Dependency Parsing for Weibo: An Efficient Probabilistic Logic Programming Approach

William Yang Wang, Lingpeng Kong, Kathryn Mazaitis, William W. Cohen
Language Technologies Institute & Machine Learning Department
Carnegie Mellon University
Pittsburgh, PA 15213, USA.
{yww,lingpenk,krivard,wcohen}@cs.cmu.edu

Abstract

Dependency parsing is a core task in NLP, and it is widely used by many applications such as information extraction, question answering, and machine translation. In the era of social media, a big challenge is that parsers trained on traditional newswire corpora typically suffer from the domain mismatch issue, and thus perform poorly on social media data. We present a new GFL/FUDG-annotated Chinese treebank with more than 18K tokens from Sina Weibo (the Chinese equivalent of Twitter). We formulate the dependency parsing problem as many small and parallelizable arc prediction tasks: for each task, we use a programmable probabilistic first-order logic to infer the dependency arc of a token in the sentence. In experiments, we show that the proposed model outperforms an off-the-shelf Stanford Chinese parser, as well as a strong MaltParser baseline that is trained on the same in-domain data.

1 Introduction

Weibo, in particular Sina Weibo[1], has attracted more than 30% of Internet users (Yang et al., 2012), making it one of the most popular social media services in the world. While Weibo posts are abundantly available, NLP techniques for analyzing Weibo posts have not been well-studied in the past.

Syntactic analysis of Weibo is made difficult for three reasons: first, in the last few decades, Computational Linguistics researchers have primarily focused on building resources and tools using standard English newswire corpora[2], and thus, there are fewer resources in other languages in general. Second, microblog posts are typically short, noisy (Gimpel et al., 2011), and can be considered as a "dialect", which is very different from news data. Due to the differences in genre, part-of-speech taggers and parsers trained on newswire corpora typically fail on social media texts. Third, most existing parsers use language-independent standard features (McDonald et al., 2005), and these features may not be optimal for Chinese (Martins, 2012). To most of the application developers, the parser is more like a blackbox, which is not directly programmable. Therefore, it is non-trivial to adapt these generic parsers to language-specific social media text.

In this paper, we present a new probabilistic dependency parsing approach for Weibo, with the following contributions:

- We present a freely available Chinese Weibo dependency treebank[3], manually annotated with more than 18,000 tokens;
- We introduce a novel probabilistic logic programming approach for dependency arc prediction, making the parser directly programmable for theory engineering;
- We show that the proposed approach outperforms an off-the-shelf dependency parser, as well as a strong baseline trained on the same in-domain data.

In the next section, we describe existing work on dependency parsing for Chinese. In Section 3, we present the new Chinese Weibo Treebank to the research community. In Section 4, we introduce the proposed efficient probabilistic programming approach for parsing Weibo. We show the experimental results in Section 5, and conclude in Section 6.

[1] http://en.wikipedia.org/wiki/Sina_Weibo

[2] For example, Wall Street Journal articles are used for building the Penn Treebank (Marcus et al., 1993).

[3] http://www.cs.cmu.edu/~yww/data/WeiboTreebank.zip

2 Related Work

Chinese dependency parsing has attracted many interests in the last fifteen years. Bikel and Chiang (2000; 2002) are among the first to use Penn Chinese Tree Bank for dependency parsing, where they adapted Xia's head rules (Xia, 1999). An important milestone for Chinese dependency parsing is that, a few years later, the CoNLL shared task launched a track for multilingual dependency parsing, which also included Chinese (Buchholz and Marsi, 2006; Nilsson et al., 2007). These shared tasks soon popularized Chinese dependency parsing by making datasets available, and there has been growing amount of literature since then (Zhang and Clark, 2008; Nivre et al., 2007; Sagae and Tsujii, 2007; Che et al., 2010; Carreras, 2007; Duan et al., 2007).

