corresponding to procedure 1-3 in Algorithm 1), the common practice is to construct a similarity graph for the labeled data and unlabeled data, and aim at assigning a POS tag to unlabeled data in a vertex constructing and label propagation tradition. The effect of the label propagation depends heavily on the the quality of the graph. Thus graph construction plays a central role in graph-based label propagation (Zhu et al., 2003).

In this stage, we represent vertices by all of the word trigrams with occurrences in labeled and unlabeled sentences to construct the first graph. The graph construction is non-trivial. As Das and Petrov (2011) mentioned that taking individual words as the vertices would result in various ambiguities and the similarity measurement is still challenging. Therefore, in this paper, we follow the same intuitions of graph construction from (Subramanya et al., 2010) by using trigram and the objective focuses on the center word in each vertex. Formally, we are given a set of labeled texts $T_l = \{w_i\}_{i=1}^{l}$, and a set of unlabeled texts $T_u = \{w_i\}_{i=l+1}^{l+u}$. The goal is to form an undirected weighted graph $G = (V, E)$, in which V as the set of vertices, which covers all trigrams extracted from T_l and T_u. Here $V = V_l \cup V_u$, where V_l refers to trigrams that occurs at least once in labeled data and V_u refers to trigrams that occurs only in the unlabeled data. The edge $E \in V \times V$. In our case, we make use of the *k*-nearest neighbors (*k*-NN) (*k*=5) method to construct the graph and the edge weights are measured by a symmetric similarity function as follows:

$$w_{i,j} = \begin{cases} sim(x_i, x_j), & if\ j \in K(i) or\ i \in K(j) \\ 0, & otherwise \end{cases} \quad (4)$$

where x denotes one vertex in the graph, $K(i)$ is the k nearest neighbors of x_i ($|K(i) = k, \forall i|$) and $sim(x_i, x_j)$ is a symmetric similarity measure between two vertices. The similarity function is computed based on the co-occurrence statistics over the features shown in Table 1.

To induce label distributions of unlabeled word from labeled vertices to entire graph, the label propagation algorithm, Sparsity-Inducing Penalties (Sparsity) proposed by (Das and Smith, 2012) is employed in this study. The following convex objective function is optimized in our case:

$$\arg\min_{q} \sum_{j=1}^{l} \|q_j - r_j\|^2 + \mu \sum_{i=1, k \in N(i)}^{m} w_{ik} \|q_i - q_k\|^2 + \lambda \sum_{i=1}^{m} {q_i}^2$$

$$s.t.\ q \geq 0,\ \forall i \in V,\ \|q_i\|_1 = 1. \quad (5)$$

where r_j denotes empirical label distributions of labeled vertices and q_i denotes unnormalized estimate measures in every vertex. The w_{ik} refers the similarity between trigram i and trigram k, and $N(i)$ is a set of neighbors of trigram i. μ and λ are two hyperparameters. The squared-loss[7] criterion is used to formulate the objective function. The first term in Equation (5) is the seed match loss which penalizes q_j if they go too far away from the empirical labeled distribution r_j. The second term is the edge smoothness loss that requires q_i to be smoothed with respect to the graph, such that two vertices connected by an edge with high weight should be assigned similar labels. The final term is a regularizer to incorporate the prior knowledge, e.g. uniform distributions used in (Das and Petrov, 2011; Subramanya et al., 2010).

The estimated label distribution q_i in Equation (5) is relaxed to be unnormalized, which simplifies the optimization. Thus, the objective function in Equation (5) can be optimized by

[7]E.g. $\|p\|^2 = \sum_y p^2(y)$, it can be seen as a multi-class extension of the quadratic cost criterion (Bengio et al., 2007) or as a variant of one of the objectives in (Zhu et al., 2003).

LBFGS-B (Zhu et al., 1997), a generic quasi-Newton gradient-based optimizer.

Mathematically, the problem of label propagation is to get the optimal emission label distribution q_i of every labeled vertex. Integrating the similarity between every two vertices, we can project the most probable POS (selection from the q_i) tag to the unlabeled words.

Through the construction of similarity graph and propagation of labels in this stage, each unlabeled word will get a POS tag.

