Parsing Chinese Synthetic Words with a Character-based Dependency Model

Fei Cheng, Kevin Duh, Yuji Matsumoto

Graduate School of Information Science
Nara Institute of Science and Technology
8916-5 Takayama, Ikoma, Nara 630-0192, Japan
{fei-c, kevinduh, matsu}@is.naist.jp

Abstract

Synthetic word analysis is a potentially important but relatively unexplored problem in Chinese natural language processing. Two issues with the conventional pipeline methods involving word segmentation are (1) the lack of a common segmentation standard and (2) the poor segmentation performance on OOV words. These issues may be circumvented if we adopt the view of character-based parsing, providing both internal structures to synthetic words and global structure to sentences in a seamless fashion. However, the accuracy of synthetic word parsing is not yet satisfactory, due to the lack of research. In view of this, we propose and present experiments on several synthetic word parsers. Additionally, we demonstrate the usefulness of incorporating large unlabelled corpora and a dictionary for this task. Our parsers significantly outperform the baseline (a pipeline method).

Keywords: parsing, internal structure, synthetic word

1. Introduction

Word segmentation is considered as the fundamental step in Chinese natural language processing, since Chinese has no spaces between words to indicate word boundaries. In recent years, research in Chinese word segmentation has progressed significantly, with state-of-the-art performing at around 97% in precision and recall (Xue and others, 2003; Zhang and Clark, 2007; Li and Sun, 2009). But there still remain two crucial issues.

Issue 1: The lack of a common segmentation standard, due to the inherent difficulty in defining Chinese words, makes it difficult to share annotated resources. For instance, the synthetic word ”中国国际广播电台” (China Radio International) is considered to be one word in the MSRA corpus. While in the PKU corpus, it is segmented as ”中国 (China) / 国际(international) / 广播 (broadcast) / 电台 (station)”. Our parser, however, can offer flexible segmentation level output, by analysing the internal structure of words (Figure 1).

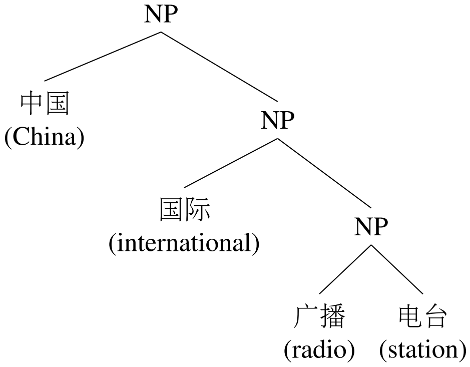


Figure 1: The Internal Structure of a Word

Issue 2: Frequent out-of-vocabulary (OOV) words lower the accuracy of word segmentation. Chinese words can be highly productive. For instance, Penn Chinese Treebank 6.0[1] does not contain the word ”成功者” (one that succeeds), even though the word ”成功” (succeed) and ”者” (person) appear hundreds of times. Li and Zhou (2012) defined such cases as pseudo-OOVs (i.e. words that are OOV but consisting of frequent internal parts) and estimated that over 60% of OOVs are pseudo OOVs in five common Chinese corpora. Goh et al. (2006) also claimed that most OOVs are proper nouns taking the form of Chinese synthetic words. These previous works suggest that analysing internal structures of Chinese synthetic words has the potential to improve the OOV problem.

Both issues can be better handled if we knew the internal information of Chinese words. We believe that parsing internal structures of Chinese synthetic words is an overlooked but potentially important task, which can benefit other Chinese NLP tasks.

However, correctly parsing Chinese synthetic words is challenging, not only because word segmentation step exists, but also for the reason that standard part-of-speech (POS) tags provide limited information. For instance, ”中国_NN / 国际_NN / 广播_NN / 电台_NN” contains a sequence of identical NN tags, giving little clue about their internal branching structure. Our work is concerned with parsing Chinese synthetic words into a parse tree without replying on POS tagging.

In this paper, we first introduce the classification of Chinese words in Section 2. Then we explain our annotation work and standard in Section 3. Section 4 describes the two types of character-based dependency models. The experiment setting and the comparison between different parsers and features are described in Section 5. Section 6 describes the recent related work. Finally we make the conclusion in Section 7.

2. Definition of Word in Chinese

It is generally considered that in Chinese, there isn’t a clear notion of **word**, unlike **character**. However, for native

[1] http://catalog.ldc.upenn.edu/LDC2007T36

Chinese speakers, a **word** is a lexical entry, representing a whole meaning. We adopt the classification of Chinese word proposed by (Lu et al., 2008), which divided Chinese words into the following two types.