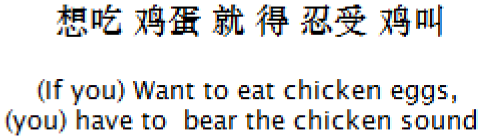
Besides the CoNLL shared tasks, there are also many interesting studies on Chinese dependency parsing. For example, researchers have studied case (Yu et al., 2008) and morphological (Li and Zhou, 2012) structures for learning a Chinese dependency parser. Another direction is to perform joint learning and inference for POS tagging and dependency parsing (Li et al., 2011; Hatori et al., 2011; Li et al., 2011; Ma et al., 2012). In recent years, there has been growing interests in dependency arc prediction in Chinese (Che et al., 2014), and researchers have also investigated character-level Chinese dependency parsing (Zhang et al., 2014). However, even though the above methods all have merits, the results are reported only on standard newswire based Chinese Treebank (e.g. from People's Daily (Liu et al., 2006)), and it is unclear how they would perform on Weibo data.

To the best of our knowledge, together with the recent study on parsing tweets (Kong et al., 2014), we are among the first to study the problem of dependency parsing for social media text.

3 The Chinese Weibo Treebank

We use the publicly available μtopia dataset (Ling et al., 2013) for dependency annotation. An interesting aspect of this Weibo dataset is that, besides the Chinese posts, it also includes a copy of the English translations. This allows us to observe some interesting phenomena that mark the differences of the two languages. For example:

- Function words are more frequently used in English than in Chinese. When examining this English version of the Weibo corpus for the total counts of the word "the", there are 2,084 occurrences in 2,003 sentences. Whereas in Chinese, there are only 52 occurrences of the word "the" out of the 2,003 sentences.



想吃 鸡蛋 就 得 忍受 鸡叫

(If you) Want to eat chicken eggs,
(you) have to bear the chicken sound

Figure 1: An example of pro-drop phenomenon from the Weibo data.

- The other interesting thing is the position of the head. In English, the head of the tree occurs more frequent on the left-to-middle of the sentence, while the distribution of the head is more complicated in Chinese. This is also verified from the parallel Weibo data.

- Another well-known issue in Chinese is that Chinese is a pro-drop topical language. This is extremely prominent in the short text, which clearly creates a problem for parsing. For example, in the Chinese Weibo data, we have observed the sentence in Figure 1.

To facilitate the annotation process, we first preprocess the Weibo posts using the Stanford NLP pipeline, including a Chinese Word Segmenter (Tseng et al., 2005) and a Chinese Part-of-Speech tagger (Toutanova and Manning, 2000). Two native speakers of Chinese with strong linguistic backgrounds have annotated the dependency relations from 1,000 posts of the μtopia dataset, using the FUDG (Schneider et al., 2013) and GFL annotation tool (Mordowanec et al., 2014). The annotators communicate regularly during the annotation process, and a coding manual that relies majorly on the Stanford Dependencies (Chang et al., 2009) is designed. The annotation process has two stages: in the first stage, we rely on the word segmentation produced by the segmenter, and produce a draft version of the treebank; in the second stage, the annotators actively discuss the difficult cases to reach agreements, manually correct the mis-segmented word tokens, and revise the annotations of the tricky cases. The final inter-annotator agreement rate on a randomly-selected subset of 373 tokens in this treebank is 82.31%.

Fragmentary Unlabeled Dependency Grammar (FUDG) is a newly proposed flexible framework that offers a relative easy way to annotate the syntactic structure of text. Beyond the traditional tree view of dependency syntax in which the tokens of a sentence form nodes in a tree, FUDG also allows the annotation of additional lexical items such as multiword expressions. It provides special devices for coordination and coreference; and facilitates underspecified (partial) annotations where producing a complete parse would be difficult. Graph Fragment Language (GFL) is an implementation of unlabeled dependency annotations in the FUDG framework, which fully supports Chinese, English and other languages. The training set of our Chinese Weibo Treebank[4] includes 14,774 tokens, while the development and test sets include 1,846 and 1,857 tokens respectively.