3.2 Generating Latent Tag and Emission Probability to Unlabeled Words

In this stage (corresponding to procedure 4-7 in Algorithm 1), we mainly construct another type of graph $\{g\}$ to generate latent tag and emission probability to unlabeled words. As mentioned, each unlabeled word gets only one POS tag in stage one. Consequently, we build a graph for each type POS tag respectively in order to obtain an optimal emission probability distribution for each unlabeled word at this stage. When constructing the similarity graph, each vertex represents a word instead of a trigram. Because we only need to consider this word's latent tags and emission probability distribution based on its POS tag generated in the stage one. The graph construction and label propagation procedures are similar to that of the previous stage. It is worth noting that $||q_i||_1 \neq 1$ in the Equation (5) that differs from the previous stage. The emission distribution q_i is generated from all possible vertices with the same POS tag in a similarity graph instead of all of possible POS types of a vertex. Finally, the label distributions can be propagated to the unlabeled words, and the label distribution content is same as the Berkeley lexicon (contain the respective rule scores and words) trained by Berkeley parser.

3.3 Incorporation

After the former steps, we can get a lexicon of unlabeled words with label distribution. The lexicon is treated as an OOV lexicon which covers most of OOV words that appear in testing data but not in the training data in our system. Then this OOV lexicon should be incorporated into the Berkeley parser. Our strategy of insertion is that: when an OOV word is detected, it should be firstly examined if the OOV lexicon contains such word, then corresponding estimation will be used; otherwise, the built-in OOV word model (mentioned in the section 2.1) will be used. During the parameter tuning phase, we try to use linear incorporation to inspect the impact of our OOV model to the whole parsing model:

$$\alpha\theta_o + (1-\alpha)\theta_b \quad (6)$$
$$s.t.\ 0 \leq \alpha \leq 1$$

where θ_o, θ_b denote the estimation generated by our proposed OOV model and the Berkeley model respectively.

3.4 Comparison with Other OOV Recognition Models

The proposed approach in this paper differs from previous OOV recognition models. Collins (2003) assigned the UNKNOWN token to unknown words, and any $tag/word$ pairs not seen in training data would give a zero of estimation. While in (Klein and Manning, 2003), the unknown words were split into one of several word-class categories, based on capitalization, suffix, digit, and other character features. For each of these categories, they took the maximum-likelihood estimation of $P(tag|wordclass)$ and add a parameter k to smooth and accommodate unknown words. In (Petrov et al., 2006), they mainly utilized the estimation of rare words to reflect the appearance likelihood of OOV words and the details of the method have been mentioned in section 2.1. In fact, Chinese words are quite different from English, and the word formation processing for Chinese can be quite complex. Huang et al., (2007) reflected the fact that the characters in any position (prefix, infix, or suffix) can be predictive of the POS type for Chinese words. Inspired by their work, Huang and Harper (2009) improved Chinese unknown word parsing performance by using the geometric average of emission probabilities of all of the characters in the word. Differing from their concerns, we make use of a new perspective to employ unlabeled data to augment the supervised model and to handle the OOV word by graph-based semi-supervised learning. Our emphasis is to learn the semi-supervised model by smoothing the label distributions that are derived from a specific graph constructed with labeled and unlabeled data. Though graph-based knowledge, the OOV label distribution can be generated. It is worth nothing that the selection of unlabeled data should cover OOV words as much as possible. Because this approach is mainly used to assign a POS tag and emission probabilities to each

	Train	Unlabeled	Dev	Test
#Sentence	7,176	19,075	893	912
#Word	201,460	1,110,947	26,170	26,134
#OOV	-	-	2,168	2,223

Table 2: The statistics summary of data.

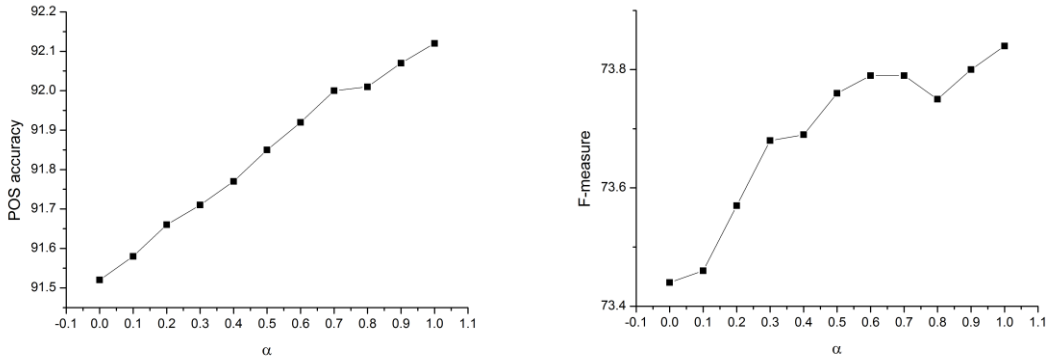

Figure 1: POS and parsing accuracy on development set, corresponding to different α.

