11-712: NLP Lab Report

Dependency Parsing for Weibo: A Dependency Arc Inference Approach using Efficient First-Order Probabilistic Logic Programming

William Wang
ww@cmu.edu

May 2, 2014

Abstract

Dependency parsing is a core task in NLP, and it is a widely used by many applications such as information extraction, question answering, and machine translation. In general, the resources for Chinese dependency parsing are less accessible than English, and publicly available Chinese dependency parsers are still very limited. In this project, the goal is to build a Chinese dependency parser for Weibo — China's equivalent of Twitter. To do this, we formulate the dependeny parsing problem as many small and parallelizable structure prediction tasks: for each task, we use a locally groundable 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, and that leveraging lexical, syntactic, and distributional cues help improving the performance. In addition to this, we provide a publicly available GFL/FUDG-annotated Weibo treebank.

1 Basic Information about Chinese Dependency Parsing

Chinese dependency parsing has attracted many interests in the past decade. Bikel and Chiang (2000) and Chiang and Bikel (2002) are among the first to use Penn Chinese Tree Bank for dependency parsing, where they adapted Xia (1999)'s head rules. A few years later, the CoNLL shared task opened 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 (Carreras, 2007; Che et al., 2010; Duan et al., 2007; Nivre et al., 2007; Sagae and Tsujii, 2007; Zhang and Clark, 2008). In this work, we aim at building a new publicly available Chinese dependency parsing tool, using a new parsing algorithm that leverages techniques from the field of statistical relational learning.

2 Past Work on the Syntax of Chinese

Chao (1968) is among the first to study the syntax of Chinese. Unlike English, there has been long debate on the wordhood of Chinese (Duanmu, 1998). Chao and others' work investigate the free and bound forms, prosodic aspects (Shen, 1990), semantics Li (1972); Wu (1999), and morphological aspects (Dai, 1992; Sproat and Shih, 2002; Tang, 1989) to define the unit of word in Chinese. In addition, he has also studied the complex compound constructions (Zhang et al., 2000; Zhou et al.,

1999) in Chinese, as well as the parts of speech such as nouns and verbs (Krifka, 1995). More recently, Huang et al. (2009) have studied the lexical and functional categories, argument structure, and the verb phrase in Chinese. Moreover, they have discussed the more unique and challenging parts of syntax in Chinese: the passives, the *ba* construction, and the topic & relative constructions. Interestingly, they have also shed light on some advanced Chinese linguistic issues that have not been well studied in the past: questions, nominal expressions, and anaphora.

Even though there has been many interesting linguistics papers on various aspects of syntax in Chinese, the corresponding computational modeling work has been rather limited. One of the most popular computational tasks in Chinese NLP is word segmentation (Sproat and Emerson, 2003; Xue and Shen, 2003). where researchers have previously investigated sequencial models such as hierarchial hidden Markov model (Zhang et al., 2003), maximum entropy Markov model (Xue and Shen, 2003), and conditional random fields (Zhao et al., 2006) for this task. In addition to tokenization and segmentation, standard structure prediction tasks such as named entity recognition (Wu et al., 2005; Xue and Shen, 2003), part-of-speech tagging (Jiang et al., 2008; Ng and Low, 2004), and constituency parsing (Wang et al., 2006; Wu, 1997) have also been studied in the language-specific setups. As mentioned in Section 1, Chinese dependency parsing was first introduced by Bikel and Chiang (2000), and then became popular after the CoNLL multilingual shared tasks (Buchholz and Marsi, 2006; Nilsson et al., 2007).

In the past decade, there have been growing number of publicly availble Chinese language processing tools. ICTCLAS[1] is one of the most popular word segmentation tool in Chinese NLP. The Stanford Chinese NLP constituency parser (Levy and Manning, 2003), and the dependency parser (Chang et al., 2009) also provide insights for many Chinese NLP applications. More recently, more comphrehensive and Chinese-optimized toolkits were also made available (Che et al., 2010; Qiu et al., 2013). To the best of my knowledge, even though systems such as Malt parser (Nivre et al., 2007) provides solutions to multilingual dependency parsing, but they are not optimized for Chinese, and the accuracy on Penn Chinese Treebank is typically around 70% and lower 80%, which falls behind languages like English and German.

