An Empirical Investigation of Word Representations for Parsing the Web

Sorami Hisamoto Kevin Duh Yuji Matsumoto
Graduate School of Information Science
Nara Institute of Science and Technology

{sorami-h, kevinduh, matsu}@is.naist.jp

1 Introduction

Parsing web text is progressively becoming important for many applications in natural language processing, such as machine translation, information retrieval, and sentiment analysis. Current syntactic parsing has been focused on canonical data such as newswires. When evaluated on standard benchmarks such as Wall Street Journal data set, current state-of-the-art parsers achieve accuracies well above 90%. However the accuracy drops dramatically when they are applied to new domains such as web data, barely over 80%. In order to make progress in many applications that rely on parsing, we need robust parsers that can handle such texts.

One approach that is becoming popular recently is to use unsupervised word representations as extra features. Koo et al. [1] has shown that unsupervised clustering features are effective to improve dependency parsing. Turian et al. [2] examined clustering and unsupervised word embedding features on chunking and named entity recognition tasks. Unsupervised word embeddings are dense, low-dimensional and real-value vectors representing words, often induced by neural language models. They have shown that these word representation features lead to improvement in the performances. These word representations are induced by unsupervised methods, thus they are good for new domains such as the web, which has enormous amount of unlabeled data but little labeled data.

In this paper we investigate the effect of unsupervised word representation features on dependency parsing with web texts. We consider two different kinds of word representations, namely Brown clustering and word embeddings induced from a neural language model. To the best of our knowledge, this is the first work that systematically examines these word representations on the task of dependency parsing on web text.

2 Dependency Parsing of Web Text

For the investigation we use Google Web Treebank, which were used for SANCL 2012 shared task on parsing the web [3]. The treebank covers five domains: Yahoo! Answers, Emails, Newsgroups, Local Business Reviews and Weblogs. For each domain there is a large unlabeled data sets. A much smaller subset is sampled randomly and manually annotated, and are divided into development and test sets.

Note that this investigation does not exactly follow the setup of the shared task; For the original shared task, they use Wall Street Journal data set provided for training and then use development and test sets from each domain. However, for this investigation we use development set of each domain for training, thus this is not a domain adaptation task. We assume the situation where we have limited labeled data sets and large unlabeled data sets in new

domains.

The shared task has constituency and dependency parsing tracks, and we focus on dependency parsing for this investigation. For dependency parsing, we use graph-based parser with arc-factored model [4, 5]. We adopt off-the-shelf parser implementation[1] and add extra word representation features on top of the default features. The implementation has higher-order parsing model but we only consider first-order model here.

3 Word Representations

We use two different kinds of unsupervised word representations as extra features for dependency parsing, namely Brown clustering and Collobert and Weston word embeddings. We chose these two due to their good performance reported in chunking and named entity recognition in [2].

3.1 Brown Clustering

Brown clustering [6] is a hierarchical clustering algorithm based on class-based bigram language model. Brown clustering has been shown to improve accuracy of dependency parser [1]. Previous works use short bit-string prefixes of the hierarchy, combined with words or part-of-speech tags, as features. We follow the same strategy to construct feature templates. We use off-the-shelf implementation[2] to cluster development set and unlabeled data set of the domain. For the experiments, we set the cluster number to 50. Note that we perform the clustering separately for each domain, since we focus here on building independent parsers.

3.2 Word Embeddings

Another word representation we consider is Collobert and Weston (C&W) word embeddings [7]. A word embedding, word represented in a dense low-dimensional real-value vector form, is induced from a neural language model, which uses a neural network as the underlying predictive model. In this investigation we consider two pre-processing methods to adopt the word embeddings as features; one is to convert embeddings directly to bit-strings, and the other to cluster embeddings then use these clustering information obtained as features. For both methods we use the embedding information combined with other information such as words or part-of-speech tags as features.