4 A Programmable Parser with Personalized PageRank Inference

A key problem in multilingual dependency parsing is that generic feature templates may not work well for every language. For example, Martins (2012) shows that for Chinese dependency parsing, when adding the generic grandparents and siblings features, the performance was worse than using the standard bilexical, unilexical, and part-of-speech features. Unfortunately, for many parsers such as Stanford Chinese Parser (Levy and Manning, 2003) and MaltParser (Nivre et al., 2007), it is very difficult for programmers to specify the feature templates and inference rules for dependency arc prediction.

In this work, we present a Chinese dependency parsing method for Weibo, based on efficient probabilistic first-order logic programming (Wang et al., 2013). The advantage of probabilistic programming for parsing is that, software engineers can simply conduct theory engineering, and optimize the performance of the parser for a specific genre of the target language. Recently, probabilistic programming approaches (Goodman et al., 2012; Wang et al., 2013; Lloyd et al., 2014) have demonstrated its efficiency and effectiveness in many areas such as information extraction (Wang et al., 2014), entity linking, and text classification (Wang et al., 2013).

Algorithm 1 A Dependency Arc Inference Algorithm for Parsing Weibo

```
Given:
(1) a sentence with tokens $T_i$, where $i$ is the index, and $L$ is the length;
(2) a database $D$ of token relations from the corpus;
(3) first-order logic inference rule set $R$.

for $i = 1 \rightarrow L$ tokens do
    $\mathbb{S} \leftarrow ConstructSearchSpace(T_i, R, D)$;
    $\vec{P_i} \leftarrow InferParentUsingProPPR(T_i, \mathbb{S})$;
end for

Greedy Global Inference
for $i = 1 \rightarrow L$ tokens do
    $Y_i = \arg\max \vec{P_i}$;
end for
```

4.1 Problem Formulation

We formulate the dependency parsing problem as many small dependency arc prediction problems. For each token, we form the parent inference problem of a token T_i as solving a query $edge(T_i, ?)$ using stochastic theorem proving on a search graph. Our approach relies on a database D of inter-token relations. To construct the database, we automatically extract the token relations from the text data. For example, to denote the adjacency of two tokens T_1 and T_2, we store the entry $adjacent(T_1, T_2)$ in D. One can also store the part-of-speech tag of a token in the form $haspos(T_1, DT)$. There is no limitations on the arity and the types of the predicates in the database.

Given the database of token relations, one then needs to construct the first-order logic inference theory R for predicting dependency arcs. For example, to construct simple bilexical and bi-POS inference rules to model the dependency of an adjacent head and a modifier, one can write first-order clauses such as:

```
edge(V1,V2) :-
  adjacent(V1,V2),hasword(V1,W1),
  hasword(V2,W2),keyword(W1,W2) #adjWord.

edge(V1,V2) :-
  adjacent(V1,V2),haspos(V1,W1),
  haspos(V2,W2),keypos(W1,W2) #adjPos.

keyword(W1,W2) :- # kw(W1,W2).
keypos(W1,W2) :- # kp(W1,W2).
```

[4] The corpus is freely available for download at the URL specified in Section 1.

