Character-level Annotation for Chinese Surface-Syntactic Universal Dependencies

Chuanming Dong, Yixuan Li, Kim Gerdes

Character-level Annotation for Chinese Surface-Syntactic Universal Dependencies

Chuanming Dong
Institut National des
Langues et Civilisations
Orientales
dongchuanming@yahoo.com

Yixuan Li
Sorbonne Nouvelle
Lattice (CNRS)
yixuan.li@sorbonne-nouvelle.fr

Kim Gerdes
Sorbonne Nouvelle
Almanach (Inria)
LPP (CNRS)
kim.gerdes@sorbonne-nouvelle.fr

Abstract

This paper presents a new schema to annotate Chinese Treebanks on the character level. The original Universal Dependencies (UD) and Surface-Syntactic Universal Dependencies (SUD) projects provide token-level resources with rich morphosyntactic language details. However, without any commonly accepted word definition for Chinese, the dependency parsing always faces the dilemma of word segmentation. Therefore we present a character-level annotation schema integrated into the existing Universal Dependencies schema as an extension.

1 Introduction

With its writing system being a *Scriptua Continua*, Chinese is a language without explicit word delimiters and thus the "wordhood" is a particularly unclear notion. Yet, the vast majority of downstream NLP tasks for any language are based on "tokens", which mostly boils down to some kind of spelling-based tokenizer. Yet, in the case of Chinese, this step requires a preprocessing step called "word segmentation", whose performance has an non-neglectable influence on the final results. While the F-score of the segmentation task of general texts is in the high nineties since more than 10 years (Emerson 2005) and results have even been slightly improved by recent neural models (Chen et al. 2015, Cai & Zhao 2016), these numbers drop to below 10% for Out-of-Vocabulary terms, i.e. where the system has to take educated guesses on where the word borders are. This leads to catastrophic results for domain specialized texts that use a great number of neologisms unknown to the system, such as patent texts (Li & Gerdes 2019).

Since (Zhao 2009) proposed the first method for character-level dependencies parsing on the Chinese Penn Treebank, a series of research involving the character-based annotation (Li & Zhou 2012; Zhang & al. 2014; Li & al. 2018) have already shown the usefulness of the word-internal structures in Chinese syntactic parsing by obtaining limited but real improvements by means of extra character-level information (character POS, head character position and word internal dependency relation). (Zhao 2009) and (Zhang & al. 2013) have annotated a large-scale word list on Penn Treebank (PTB) and constituent Chinese Treebank (CTB) on the morphological level. Other character-based parsing attempts are generally based on these two annotated corpora.

In this work, we report on the integration of character-level annotations into the Chinese UD treebanks with the goal to find a joint segmentation-parsing method, which enables a multi-granularity analysis on Chinese sentences. Besides the final goal to improve the performance of the dependency parser with character-level information, in particular on out-of-domain texts, this work can also be regarded as a new Chinese word segmentation method: As we distinguish the morphological and syntactic relations between characters by a different set of dependency relation labels, we can ultimately fuse the character parsing results into a simple word segmentation, which can be compared to the original UD word segmentation. The character-level parse tree can thus also be projected onto a dependency tree on the words, which allows us to compare our parsing results with a simple token-based model.

In Section 2 we will briefly introduce various internal structures of Chinese words before presenting our annotation scheme for character-level POS and word internal dependency structures. The experiments and the results obtained are shown in Section 3, followed by the conclusion in Section 4.

2 Internal Dependency Structure of Chinese Words

Chinese words can be largely divided into two categories according to the number of morphemes contained:

1. simple words that contain only a single morpheme (monosyllabic (e.g.花, hua, '*flower*') or polysyllabic (e.g. 巧克力, qiao-ke-li, '*chocolate*'))
2. complex words that contains two or more morphemes.

Polysyllabic simple words are often words that have been directly transliterated form foreign languages and in which all characters have a semantically and syntactically equal status in the word formation. On the other hand, polysyllabic complex words, presenting the overwhelming majority of Chinese words, have more complex relations at the character-level and can also be divided into different subcategories. In the most widely accepted Chinese morphological theory (Feng 1997; Zhang 2003; Pan & al. 2004; Dong 2011), complex words are derivative words or compound words. The latter group includes five types: modifier-head type, coordinative type, predicate-object type, predicate-complement type, and subject-predicate type. In this work, without intention to give a theoretical definition of Chinese word, we aim to analyse the inner structure of already segmented words in UD treebanks.