Single-morpheme word: These words only have one morpheme inside them and cannot be segmented further. It means that the meanings of the individual parts do not indicate the meaning of the original word. The following are three subtypes of single-morpheme words.

- One-character single-morpheme word:
  人(human), 睡(sleep), 热(hot)
- Multi-character single-morpheme word:
  鹌鹑(quail), 鸳鸯(mandarin duck)
- Transliterated word: Those words are translated from foreign words based on the pronunciations.
  麦当劳(McDonald's), 瓦伦西亚(Valencia)

Synthetic word: These words are composed of two or more single-morpheme words and represent a new meaning which can be indicated from the internal constituents. The following is the internal structure of the synthetic word 总 / 工程 / 师 (chief engineer):

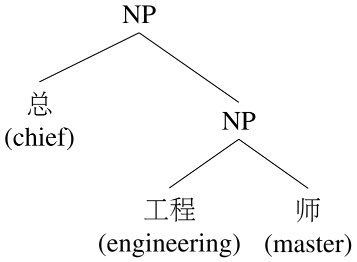


Figure 2: The Internal Structure of a Synthetic Word

Even if we don't know the word "总工程师", we can guess the meaning as 'chief engineer', based on the meanings of its internal parts.

In this paper, we treat synthetic words as our parsing target. The single-morpheme words are the smallest units (leaves) in our tree structure representation.

3. Annotation

A major challenge for synthetic word analysis is the lack of available annotated data. Therefore, we decided to annotate internal structures of Chinese synthetic words ourselves. We adopt a lexicon management system named Cradle (Lu, 2011) to annotate and represent tree structures.

3.1. Annotation Standard

We establish the following annotation standard based on the Chinese word definition given in Section 2.

- Determine whether the target word is a synthetic word or not. If it is a single-morpheme word, the annotator skips to the next word.
- Split the target word into parts on each level from top to bottom.
- Stop annotating until that all the split parts are single-morpheme words.

3.2. Annotation Data

The article titles of the Chinese Wikipedia is a rich resource of Chinese synthetic words. There are 826,557 article titles in our 2012 crawl of the Chinese Wikipedia. According to our annotation standard, four students randomly annotated 10,000 words[2] with the length distribution shown in Table 1. Each student's annotation is checked and revised by another student.

Length	Number of words
4	2292
5	1838
6	1516
7	1433
≥ 8	2922

Table 1: The Character Length Distribution of the Annotated Words. We exclude data with less than 4 characters because two or three character words contain very limited structure types.

For investigating the quality of our annotation, we required two of the students to annotate additional 200 words. We evaluate the annotation agreement in two levels. They first do word segmentation on the input words. Secondly, they annotate brackets on the gold segmented words. The Kappa-coefficient on the word boundary between characters in the first step is 0.947. The Kappa-coefficient on matching the brackets is 0.921.

4. Character-based Dependency Model

We now describe a character-based dependency model for predicting internal word structure. This model allows joint word segmentation and internal structure parsing. First, we introduce a label set (Table 2) to represent the morphological relations between two characters.

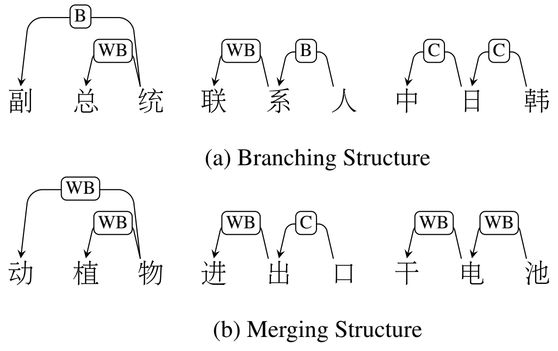
Label	Dependency relation
B	Branching relation
C	Coordinate relation
WB	Beginning inside a single-morpheme word
WI	Other part inside a single-morpheme word

Table 2: Character-level Morphological Relation

Although Chinese is a character-based language, it's ambiguous to decide the dependency direction between two characters inside the single-morpheme words. However, the majority of the dependency between parts in the Chinese synthetic words are left arcs. For instance, the synthetic word "工程师" (Engineer) is composed by "工程" (Engineering) and "师" (master) with a left arc. But we can hardly recognize the semantic head character inside the

[2] In this paper,we skipped all domain-specific words such as technical terms, judged by the annotators.

single-morpheme word ”工程”. In this paper, we do not distinguish the semantic modifier and head. For any dependency between two characters, we specify the right character as the head, which means only left arcs from the head to the modifier exist in our representation. More types of semantic relation and dependency relation information annotation is conceivable and we leave it as future work.