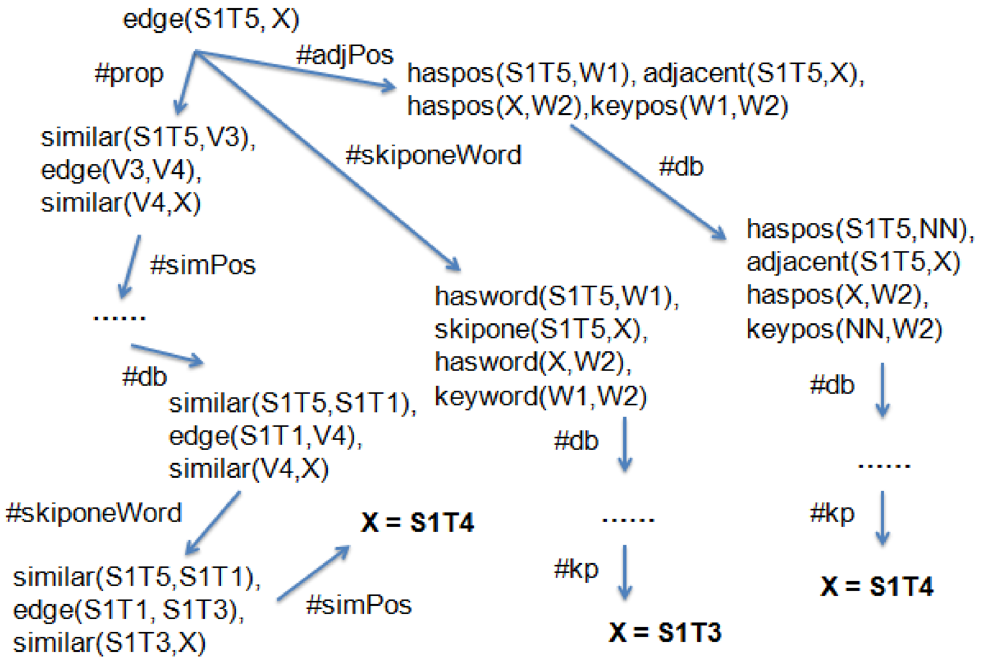


Figure 2: After mapping the database D to theory R, here is an example of search space for dependency arc inference. The query is $edge(S_1T_5, X)$, and there exists one correct and multiple incorrect solutions (highlighted in **bold**).

Here, we associate a feature vector ϕ_c with each clause, which is annotated using the # symbol after each clause in the theory set. Note that the last two (*keyword* and *keypos*) clauses are feature templates that allow us to learn the specific bi-POS tags and bilexical words from the data. In order for one to solve the query $edge(T_i, ?)$, we first need to map the entities from D to R to construct the search space. The details for constructing and searching in the graph can be found in previous studies on probabilistic first-order logic (Wang et al., 2013) and stochastic logic programs (Cussens, 2001). An example search space is illustrated in Figure 2. Note that now the edges in the search graph correspond to the feature vector ϕ_c in R.

The overall dependency arc inference algorithm can be found in Algorithm 1. For each of the parent inference subtask, we use ProPPR (Wang et al., 2013) to perform efficient personalized PageRank inference. Note that to ensure the validity of the dependency tree, we break the loops in the final parse graph into a parse tree using the maximum personalized PageRank score criteria. When multiple roots are predicted, we also select the most likely root by comparing the personalized PageRank solution scores.

To learn the more plausible theories, one needs to upweight weights for relevant features, so that they have higher transition probabilities on the corresponding edges. To do this, we use stochastic gradient descent to learn from training queries, where the correct and incorrect solutions are known. The details of the learning algorithm are described in the last part of this section.

4.2 Personalized PageRank Inference

For the inference of the parent of each token, we utilize ProPPR (Wang et al., 2013). ProPPR allows a fast approximate proof procedure, in which only a small subset of the full proof graph is generated. In particular, if α upper-bounds the reset probability, and d upperbounds the degree of nodes in the graph, then one can efficiently find a subgraph with $O(\frac{1}{\alpha\epsilon})$ nodes which approximates the weight for every node within an error of $d\epsilon$ (Wang et al., 2013), using a variant of the PageRank-Nibble algorithm of Andersen *et al* (2008).