We use a very simple method to convert an embedding, a real-value vector, into a bit-string. For each real-value in vector, we give a bit *1* if the value is positive and else give bit *0*. Then we concatenate all bits to construct a bit-string that has the same length as embedding dimension size.

Another method is to cluster the word embeddings. We use a clustering software package called CLUTO[3] with repeated bisection algorithm to cluster embeddings, then use acquired cluster IDs as features. For the experiments we cluster word embeddings to 50 and 1000 clusters.

We download word embeddings already trained on a neural language model[4] for the experiments. We use 50 dimensions unscaled C&W embeddings for the experiments. Note that these embeddings are trained on RCV1 corpus, which contains one year of Reuters English newswire data.

4 Experiments

We conduct experiments with five different systems; default system (*Baseline*), a system with Brown clustering features (*Brown, 50 clusters*), a system with word embedding converted to bit-string features (*C&W, bit-string*), and systems with word embedding cluster features (*C&W, n clusters*).

We use data sets with gold and predicted part-of-speech tags. To make data sets with predicted part-of-speech tags, we used Stanford Part-Of-Speech Tagger[5]. For tagging we used the model trained on Wall Street Journal data sets, which came with the tagger package.

We adopt the projective parsing algorithm, i.e.,

[1] http://sourceforge.net/projects/mstparser/
[2] https://github.com/percyliang/brown-cluster
[3] http://glaros.dtc.umn.edu/gkhome/cluto/cluto/overview
[4] http://metaoptimize.com/projects/wordreprs/
[5] http://nlp.stanford.edu/software/tagger.shtml

Table 1: Unlabeled accuracies relative to baseline. Boldface indicates improvements of 0.3% or more over the baseline.

	Gold POS tags					Predicted POS tags				
System	Answer	Email	Newsgroup	Review	Weblog	Answer	Email	Newsgroup	Review	Weblog
Baseline	81.45	83.41	80.94	84.18	81.54	80.51	81.29	78.88	82.78	78.82
Brown, 50 clusters	-0.59	-0.10	-0.35	-0.18	-0.06	-0.61	**+0.91**	**+0.33**	**+0.53**	**+0.91**
C&W, bit-string	**+0.38**	+0.08	-0.21	-0.03	-0.34	-0.08	+0.11	-0.48	-0.18	-0.04
C&W, 50 clusters	-0.07	+0.13	-0.93	-0.47	-0.29	+0.04	**+0.83**	+0.19	+0.08	**+0.48**
C&W, 1000 clusters	-0.10	+0.10	-0.57	+0.01	-0.08	-0.11	**+0.96**	+0.04	**+0.34**	**+0.33**

Table 2: Labeled accuracies relative to baseline.

	Gold POS tags					Predicted POS tags				
System	Answer	Email	Newsgroup	Review	Weblog	Answer	Email	Newsgroup	Review	Weblog
Baseline	77.26	80.52	77.63	80.32	78.48	75.96	77.96	75.11	78.54	75.05
Brown, 50 clusters	-0.47	-0.10	-0.26	-0.13	-0.05	-0.51	**+0.92**	**+0.33**	**+0.49**	**+1.10**
C&W, bit-string	**+0.32**	+0.10	-0.15	-0.02	-0.36	+0.02	+0.18	-0.41	-0.06	+0.23
C&W, 50 clusters	-0.03	+0.09	-0.82	-0.36	-0.33	-0.02	**+0.89**	+0.13	+0.12	**+0.57**
C&W, 1000 clusters	-0.05	+0.06	-0.43	+0.03	+0.01	-0.15	**+0.90**	+0.01	**+0.35**	**+0.57**

Eisner algorithm during training. We also tried non-projective algorithm, i.e., Chu-Liu-Edmonds algorithm with the same experiment setting and observed similar results and trends. However the results with non-projective algorithm were slightly lower than the ones with projective algorithm, as all dependencies in the data sets are projective and projective algorithm has constraint that all dependencies are projective.