(a) Branching Structure

(b) Merging Structure

Figure 3: The Character-based Dependency Trees of Three-character Synthetic Words

Before discussing specific examples, we define two types of morphological structures. **Branching** is the most common morphological process in Chinese. The branching structures of a three character word ’ABC’ can be enumerated as ’AB + C’, ’A + BC’ and ’A + B + C’. **Merging** is another language phenomenon in Chinese. It means two semantically related words, which have an internal part in common, can merge into one word by removing one of the common parts. For instance, a word ’ABC’ can be composed by ’AB + AC’ sharing the common ’A’.

For demonstrating the usage of this label set, we present all possible structure types of three-character synthetic words into our character-based dependency representation (Figure 3). Three branching structure examples are listed in Figure 3a. ”副总统” (vice president) is a synthetic word composed by two single-morpheme words ”副” (vice) and ”总统” (president). ”联系人” (contact person) is a synthetic word composed by two single-morpheme words ”联系” (contact) and ”人” (person). ”中日韩” (China, Japan and Korea) is composed by three single-morpheme words ”中”, ”日” and ”韩” with coordinate relation between characters. We can also present the ’Merging’ type by our representation (Figure 3b). The example ”动植物” (animal and vegetation) is consisted by two single-morpheme words ”动物” (animal) and ”植物” (vegetation) sharing the common right character ”物” (object). The example ”进出口” (import and export) is consisted by two single-morpheme words ”进口” (import) and ”出口” (export) sharing the common left character ”口” (port). The example ”干电池” (dry cell) is consisted by two single-morpheme words ”干电” (dry power) and ”电池” (battery) sharing the common middle character ”电” (electricity).

The internal structure of a long synthetic word ’Olympic Games’ can be represented as a character-level dependency tree as shown in Figure 4. ”奥林匹克” with the labels ’WB’, ’WI’ and ’WI’ represent a single-morpheme transliterated word of ’Olympic’. ”运动会” (sports competition) is composed by two single-morpheme word ”运动” (sports) and ”会” (competition). There is a branching relation between ”奥林匹克” and ”运动会”.

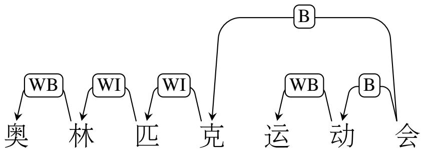


Figure 4: The Character-level Dependency Tree of a Long Synthetic Word.

4.1. Transition-based Parser

The transition-based parser is a step-wise approach to dependency parsing. In each step, the discriminative classifier uses a number of context features to check a node pair, the top node S_0 of a **stack** and the first node Q_0 in a **queue** (unprocessed sequence) to determine if a dependency should be established between them. Since only left arcs exist in our dependency representation, two actions are defined as follows:

> **Left-arc**: Add an arc from Q_0 to S_0 and pop S_0 from the **stack**.
>
> **Shift**: Push Q_0 into the **stack**.

The implementation of the transition-based model adopted in this work is the MaltParser (Malt) (Nivre et al., 2006), which uses support vector machines to learn transition actions.

4.2. Graph-based Parser

Graph-based dependency parser defines the score of a dependency graph as the sum of the scores of all the arcs $s(i, j, l)$ it contains. Here, $s(i, j, l)$ is the arc between words i and j with label l. This problem is equivalent to finding the highest scoring directed spanning tree in the complete graph over the input sentence. It is represented by:

$$G = \arg\max_{G=(V,A)} \sum_{(i,j,l)\in A} s(i,j,l) \qquad (1)$$

Second order sibling factorization (2nd-order) showed the significant improvement compared to first order parsing (McDonald, 2006; Carreras, 2007).

The implementation of the graph-based model adopted in this work is the MSTParser (MST) (McDonald, 2006), which uses standard structured learning techniques, globally setting parameters to maximize parsing performance on the training set.

4.3. Extra Features

As we mentioned, POS information is not sufficient as features for this task. Therefore we improve our parsers by incorporating features extracted from a large-scale corpus and a dictionary.