4.3 Parameter Estimation

Our parameter learning algorithm is implemented using a parallel stochastic gradient descent variant to optimize the log loss using the supervised personalized PageRank algorithm (Backstrom and

Method	Dev.	Test
Stanford Parser (Xinhua)	0.507	0.489
Stanford Parser (Chinese)	0.597	0.581
MaltParser (Full)	**0.669**	0.654
Our methods — ProPPR		
ReLU (Bi-POS)	0.506	0.517
ReLU (Bilexical)	0.635	0.616
ReLU (Full)	0.668	0.666
Truncated $tanh$ (Bi-POS)	0.601	0.594
Truncated $tanh$ (Bilexical)	0.650	0.634
Truncated $tanh$ (Full)	0.667	**0.675***

Table 1: Comparing our Weibo parser to other baselines (UAS). The off-the-shelf Stanford parser uses its attached Xinhua and Chinese factored models, which are trained on external Chinese treebank of newswire data. MaltParser was trained on the same in-domain data as our proposed approach. * indicates $p < .001$ comparing to the MaltParser.

Leskovec, 2011). The idea is that, given the training queries, we perform a random walk with restart process, and upweight the edges that are more likely to end up with a known correct parent. We learn the transition probability from two nodes (u, v) in the search graph using: $\Pr_{\mathbf{w}}(v|u) = \frac{1}{Z} f(\mathbf{w}, \Phi^c_{\text{restart}})$, where we use two popular nonlinear parameter learning functions from the deep learning community:

- Rectified Linear Unit (ReLU) (Nair and Hinton, 2010): $max(0, x)$;
- The Hyperbolic Function (Glorot and Bengio, 2010): $tanh(x)$.

as the f in this study. ReLU is a desirable non-linear function, because it does not have the vanishing gradient problem, and produces sparse weights. For the weights learned from $tanh(x)$, we truncate the negative weights on the edges, since the default weight on the feature edges is $\mathbf{w} = 1.0$ (existence), and $\mathbf{w} = 0.0$ means that the edge does not exist in the inference stage.

5 Experiments

In this experiment, we compare the proposed parser with two well-known baselines. First, we compare with an off-the-shelf Stanford Chinese Parser (Levy and Manning, 2003). Second, we compare with the MaltParser (Nivre et al., 2007) that is trained on the same in-domain Weibo dataset. The train, development, and test splits are described in Section 3. We tune the regularization hyperparameters of the models on the dev. set, and report Unlabeled Attachment Score (UAS) results for both the dev. set and the hold-out test set. We experiment with the bilexical and bi-POS first-order logic theory separately, as well as a combined full model with directional and distance features.

The results are shown in Table 1. We see that both of the two attached pre-trained models from the Stanford parser do not perform very well on this Weibo dataset, probably because of the mismatched training and test data. MaltParser is widely considered as one of the most popular dependency parsers, not only because of its speed, but also the acclaimed accuracy. We see that when using the full model, the UAS results between our methods and MaltParser are very similar on the development set, but both of our approaches outperform the Maltparser in the holdout test set. The truncated $tanh$ variant of ProPPR obtains the best UAS score of 0.675.

6 Conclusion

In this paper, we present a novel Chinese dependency treebank, annotated using Weibo data. We introduce a probabilistic programming dependency arc prediction approach, where theory engineering is made easy. In experiments, we show that our methods outperform an off-the-shelf Stanford Chinese Parser, as well a strong MaltParser that is trained on the same in-domain data. The Chinese Weibo Treebank is made freely available to the research community. In the future, we plan to apply the proposed approaches to dependency and semantic parsing of other languages.

Acknowledgements

We are grateful to anonymous reviewers for useful comments. This research was supported in part by DARPA grant FA8750-12-2-0342 funded under the DEFT program, and a Google Research Award. The authors are solely responsible for the contents of the paper, and the opinions expressed in this publication do not reflect those of the funding agencies.

References

Reid Andersen, Fan R. K. Chung, and Kevin J. Lang. 2008. Local partitioning for directed graphs using pagerank. *Internet Mathematics*, 5(1):3–22.

Lars Backstrom and Jure Leskovec. 2011. Supervised random walks: predicting and recommending links in social networks. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 635–644. ACM.

Daniel M Bikel and David Chiang. 2000. Two statistical parsing models applied to the chinese treebank. In *Proceedings of the second workshop on Chinese language processing: held in conjunction with the 38th Annual Meeting of the Association for Computational Linguistics-Volume 12*, pages 1–6. Association for Computational Linguistics.