3 Available Resources

After some research and hands-on experiments on real data, I decided to use the open-source Stanford Word Segmenter[2] as the segmentation tool. Comparing to other popular Chinese word segmenters, the Stanford segmenter is well-maintained, and well-documented. The open-source Chinese lexicon I plan to use is also attached in the distribution of Stanford Chinese Segmenter: the Penn Tree Bank lexicon and the PKU lexicon. For the Chinese reference grammar, I am currently investigating the Stanford Dependencies[3], but I am also open to other suggestions.

Both of my test datasets (TestA and TestB) are from a subset of Wang Ling's μtopia Chinese microblog dataset[4]. This dataset is very challenging for dependency parsing: since pre-existing word segmentation and Chinese part-of-speech tagging tools are trained on standard Chinese news corpora, their perform poorly on Weibo data. In addition to this, since Chinese is a pro-drop and topical language, finding the correct head verb for a long Weibo post could be very difficult.

[1]http://sewm.pku.edu.cn/QA/reference/ICTCLAS/FreeICTCLAS/English.html
[2]http://nlp.stanford.edu/software/segmenter.shtml
[3]http://nlp.stanford.edu/software/stanford-dependencies.shtml
[4]http://www.cs.cmu.edu/ lingwang/microtopia/

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
for $i = 1 \rightarrow L$ tokens **do**
  $Y_i = \arg\max \vec{P_i}$;
end for

4 Survey of Phenomena in Chinese Dependency Parsing

Like I mentioned in the previous section, one notable difference between English and Chinese dependency parsing is the Chinese word segmentation issue, while both English and Chinese parser may also suffer from the issue of part-of-speech tagging errors (due to train/test domain/genre mismatch). However, despite these issues, there are still some interesting phenomena that mark the differences of the two languages:

- Function words are more frequently used in English than in Chinese. For example, when examining Wang Ling's parallel English-Chinese for the total counts of the word "the", there are 2,084 occurances in 2,003 sentences. Whereas in Chinese, there are only 52 occurances of the word "the" out of the 2,003 sentences.

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

- Another well-known issue in Chinese is that Chinese is a pro-drop language. This is extremely prominent in the short text. For example, in the Chinese Weibo data, I have observed the sentence "(If you) Want to eat Chicken, (you) Have to bear the sounds of chickens.".

5 Initial Design

I labeled Wang Ling's Weibo data with the help from many informant Lingpeng Kong. We have set up an online annotation environment[5] , using the FUDG (Schneider et al., 2013) and GFL annotation tool (Mordowanec et al.) introduced by Nathan Schneider and Michael Mordowanec. After discussing with my informant, we decide to start with the supervised learning approach. This means that we need to have annotation additional datasets for training. The training data has 2185 words, and the two test sets have 1020 and 1001 words respectively.

The main idea of the proposed algorithm is that, instead of solving the complex structure prediction problem as a whole, we decompose the problem into many parallelizable small local dependency arc inference task. The algorithm is outlined in Algorithm1. For each of the tokens in a sentence, we infer the parent token of this current token. To do this, we make use of a recently

[5] http://www.ark.cs.cmu.edu:7788/annotate

proposed probabilistic first-order logic inference algorithm — ProPPR (Wang et al., 2013). ProPPR is an efficient graph inference framework where the inference cost is independent of the size of the graph. To construct the search space for ProPPR, we extract lexical and syntactic (part-of-speech) cues, as well distributional information (e.g. adjacency or skip-n grams) to build the token-token relation database. Then, we use simple inference rules to map the database to construct the search space. In addition to this, we also need inference rules, such as the following joint adjacency-skip-gram rules:

```
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.

edge(V1,V2) :-
               skipone(V1,V2),hasword(V1,W1),
               hasword(V2,W2),keysoword(W1,W2) #skiponeword.

edge(V1,V2) :-
               skipone(V1,V2),haspos(V1,W1),
               haspos(V2,W2),keysopos(W1,W2) #skiponepos.

edge(V1,V2) :-
               skiptwo(V1,V2),hasword(V1,W1),
               hasword(V2,W2),keystword(W1,W2) #skiptwoword.

edge(V1,V2) :-
               skiptwo(V1,V2),haspos(V1,W1),
               haspos(V2,W2),keystpos(W1,W2) #skiptwopos.

edge(V1,V2) :-
               skipthree(V1,V2),hasword(V1,W1),
               hasword(V2,W2),keysthword(W1,W2) #skipthword.

edge(V1,V2) :-
               skipthree(V1,V2),haspos(V1,W1),
               haspos(V2,W2),keysthpos(W1,W2) #skipthpos.

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

keysoword(W1,W2) :- # ksow(W1,W2).
keysopos(W1,W2) :- # ksop(W1,W2).

keystword(W1,W2) :- # kstw(W1,W2).
keystpos(W1,W2) :- # kstp(W1,W2).
```