unlabeled data according to the similarity between any two vertices in a graph constructing among labeled data and unlabeled data. If all of OOV words are found in the unlabeled data, then each OOV word would be recognized by our model. When we construct a graph where a portion of vertices correspond to labeled instances, and the rest is unlabeled. Pairs of vertices are connected by a weighted edge denoting the similarity between the pair. In this process, optimization of a loss function based on smoothness properties of the graph is performed to propagate labels from the labeled vertices to the unlabeled ones. Overall, our method differs in three important aspects: firstly, the existing resource (e.g. annotated Treebank and the latent variable grammars induced by Berkeley parsing model) is well utilized. Secondly, the training procedure is simpler than the (Huang and Harper, 2011). Thirdly, the derived label information from the graph is smoothed into the model by optimizing a modified objective function.

4 Experiment

4.1 Settings

In our experiment, Xinhua news and Sinorama magazine portions of the most recently released Penn Chinese Treebank 7.0 (CTB 7.0) (Xue et al., 2002) are used as labeled text T_l. Besides, the Peking University Corpus in Second International Chinese Word Segmentation Bakeoff[8] is utilized as unlabeled data T_u. The unlabeled data has been word-segmented with Stanford segmenter (Chang et al., 2008) because it adopts the same segmentation scheme used in the Treebank. The CTB 7.0 corpus was collected during different time periods from different sources with a diversity of articles. In order to obtain a representative experimental data, we refer to the splitting standard of (Huang et al., 2007; Huang and Harper, 2009), dividing the whole corpus into blocks of 10 files sorted by ascending order. For each block, the first file is used for development, the second file is used for testing, and the remaining 8 files are used for training. The corresponding statistic information on the data is shown in Table 2. The development set is used to determine the optimal α value to reflect our OOV model. EVALB (Sekine and Collins, 1997) is used for the evaluation.

4.2 Experiment Results

We firstly run the experiment on development set, the Berkeley baseline model has an overall POS tags accuracy of 91.51% on the development set, which is fairly low compared to the accuracies of importing the graph-based OOV model. In our model, the parameter α is smoothed to accommodate OOV model used in Equation 6. Figure 1 depicts the impact of combining the baseline model (lexical model in Berkeley) and

[8]http://www.sighan.org/bakeoff2005/

	Length	R	P	F	POS
Baseline	All	73.34	75.20	74.25	91.51
	<=40	75.48	76.02	75.75	91.87
α = 1	All	75.12	76.83	75.97	93.79
	<=40	77.34	77.71	77.52	94.19

Table 3: POS and parsing accuracy on testing set.

Models	**Parsing**
Answer:	(IP (NP (NR 河南) (NR 西峡)) (VP (VV 发现) (NP (NN 恐龙) (NN 骨骼) (NN 化石))))
Baseline:	((IP **(NP (NP (NR 河南))(NP (NN 西峡)))** (VP (VV 发现) **(IP (NP (NN 恐龙)**(NN 骨骼)) **(VP (VV 化石))**)))
Our model:	(IP (NP (NR 河南) (NR 西峡)) (VP (VV 发现) (NP (NN 恐龙) (NN 骨骼) (NN 化石))))

Table 4: The parsing results for sentence: 河南西峡发现恐龙骨骼化石 (The dinosaur bone fossils were found in XiXia, Henan province).

#Words in testing set	**#Tag in baseline model**	**Our model**	**Golden**
王翔-12	6-*NR*,4-*NN*,1-*VV*, 1-*AD*	12-*NR*	12-*NR*
书展-12	9-*NN*, 1-*NR*,1-*CD*,1-*JJ*	12-*NN*	12-*NN*
地对-7	5-*NN*, 1-*NR*,1-*JJ*	7-*JJ*	7-*JJ*
捐助-3	1-*VV*, 1-*NN*, 1-*VA*	3-*VV*	3-*VV*
次日-2	2-*AD*	2-*NT*	2-*NT*
轻便-1	1-*AD*	1-*VA*	1-*VA*
多所-1	1-*VV*	1-*AD*	1-*AD*

Table 5: The OOV words correctly tagged by our model.

graph-based OOV model using different α values. When $\alpha = 0$, the model uses only the lexical model estimation. While $\alpha = 1$, it uses only the graph-based OOV model prediction of words. It is interesting to note that the combination model results in significant improvement over the baseline lexical model in terms of F-score and OOV accuracy. When $\alpha = 1$, the estimation performs the best result. This strongly reveals that the knowledge derived from the similarity graph does effectively strengthen the model.