Dictionary feature: If a context character sequence exists in the NAIST Chinese dictionary[3] (with 129,560 entries), our parsers use its existing POS tags as features. It is possible for one word to correspond to multiple POS tags in the dictionary. The example entries in the dictionary are listed:

”稳定,22,22,2906,NN,*,*,稳定”
”稳定,44,44,3068,VV,*,*,稳定”

Brown cluster feature: Koo et al. (2008) trained a dependency parser in English and Czech and used Brown clusters (Brown et al., 1992) as an additional feature. We use CRF++[4] to implement a CRF-based model (Zhao et al., 2006) to do word segmentation on the Chinese Giga-word second edition[5]. Then we conduct a word-level Brown clustering on the segmented corpus. If a context character sequence exists in the word list of the segmented corpus, its corresponding cluster id is used as a feature.

Here, we demonstrate the extra features of a node in the following synthetic word:

中	国	国	际	广	播	电	台
Q_{-4}	Q_{-3}	Q_{-2}	Q_{-1}	Q_0	Q_1	Q_2	Q_3

For the node Q_0 (”广”), we search the context character sequences Q_0Q_1 (”广播”), $Q_{-1}Q_0$ (”际广”), $Q_0Q_1Q_2$ (”广播电”), $Q_{-1}Q_0Q_1$ (”际广播”), $Q_{-2}Q_{-1}Q_0$ (”国际广”), $Q_0Q_1Q_2Q_3$ (”广播电台”), $Q_{-1}Q_0Q_1Q_2$ (”际广播电”), $Q_{-2}Q_{-1}Q_0Q_1$ (”国际广播”), $Q_{-3}Q_{-2}Q_{-1}Q_0$ (”国国际广”) (with a four-character window) in the Brown clustering results and the NAIST Chinese dictionary. Then we use the corresponding clustering ids and POS tags of all these context character strings as the features of the node Q_0.

5. Experiments

5.1. Setting

Since we have a small size data with 10,000 annotated words, we use a 5-fold cross-validation to evaluate parsing performance. In each round, we split the data and used 80% of it as training set and the remaining 20% as testing set. For parameters tuning, we split the training set and used 80% of it as sub-training set and 20% as development set.

We adopt the feature templates in Table 3 for our Malt parser and add similar character sequence features into our MST and 2nd-order MST parsers. We found that the parsers reach the highest performance on the development set when the Brown cluster number is equal to 100.

As baseline, we implement a pipeline method which first uses our CRF-based model (Section 4.3.) to perform word segmentation, then uses MaltParser with Nivre arc-eager default feature[6] to perform word-level parsing. For the comparison to our character-based dependency parsers, we convert the word-level parsing results of the baseline to character-level based on our character-level morphological relation labels defined in Table 4 (Section 4.)

Category	Feature templates
Character	s_0, $s_0.h$, $s_0.lc$, s_{-1}, $s_{-1}.h$, $s_{-1}.lc$, i_0, i_1, i_2, i_3, i_4
Denpendency	$s_0.dep$, $s_0.h.dep$, $s_0.lc.dep$, $s_{-1}.dep$, $s_{-1}.h.dep$, $s_{-1}.lc.dep$
Denpendency + Character	$s_0.dep + s_0$, $s_0.h.dep + s_0.h$ $s_0.lc.dep + s_0.lc$, $s_{-1}.dep + s_{-1}$ $s_{-1}.h.dep + s_{-1}.h$, $s_{-1}.lc.dep + s_{-1}.lc$
Character Sequence	$s_{-1}.cseq$, $s_{-1}.cseq + s_0$, $s_{-1}.lc.cseq$, $s_{-1}.lc.cseq + s_0$, $s_{-1}.lc.cseq + i_0$, $s_{-1}.lc.cseq + s_0 + i_0$

Table 3: Feature Templates for the Malt Parser. s_0, s_{-1} denote the top and second characters in the stack. i_0, i_1 denote the first, second characters in the queue. dep, h and lc denote the dependency label, head and leftmost child of a character. $cseq$ denotes a character sequence started from a given character. $s_0.dep$ means dependency label of s_0. + denotes the combination of two or more features.

5.2. Results

In this section, we present the final results of our parsers and compare them to the baseline (Table 4). The evaluation metric of CoNLL 2006 shared task[7] is adopted, which includes unlabelled attachment score (UAS), unlabelled complete match (UCM), labelled attachment score (LAS) and labelled complete match (LCM). Note that we are parsing on synthetic words from Wikipedia titles (not sentences), so complete match refers to accuracy on these titles. The average length of titles is 7.04 characters.