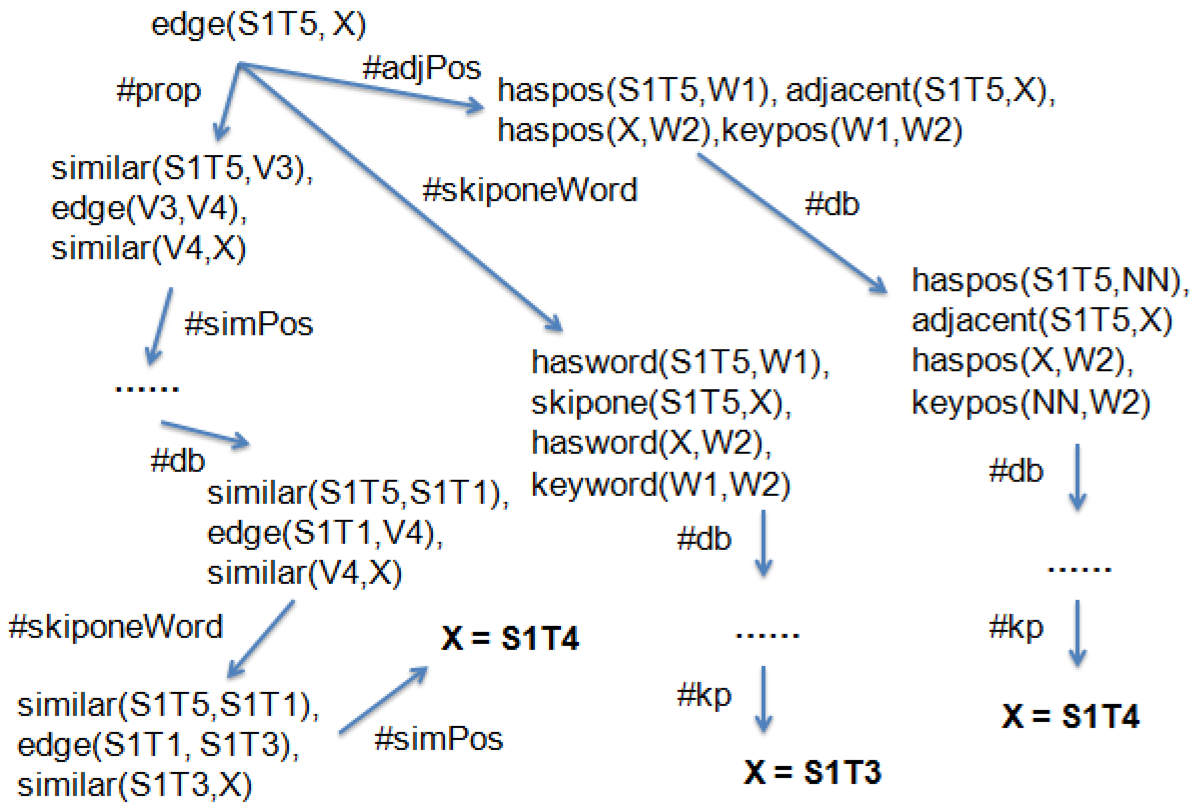

Figure 1: A search graph generated from the initially design of the first-order logic inference clauses.

```
keysthword(W1,W2) :- # ksthw(W1,W2).
keysthpos(W1,W2) :- # ksthp(W1,W2).
```

In the logic program above, we have a feature vector associated with each logic clause, and learning consists of tuning of the weights for each of the clause. Note that the last eight clauses are templated logic clause, where it capture the co-occurrence of two words/tags in a corpus. We use Backstrom and Leskovec (2011)'s supervised variant of personalized PageRank for learning, but with *tanh* edge strength function and a L_2-regularized log loss with parallel stochastic gradient descent. After mapping the entities from the database to the logic program, we can have the search graph in Fig. 1. We refer to our prior work Wang et al. (2013) for the detailed inference algorithm.