Table 3 demonstrates the parsing result in the testing set. The best improvements in POS tagging and parsing are 2.28% and 1.72% respectively, which are statistically significant.

4.3 Discussion

By incorporating unlabeled data to boost the supervised model, our model outperforms the baseline. The main reason is that unlabeled data lack information, we use transductive graph-based label distributions derived from labeled data. The derived label information is considered as prior knowledge relative to unlabeled data, thereby enriching the training data. Most importantly, the similarity graph can also be allowed to propagate the label distributions for unknown words. The improved performance of the described model can be illustrated by the excerpt in Table 4, extracted from the test data. The table shows the golden parsing in the first line, and the parsing results given by the Berkeley baseline model and our OOV model in the following lines. Parsing errors are marked in red bold. The results achieved by our model for this example are totally correct, whereas the baseline model get the erroneous parsing mainly occurred in generating extra phrasal tags (e.g. NP, IP, VP) and mis-tagging a POS tag (e.g. VV). In which the word “化石” (fossil) is an OOV word in the test data. Our model can properly determine the POS tag for this word with the help of the label distribution by constructing the similarity graph. As mentioned before the OOV lexicon which concludes almost OOV words, and we found the word “化石” (fossil) has assigned with the *NN* tag. So the corresponding estimation with this tag will be used firstly by our model during the parsing. According to the result shown in the Table 3, the POS tag has about 2.3%

improvement. To a great extent, it mainly contributes to the incorporating of the OOV lexicon into the Berkeley parser. The Table 5 shows the sample OOV words are correctly tagged by utilizing the OOV lexicon in parsing. The first column stands for the number of times the word appears in the test data (e.g. 王翔 (WangXiang) - 12 means the word "王翔 (WangXiang)" appears 12 times in the test set). The other three columns stand for the times of this word's with certain POS tag type when paring in the baseline model, our model and golden file respectively. From the table, we can see our OOV model has a high POS accuracy by incorporating the OOV lexicon into the parser. Simultaneously, it proves that the label distribution derived from the similarity graph can augment the parsing of unknown words.

5 Conclusion

In this paper, we show for the first time that the graph-based semi-supervised learning is able to improve the performance of a PCFG-LA parser on OOV words. The approach mainly uses a *k*-nearest-neighbor algorithm to construct a similarity graph based on labeled and unlabeled data and then incorporates the graph knowledge into the Berkeley parser. Experimental comparisons on the Chinese Treebank corpus indicate that the proposed approach yields much better results than the baseline case without using unlabeled data.

In future work, we will concentrate on applying the graph-based OOV model into other parsing model (e.g. coarse-to-fine) and apply the model to other languages.

Acknowledgments

The authors are grateful to the Science and Technology Development Fund of Macau and the Research Committee of the University of Macau for the funding support for our research, under the reference No. 057/2009/A2 and MYRG076(Y1-L2)-FST13-WF. The authors also wish to thank the anonymous reviewers for many helpful comments.

References

Mohammed Attia, Jennifer Foster, Deirdre Hogan, Joseph Le Roux, Lamia Tounsi, and Josef Van Genabith. 2010. Handling unknown words in statistical latent-variable parsing models for Arabic, English and French. In *Proceedings of the NAACL HLT 2010 First Workshop on Statistical Parsing of Morphologically-Rich Languages*, pp. 67–75.

Shumeet Baluja, Rohan Seth, D. Sivakumar, Yushi Jing, Jay Yagnik, Shankar Kumar, Deepak Ravichandran, and Mohamed Aly. 2008. Video suggestion and discovery for youtube: taking random walks through the view graph. In *Proceedings of the 17th international conference on World Wide Web*, pp. 895–904.

Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. 2006. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *The Journal of Machine Learning Research*, 7, 2399–2434.