Sabine Buchholz and Erwin Marsi. 2006. Conll-x shared task on multilingual dependency parsing. In *Proceedings of the Tenth Conference on Computational Natural Language Learning*, pages 149–164. Association for Computational Linguistics.

Xavier Carreras. 2007. Experiments with a higher-order projective dependency parser. In *EMNLP-CoNLL*, pages 957–961.

Pi-Chuan Chang, Huihsin Tseng, Dan Jurafsky, and Christopher D Manning. 2009. Discriminative reordering with chinese grammatical relations features. In *Proceedings of the Third Workshop on Syntax and Structure in Statistical Translation*, pages 51–59. Association for Computational Linguistics.

Wanxiang Che, Zhenghua Li, and Ting Liu. 2010. Ltp: A chinese language technology platform. In *Proceedings of the 23rd International Conference on Computational Linguistics: Demonstrations*, pages 13–16. Association for Computational Linguistics.

Wanxiang Che, Jiang Guo, and Ting Liu. 2014. Reliable dependency arc recognition. *Expert Systems with Applications*, 41(4):1716–1722.

David Chiang and Daniel M. Bikel. 2002. Recovering latent information in treebanks. In *Proceedings of the 19th International Conference on Computational Linguistics - Volume 1*, COLING '02, pages 1–7, Stroudsburg, PA, USA. Association for Computational Linguistics.

James Cussens. 2001. Parameter estimation in stochastic logic programs. *Machine Learning*, 44(3):245–271.

Xiangyu Duan, Jun Zhao, and Bo Xu. 2007. Probabilistic parsing action models for multi-lingual dependency parsing. In *EMNLP-CoNLL*, pages 940–946.

Kevin Gimpel, Nathan Schneider, Brendan O'Connor, Dipanjan Das, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan, and Noah A Smith. 2011. Part-of-speech tagging for twitter: Annotation, features, and experiments. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 42–47. Association for Computational Linguistics.

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *International Conference on Artificial Intelligence and Statistics*, pages 249–256.

Noah Goodman, Vikash Mansinghka, Daniel Roy, Keith Bonawitz, and Daniel Tarlow. 2012. Church: a language for generative models. *arXiv preprint arXiv:1206.3255*.

Jun Hatori, Takuya Matsuzaki, Yusuke Miyao, and Jun'ichi Tsujii. 2011. Incremental joint pos tagging and dependency parsing in chinese. In *IJCNLP*, pages 1216–1224.

Lingpeng Kong, Nathan Schneider, Swabha Swayamdipta, Archna Bhatia, Chris Dyer, and Noah A. Smith. 2014. A dependency parser for tweets. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*, Doha, Qatar, October. ACL.

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*, pages 439–446. Association for Computational Linguistics.

Zhongguo Li and Guodong Zhou. 2012. Unified dependency parsing of chinese morphological and syntactic structures. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1445–1454. Association for Computational Linguistics.

Zhenghua Li, Min Zhang, Wanxiang Che, Ting Liu, Wenliang Chen, and Haizhou Li. 2011. Joint models for chinese pos tagging and dependency parsing. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1180–1191. Association for Computational Linguistics.

Wang Ling, Guang Xiang, Chris Dyer, Alan Black, and Isabel Trancoso. 2013. Microblogs as parallel corpora. In *Proceedings of the 51st Annual Meeting on Association for Computational Linguistics*, ACL '13. Association for Computational Linguistics.

Ting Liu, Jinshan Ma, and Sheng Li. 2006. Building a dependency treebank for improving chinese parser. *Journal of Chinese Language and Computing*, 16(4):207–224.

James Robert Lloyd, David Duvenaud, Roger Grosse, Joshua B Tenenbaum, and Zoubin Ghahramani. 2014. Automatic construction and natural-language description of nonparametric regression models. *arXiv preprint arXiv:1402.4304*.