6 System Analysis on Corpus A

In the initial UAS result we obtain for corpus A, we observe that if using the simple adjacency rule, the UAS is 0.388. When using part-of-speech information, we observe a result of 0.438. When using the lexical information, the UAS is 0.477. We also show some of the learned parameters in Fig. 2.

```
kp(vv,ru)            23.1130        // verb's parent is the root
kw( “是” ,root)   9.61153       // word8 is the word "be"
kstp(nn,vv)          8.42095       // verb <word> <word> noun
ksop(vv,vv)          6.60828       // verb of sbar -> main verb
kp(vc,ru)            6.60577      // copula verb -> root
ksop(vv,ad)          3.30234      // adverb modifies verb
ksow( “微博” , “转发” )    1.74399
                                   // Weibo (noun) -> Retweet (verb)
```

Figure 2: Some top-ranked parameters learned from the initially design of the first-order logic inference clauses.

7 Lessons Learned and Revised Design

From the learned parameter in Fig. 2, we can see that our model definitely captures some interesting dependency structure from the data. For example, we see that in the first parameter, verb's parent is more likely to be the root. Also, from the second rule, the copula verb “be” in Chinese is also likely to be pointed to the root. In addition to that, we see that from the third parameter that a noun is likely to be pointed to a verb if there are two words in between them. To improve the performance of the parser, we now consider the similarity clauses:

```
# similarity

edge(V1,V2) :- similar(V1,V3),edge(V3,V4),similar(V4,V2)  #prop.
similar(V1,V2) :- samesent(V1,V2),hasword(V2,W),hasword (V1,W) #simword.
similar(V1,V2) :- samesent(V1,V2),haspos(V2,W),haspos(V1,W) #simpos.
similar(X,X) :- .
```

The similarity rule is extremely powerful: it allows us to perform believe propagation across similar parses. To constrain the search space, one can define intra-sentence similarity function such as the two above. However, it is entirely possible to relax this assumption, and consider inter-sentence similarity that explores similar parses across multiple sentences in the corpus.

8 System Analysis on Corpus B

On the corpus B, we observe that using the simple adjacency rule, we are able to obtain an UAS of 0.369. The part-of-speech rule itself can bring up the results to 0.480. Using the simple lexical features, the accuracy is 0.470. After adding the similarity clauses, we see that our system has obtained an UAS of 0.541 on the TestB dataset.

	TestA	TestB
Stanford (Xinhua Factored)	0.318	0.308
Stanford(Chinese Factored)	0.369	0.316
Adjacency	0.388	0.369
Part-of-speech	0.438	0.480
Word	0.477	0.470
Full (Adj+Pos+Word+Sim)	**0.530**	**0.541**

Figure 3: UAS results, comparing to an off-the-shelf Stanford Chinese dependency parser.

	wds/sec	sents/sec
Stanford (Chinese Factored) TestA	14.99	0.79
Stanford(Xinhua Factored) TestA	**83.90**	**4.44**
Full (Adj+Pos+Word+Sim) TestA	65.38	3.65
Stanford (Chinese Factored) TestB	18.96	1.02
Stanford(Xinhua Factored) TestB	**89.85**	**4.84**
Full (Adj+Pos+Word+Sim) TestB	64.41	3.60

Figure 4: Runtime speed, comparing to an off-the-shelf Stanford Chinese dependency parser.

9 Final Revisions

Using the newly added similarity clauses, we obtain the final results in the Fig. 3. We see that the final performance is much better than an off-the-shelf Stanford Chinese dependency parser. We also compare the runtime of our parser with the Stanford parser. The results are shown in the Fig. 4. We see that although Stanford parser with the compact Xinhua model runs faster, our model brings much better accuracy, without sacrificing much about the runtime.

10 Conclusion and Future Work

This work is among the first studies of dependency parsing for Weibo. We also propose a novel dependency arc prediction approach based on an efficient probabilistic programming language. Another contribution is that we provide the largest-ever FUDG-annotated GFL treebank (according to Lingpeng), for free. Empirically, we obtain some interesting results on the Weibo data. In the future, we plan to annotate more data, and explore more about joint intra-sentence and inter-sentence parsing.

References

Lars Backstrom and Jure Leskovec. 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, 2011.

Daniel M Bikel and David Chiang. 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, 2000.

Sabine Buchholz and Erwin Marsi. 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, 2006.

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

Pi-Chuan Chang, Huihsin Tseng, Dan Jurafsky, and Christopher D Manning. 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, 2009.

Yuen Ren Chao. *A grammar of spoken Chinese*. University of California Pr, 1968.