Pi-Chuan Chang, Michel Galley, and Christopher D. Manning. 2008. Optimizing Chinese word segmentation for machine translation performance. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pp. 224–232.

Olivier Chapelle, Bernhard Schölkopf, Alexander Zien. 2006. *Semi-supervised learning*. MIT press Cambridge.

Eugene Charniak. 2000. A maximum-entropy-inspired parser. In *Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference*, pp. 132–139.

Eugene Charniak and Mark Johnson. 2005. Coarse-to-fine n-best parsing and MaxEnt discriminative reranking. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pp. 173–180.

Michael Collins. 2003. Head-driven statistical models for natural language parsing. *Computational linguistics*, 29(4), 589–637.

Dipanjan Das and Slav Petrov. 2011. Unsupervised part-of-speech tagging with bilingual graph-based projections. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 600–609.

Dipanjan Das and Noah A. Smith. 2012. Graph-based lexicon expansion with sparsity-inducing penalties. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 677–687.

Mary Harper and Zhongqiang Huang. 2011. Chinese statistic parsing. In Joseph Olive, Caitlin Christianson, and John McCary, editors, *Handbook of Natural Language Processing and Machine Translation*. Springer Verlag.

Liangye He, Derek F. Wong, and Lidia S. Chao. 2012. Adapting multilingual parsing models to Sinica treebank. In *Proceedings of the Second CIPS-SIGHAN Joint Conference on Chinese Language Processing*, pp. 211-215.

Qiuping Huang, Liangye He, Derek F. Wong, and Lidia S. Chao. 2012. A simplified Chinese parser with factored model. In *Proceedings of the Second CIPS-SIGHAN Joint Conference on Chinese Language Processing*, pp. 188-193.

Zhongqiang Huang and Mary Harper. 2009. Self-Training PCFG grammars with latent annotations across languages. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2-Volume 2*, pp. 832–841.

Zhongqiang Huang and Mary Harper. 2011. Feature-rich log-linear lexical model for latent variable PCFG grammars. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pp. 219–227.

Zhongqiang Huang, Mary Harper, and Wen Wang. 2007. Mandarin part-of-speech tagging and discriminative reranking. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pp. 1093–1102.

Dan Klein and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pp. 423–430.

Roger Levy and Christopher Manning. 2003. Is it harder to parse Chinese, or the Chinese Treebank? In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pp. 439–446.

Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein. 2006. Learning accurate, compact, and interpretable tree annotation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pp. 433–440.

Slav Petrov and Dan Klein. 2007. Improved inference for unlexicalized parsing. *Human language technologies 2007: the conference of the North American chapter of the Association for Computational Linguistics*, pp. 404–411.

Satoshi Sekine and Michael Collins. 1997. *Evalb*. Available at nlp.cs.nyu.edu/evalb.

Amarnag Subramanya and Jeff Bilmes. 2008. Soft-supervised learning for text classification. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 1090–1099.

Amarnag Subramanya, Slav Petrov, and Fernando Pereira. 2010. Efficient graph-based semi-supervised learning of structured tagging models. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pp. 167–176.

Partha Pratim Talukdar and Koby Crammer. 2009. New regularized algorithms for transductive learning. *Machine Learning and Knowledge Discovery in Databases*, pp. 442–457.

Nianwen Xue, Fu-Dong Chiou, and Martha Palmer. 2002. Building a large-scale annotated Chinese corpus. In *Proceedings of the 19th international conference on Computational linguistics-Volume 1*, pp. 1–8.

Xiaodong Zeng, Derek F. Wong, Lidia S. Chao, and Isabel Trancoso. 2013. Graph-based semi-supervised model for Joint Chinese word segmentation and part-of-speech tagging. In *Proceeding of the 51st Annual Meeting of the Association for Computational Linguistics*, pp. 770–779. Sofia, Bulgaria.

Ciyou Zhu, Richard H. Byrd, Peihuang Lu, and Jorge Nocedal. 1997. L-BFGS-B: Fortran subroutines for large scale bound constrained optimization. *ACM Transactions on Mathematical Software*, 23:550–560.

Xiaojin Zhu, John Lafferty, and Zoubin Ghahramani. 2003. Combining active learning and semi-supervised learning using Gaussian fields and harmonic functions. *ICML 2003 workshop on the continuum from labeled to unlabeled data in machine learning and data mining*, pp. 58–65.