Ji Ma, Tong Xiao, Jingbo Zhu, and Feiliang Ren. 2012. Easy-first chinese pos tagging and dependency parsing. In *COLING*, pages 1731–1746.

Mitchell P Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of english: The penn treebank. *Computational linguistics*, 19(2):313–330.

André Filipe Torres Martins. 2012. *The Geometry of Constrained Structured Prediction: Applications to Inference and Learning of Natural Language Syntax*. Ph.D. thesis, Columbia University.

Ryan McDonald, Koby Crammer, and Fernando Pereira. 2005. Online large-margin training of dependency parsers. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 91–98. Association for Computational Linguistics.

Michael T. Mordowanec, Nathan Schneider, Chris Dyer, and Noah A. Smith. 2014. Simplified dependency annotations with gfl-web. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. ACL.

Vinod Nair and Geoffrey E Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 807–814.

Jens Nilsson, Sebastian Riedel, and Deniz Yuret. 2007. The conll 2007 shared task on dependency parsing. In *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL*, pages 915–932. sn.

Joakim Nivre, Johan Hall, Jens Nilsson, Atanas Chanev, Gülsen Eryigit, Sandra Kübler, Svetoslav Marinov, and Erwin Marsi. 2007. Maltparser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering*, 13(2):95–135.

Kenji Sagae and Jun'ichi Tsujii. 2007. Dependency parsing and domain adaptation with lr models and parser ensembles. In *EMNLP-CoNLL*, volume 2007, pages 1044–1050.

Nathan Schneider, Brendan O'Connor, Naomi Saphra, David Bamman, Manaal Faruqui, Noah A Smith, Chris Dyer, and Jason Baldridge. 2013. A framework for (under) specifying dependency syntax without overloading annotators. *arXiv preprint arXiv:1306.2091*.

Kristina Toutanova and Christopher D Manning. 2000. Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In *Proceedings of the 2000 Joint SIGDAT conference on Empirical methods in natural language processing and very large corpora: held in conjunction with the 38th Annual Meeting of the Association for Computational Linguistics-Volume 13*, pages 63–70. Association for Computational Linguistics.

Huihsin Tseng, Pichuan Chang, Galen Andrew, Daniel Jurafsky, and Christopher Manning. 2005. A conditional random field word segmenter for sighan bakeoff 2005. In *Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing*, volume 171.

William Yang Wang, Kathryn Mazaitis, and William W Cohen. 2013. Programming with personalized pagerank: a locally groundable first-order probabilistic logic. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, pages 2129–2138. ACM.

William Yang Wang, Kathryn Mazaitis, Ni Lao, Tom Mitchell, and William W Cohen. 2014. Efficient inference and learning in a large knowledge base: Reasoning with extracted information using a locally groundable first-order probabilistic logic. *arXiv preprint arXiv:1404.3301*.

Fei Xia. 1999. Extracting tree adjoining grammars from bracketed corpora. In *Proceedings of the 5th Natural Language Processing Pacific Rim Symposium (NLPRS-99)*, pages 398–403.

Fan Yang, Yang Liu, Xiaohui Yu, and Min Yang. 2012. Automatic detection of rumor on sina weibo. In *Proceedings of the ACM SIGKDD Workshop on Mining Data Semantics*, page 13. ACM.

Kun Yu, Daisuke Kawahara, and Sadao Kurohashi. 2008. Chinese dependency parsing with large scale automatically constructed case structures. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pages 1049–1056. Association for Computational Linguistics.

Yue Zhang and Stephen Clark. 2008. A tale of two parsers: investigating and combining graph-based and transition-based dependency parsing using beam-search. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 562–571. Association for Computational Linguistics.

Meishan Zhang, Yue Zhang, Wanxiang Che, and Ting Liu. 2014. Character-level chinese dependency parsing. In *Proceedings of the 52th Annual Meeting of the Association for Computational Linguistics (ACL 2014)*, Baltimore, MD, USA, June. ACL.