4.1 Results and Discussion

Table 1 and 2 show unlabeled and labeled accuracies of each system relative to the baseline, respectively. Here accuracy is the percentage of words that have correct head tokens, and for the labeled case both the head and the arc label must be correct. In order to see trends easily, we boldface all results that obtain a 0.3 improvement over the baseline.

We can clearly see that the results with gold part-of-speech tags outperform the results with predicted part-of-speech tags significantly. From this we can observe tagging accuracy highly affects the dependency parsing results, as reported in [3].

For experiments with gold part-of-speech tag data sets, we cannot see much improvement in results by adding extra features to the baseline system for most cases. However, for experiments with predicted part-of-speech tag data sets, we see good improvement in accuracies. For example, in the Email domain, both Brown clustering and C&W 1000 clusters improved unlabeled accuracy by 0.9%, improving the baseline accuracy from 81.29% to 82.2%.

Even for the same system, we can observe that the trend differs among different domains. A significant improvement in one domain does not necessary mean the same trend on other domains.

When we examine results in predicted part-of-speech tag setting, systems on Answer domain performs significantly different compared to other domains. Results of all the systems with embedding features do not vary much from the baseline. On the other hand, the system with Brown cluster features performs significantly worse than the baseline, whereas the results on all other domains are significantly better. Conceivably this is because the

Answer domain is furthest from canonical newswire domain, especially in the kinds of syntactic structures that it contains (questions, imperatives, etc.), as reported in [3].

For majority of cases with predicted part-of-speech tag data sets, the system with Brown clustering features outperforms ones with word embedding features. This is possibly because the Brown clusters were obtained from in-domain web data sets, whereas the word embeddings we used are induced from an out-of-domain newswire data set. Nevertheless, it is promising to see that the cases with embedding cluster features generally does not degrade the baseline.

On predicted part-of-speech tag setting, systems with embedding cluster features generally improve the results, and more number of cluster does not necessary mean better results. On the other hand, the system with embedding bit-string features does not give good improvement for any of the domains, or even degrade the baseline.

5 Conclusions

We investigated the effect of unsupervised word representations on parsing the web texts. We considered two different word representations, namely Brown clustering and Collobert and Weston word embeddings. We observed the word representation features does not improve the results when we evaluate them on data sets with gold part-of-speech tags. However, when we evaluated these features on data sets with predicted part-of-speech tags, we observe improvements in 4 out of 5 domains, using the Brown clustering features. This implies that the clustering features capture some of the essential part-of-speech information for dependency parsing.

Note that the word embeddings we used here are trained on newswire data sets. We can assume by using the word embeddings trained on in-domain data sets that we train and evaluate parser on, we will observe better improvement in accuracies. Future work should train neural language model using web texts to induce word embeddings, and use them as the features.

References

[1] Terry Koo, Xavier Carreras, and Michael Collins. Simple semi-supervised dependency parsing. In *Meeting of the Association for Computational Linguistics*, pages 595–603, 2008.

[2] Joseph Turian, Lev Ratinov, and Yoshua Bengio. Word representations: A simple and general method for semi-supervised learning. In *Meeting of the Association for Computational Linguistics*, pages 384–394, 2010.

[3] Slav Petrov and Ryan McDonald. Overview of the 2012 shared task on parsing the web. Notes of the First Workshop on Syntactic Analysis of Non-Canonical Language, 2012.

[4] Ryan McDonald, Koby Crammer, and Fernando Pereira. Online large-margin training of dependency parsers. In *Meeting of the Association for Computational Linguistics*, pages 91–98, 2005.

[5] Ryan McDonald, Fernando Pereira, Kiril Ribarov, and Jan Hajič. Non-projective dependency parsing using spanning tree algorithms. In *Human Language Technologies and Empirical Methods in Natural Language Processing*, pages 523–530, 2005.

[6] Peter Brown, Peter Desouza, Robert Mercer, Vincent Della Pietra, and Jenifer Lai. Class-based n-gram models of natural language. *Computational Linguistics*, 18:467–479, 1992.

[7] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493–2537, 2011.