	UAS	UCM	LAS	LCM
Baseline	83.04	41.23	80.57	35.04
Malt	93.63	72.23	90.93	66.54
MST	95.48	75.87	91.76	61.66
2nd-order MST	95.63	76.57	91.97	62.27
Malt+feats	95.45	76.13	93.51	70.63
MST+feats	95.73	76.45	93.54	67.44
2nd-order MST+feats	96.03	77.76	93.72	68.45

Table 4: Final Parsing Results. ’feats’ denote that the extra features are incorporated into the model.

In the upper part of Table 4, all the character-based dependency parsers highly outperform the baseline without using any extra features mentioned in Section 4.3. Graph-based MST and 2nd-order MST show obvious advantage on UAS, UCM and LAS and transition-based Malt gets the highest LCM.

In the second part, we incorporate extra features into parsing. Malt starts to reach comparable performance to MST and gets the highest LCM score 70.86. 2nd-order MST still leads the highest UAS, UCM and LAS scores. The extra features offer the improvement of 1.8 percentage points in UAS and 2.6 percentage points in LAS for Malt and 1.8

[3] http://cl.naist.jp/index.php?%B8%F8%B3%AB%A5%EA%A5%BD%A1%BC%A5%B9%2FNCD.

[4] http://crfpp.googlecode.com/svn/trunk/doc/index.html?source=navbar

[5] http://catalog.ldc.upenn.edu/LDC2005T14

[6] http://www.maltparser.org/userguide.html#featurespec

[7] http://ilk.uvt.nl/conll/

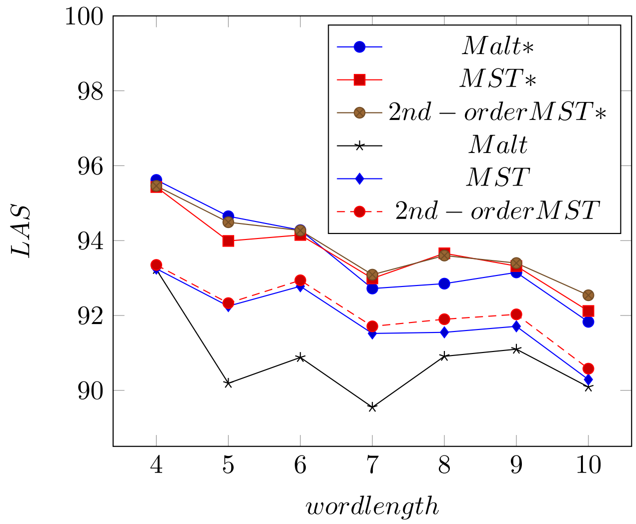


Figure 5: LAS Relative to Word Length, where '*' denotes that the parser incorporates the extra features.

percentage points in LAS for MST. It is thus clear that all the models have the capacity to learn from these features. The results demonstrate three points: (1) Our character-based dependency parsers outperform the baseline (the pipeline method). (2) Features extracted from a large-scale corpus and a dictionary can significantly improve parsing performance. (3) Graph-based parsers outperform transition-based parsers for this task.

5.3. Analysis

Figure 5 show LAS relative to word length of our character-based parsers. As expected, MST parsers outperform Malt over the entire range of word lengths. By incorporating extra features, Malt shows a strong tendency to improve LAS even outperforming the 2nd-order MST on the range of short word length. Graph-based MST and 2nd-order MST still keep higher LAS on the word length equal to or longer than 7 characters. All of the three models show very consistent increase in LAS on all word lengths compared to their base models. As expected, Malt, MST and 2nd-order MST finally reach similar LAS for short words and Malt degrade more rapidly with increasing word length because of error propagation (McDonald and Nivre, 2007).

6. Related work

Recently, some work on using the internal structure of words to improve Chinese process show promising results on different tasks. Li (2011) claimed the importance of word structures. They proposed a new paradigm for Chinese word segmentation in which not only flat word structures were identified but with internal structures were also parsed in a sentence. They aimed to integrate word structure information to improve the performance of word segmentation, parsing or other NLP tasks on sentences. Zhang et al. (2013) manually annotated the structures of 37,382 words, which cover the entire CTB5. They build a shift-reduce parser to jointly perform word segmentation, Part-of-speech tagging and phrase-structure parsing. Their system significantly outperform the state-of-art word-based pipeline methods on CTB5 test.