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

David Chiang and Daniel M. Bikel. 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, 2002. Association for Computational Linguistics. doi: 10.3115/1072228.1072354. URL `http://dx.doi.org/10.3115/1072228.1072354`.

Xiang-Ling Dai. *Chinese morphology and its interface with syntax*. PhD thesis, Ohio State University, 1992.

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

San Duanmu. Wordhood in chinese. *New approaches to Chinese word formation: Morphology, phonology and the lexicon in modern and ancient Chinese*, pages 135–196, 1998.

Cheng-Teh James Huang, Yen-hui Audrey Li, and Yafei Li. *The syntax of Chinese*. Cambridge University Press Cambridge, 2009.

Wenbin Jiang, Liang Huang, Qun Liu, and Yajuan Lü. A cascaded linear model for joint chinese word segmentation and part-of-speech tagging. In *In Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics*. Citeseer, 2008.

Manfred Krifka. Common nouns: A contrastive analysis of chinese and english. *The generic book*, pages 398–411, 1995.

Roger Levy and Christopher Manning. 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, 2003.

Charles Na Li. *Semantics and the structure of compounds in Chinese*. PhD thesis, University of California, Berkeley, 1972.

Michael T. Mordowanec, Nathan Schneider, Chris Dyer, and Noah A. Smith. Simplified dependency annotations with gfl-web.

Hwee Tou Ng and Jin Kiat Low. Chinese part-of-speech tagging: One-at-a-time or all-at-once? word-based or character-based? In *EMNLP*, pages 277–284, 2004.

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

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

Xipeng Qiu, Qi Zhang, and Xuanjing Huang. Fudannlp: A toolkit for chinese natural language processing. In *Proceedings of ACL*. Citeseer, 2013.

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

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

Xiao-nan Susan Shen. *The Prosody of Mandarin Chinese*, volume 118. University of California Pr, 1990.

Richard Sproat and Thomas Emerson. The first international chinese word segmentation bakeoff. In *Proceedings of the second SIGHAN workshop on Chinese language processing-Volume 17*, pages 133–143. Association for Computational Linguistics, 2003.

Richard Sproat and Chilin Shih. Corpus-based methods in chinese morphology. *Tutorial at the 19th COLING*, 2002.

Ting-Chi Tang. Studies on chinese morphology and syntax: 2. *Taipei: Student Book Co*, 1989.

Mengqiu Wang, Kenji Sagae, and Teruko Mitamura. A fast, accurate deterministic parser for chinese. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 425–432. Association for Computational Linguistics, 2006.

William Yang Wang, Kathryn Mazaitis, and William W Cohen. Programming with personalized pagerank: A locally groundable first-order probabilistic logic. *in Proceedings of the 22nd ACM International Conference on Information and Knowledge Management (CIKM 2013)*, 2013.

Dekai Wu. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational linguistics*, 23(3):377–403, 1997.

Jianxin Wu. *Syntax and semantics of quantification in Chinese*. PhD thesis, research directed by Dept. of Linguistics.University of Maryland, College Park, 1999.

Youzheng Wu, Jun Zhao, Bo Xu, and Hao Yu. Chinese named entity recognition based on multiple features. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 427–434. Association for Computational Linguistics, 2005.

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

Nianwen Xue and Libin Shen. Chinese word segmentation as lmr tagging. In *Proceedings of the second SIGHAN workshop on Chinese language processing-Volume 17*, pages 176–179. Association for Computational Linguistics, 2003.

Hua-Ping Zhang, Hong-Kui Yu, De-Yi Xiong, and Qun Liu. Hhmm-based chinese lexical analyzer ictclas. In *Proceedings of the second SIGHAN workshop on Chinese language processing-Volume 17*, pages 184–187. Association for Computational Linguistics, 2003.

Jian Zhang, Jianfeng Gao, and Ming Zhou. Extraction of chinese compound words: an experimental study on a very large corpus. 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 132–139. Association for Computational Linguistics, 2000.

Yue Zhang and Stephen Clark. 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, 2008.

Hai Zhao, Chang-Ning Huang, and Mu Li. An improved chinese word segmentation system with conditional random field. In *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing*, volume 1082117. Sydney: July, 2006.

Xiaolin Zhou, William Marslen-Wilson, Marcus Taft, and Hua Shu. Morphology, orthography, and phonology reading chinese compound words. *Language and Cognitive Processes*, 14(5-6):525–565, 1999.