Our character-based word parsing model is inspired by the work of (Lu et al., 2008; Zhao, 2009). Lu et al. (2008) describe the semantic relations between characters. They proposed a structure analysis model for three-character Chinese words. Zhao (2009) presented a character-level unlabelled dependency scheme as an alternative to linear representation of sentences for word segmentation task. Their results demonstrate that the character-dependency framework can obtain comparable performance compared to the state-of-art word segmentation models. Our work extends previous works, focusing on parsing long words using various character-based dependency models. In addition, we extract the features from a large unlabelled corpus and a dictionary to improve our models. Our character-based parsing model for Chinese synthetic words can also help transform existing annotated Chinese corpora to give more fine-grained and consistent segmentations. For instance, the previous example ”中国国际广播电台” and ”中央 / 广播 / 电台” are inconsistently annotated in the corpus. Parsing the word into ”中国 / 国际 / 广播 / 电台” can make the corpus more consistent to benefit further NLP process.

7. Conclusion

In this paper, we claim that synthetic word parsing is an important but overlooked problem in Chinese NLP. Our first contribution is that we annotated 10,000 long Chinese synthetic words, which is potentially useful to other Chinese NLP tasks. The data is to be distributed as freely available data. Our second contribution is an in-depth comparison of various parsing frameworks and features. The results show that large unlabelled corpora and a dictionary can be extremely helpful in improving parsing performance. We believe that this is a first-step toward a more robust character-based processing of Chinese that does not require explicit word segmentation. As next work, we plan to include real syntactic dependency (subject-verb, verb-object or modifier-head) into our representation. We also plan to extend the algorithm and evaluation to the sentence-level and consider applications (such as Machine Translation, Information Retrieval) that may benefit from internal structure analysis of synthetic words.

8. References

Brown, Peter F, Desouza, Peter V, Mercer, Robert L, Pietra, Vincent J Della, and Lai, Jenifer C. (1992). Class-based n-gram models of natural language. *Computational linguistics*, 18(4):467–479.

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

Goh, Chooi-Ling, Asahara, Masayuki, and Matsumoto, Yuji. (2006). Machine learning-based methods to chinese unknown word detection and pos tag guessing. *Journal of Chinese Language and Computing*, 16(4):185–206.

Koo, Terry, Carreras, Xavier, and Collins, Michael. (2008). Simple semi-supervised dependency parsing.

Li, Zhongguo and Sun, Maosong. (2009). Punctuation as implicit annotations for chinese word segmentation. *Computational Linguistics*, 35(4):505–512.

Li, Zhongguo and Zhou, Guodong. (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.

Li, Zhongguo. (2011). Parsing the internal structure of words: A new paradigm for chinese word segmentation. In *ACL*, pages 1405–1414.

Lu, Jia, Asahara, Masayuki, and Matsumoto, Yuji. (2008). Analyzing chinese synthetic words with tree-based information and a survey on chinese morphologically derived words. In *IJCNLP*, pages 53–60.

Lu, jia. (2011). *Chinese Synthetic word Analysis using Large-scale N-gram and an Extendable lexicon Management System*. Ph.D. thesis, NAIST.

McDonald, Ryan T and Nivre, Joakim. (2007). Characterizing the errors of data-driven dependency parsing models. In *EMNLP-CoNLL*, pages 122–131.

McDonald, Ryan. (2006). *Discriminative learning and spanning tree algorithms for dependency parsing*. Ph.D. thesis, University of Pennsylvania.

Nivre, Joakim, Hall, Johan, and Nilsson, Jens. (2006). Labeled pseudo-projective dependency parsing with support vector machines. In *Proceedings of the Tenth Conference on Computational Natural Language Learning*, pages 221–225. Association for Computational Linguistics.

Xue, Nianwen et al. (2003). Chinese word segmentation as character tagging. *Computational Linguistics and Chinese Language Processing*, 8(1):29–48.

Zhang, Yue and Clark, Stephen. (2007). Chinese segmentation with a word-based perceptron algorithm. In *ANNUAL MEETING-ASSOCIATION FOR COMPUTATIONAL LINGUISTICS*, page 840.

Zhang, Meishan, Zhang, Yue, Che, Wanxiang, and Liu, Ting. (2013). Chinese parsing exploiting characters. In *51st Annual Meeting of the Association for Computational Linguistics*.

Zhao, Hai, Huang, Chang-Ning, and Li, Mu. (2006). 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.

Zhao, Hai. (2009). Character-level dependencies in chinese: Usefulness and learning. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 879–887. Association for Computational Linguistics.