In order to obtain these inter-character relations, we need to establish and apply syntactic tests that allow us to establish the head of a word based on distributional criteria. In this perspective, it is important to fit the new inter-character relations into a dependency tree that has been established based on similar distributional criteria. That is why our work is based on the Surface-Syntactic Universal Dependencies (SUD) variant of UD (Gerdes & al. 2018), which is an near-isomorphic but more surface syntactic alternative schema to UD with a more classical word distribution-based dependency structure that favors functional heads.

In this section, after an introduction of the different types of complex words in Chinese and their character-level dependency structure with examples (Section 2.1), we describe the three levels of our annotation scheme: determination of the head-daughter relations (the dependency structure), the type of the dependency relation, and the words' POS (Section 2.2).

2.1 Dependency Structure of Complex Words in SUD

In order to keep a clear distinction between word-based and character-based dependency relations, we use a set of specific labels starting with *m:* (standing for morphology) for the character-based relations. In the annotation schema, all under-word level structure in Chinese have an internal relation belonging to one of the four following extended morphological syntactic relations in SUD, which largely correspond to its original SUD syntactic relation types:

1. **m:mod** label given to head-modifier relations
   such as 中<**m:mod** 国 for 中国 zhong guo *center country* 'China'
2. **m:con**j label given to coordinative relations
   such as 自>**m:conj**己 for 自己 zi ji *self self* 'self'
3. **m:arg** label given to subject-predicate, e.g. 脸红 lian hong *face red* blush, predicate-object, e.g. 惊人 jing ren *suprise person* 'superising' and predicate-complement relations in which the complement is usually the result of the predicate, e.g. 减少 jian shao *minor less* 'reduce'
   such as 毕>**m:arg**业 for 毕业 bi ye *accomplish study* 'graduate'
4. **m:flat** label given to unheaded word constructions and to unknown kinds of relations, usually transliterated directely form foreign languages
   such as 巴>**m:flat**黎 for 巴黎 ba li *expect dawn* 'Paris'

For the position of the head in a word, we encounter three different categories of head directions (Zhang & al., 2013): left-headed, right-headed, and coordination (arbitrarily left-right, as in UD/SUD).

Another large category of complex words is made up of derivative words, i.e. usually consisting of the combination of a stem and an affix or the duplication of words. This category of words are analyzed by means of two different dependency relations (**m:mod** or **m:arg**) according to our annotation guidelines.

In this case, it can be hard to determine which character acts as head in the word. For this reason, we apply a series of syntactic tests to find the head: in (1a), it is obvious that the plural affix 们men does not change the syntactic distribution of the whole word and 我wo "me" should be considered as the head; in contrast, in (1b) the verbalizing affix 化hua this time changed the distribution from a nominal compound to a verbal compound. Thus we annotate (1a) with 我wo>**m:mod**们men and (1b) with 现代xiandai<**m:arg**化hua.[1] And here we categorise head-modifier and modifier-head relations in a single group as in UD treebanks the modifier can precede or postcede the head.

(1) a. 我 们
wo men
I,me plural
'we, us'

b. 现代 化
xiandai hua
modern -ize
'modernize'

In order to obtain a systematic and reproducible word-internal dependency analysis, our annotation guide uses a detailed decision tree, that cannot be reproduced here for lack of space. For example, for establishing consistent head-daughter relations, we apply the following tests: (1) Does the added character change the entire distribution? (2) Does the individual characters have the same POS as the whole word? (3) For a given character, can we find a complete paradigm of other words or characters that can occupy the character's position? (4) Is it possible to insert the character 的/地(de, genitive marker) into the word (for testing the modifier-head relation)? (5) Is it grammatically possible to inverse the characters in a word (for testing the coordinative relation)?

We finally annotated the 500 most frequent words in the Chinese SUD corpus, among which we count in total 71 left-headed words, 221 right-headed words and 198 coordinative words. For internal relations, we annotated 222 **m:mod**, 198 **m:conj**, 64 **m:arg**, and 16 **m:flat** relations. The degree of inter-annotator agreement over 100 words reached 88%.

For the remaining words of our corpus we provide an automatic character-based analysis by annotating them with the default left-right relation.

2.2 Statistics-based Character POS Annotation

In order to train a joint tagger-parser, we also need to have character-level POS annotation. To tag the part-of-speech of each character in a Chinese word, we make a list of all the multi-character words (except the polysyllabics which are often tagged as PROPN) in the SUD corpus sorted by frequency. Then, using a character POS dictionary, we insert into the list the character level POS for each word. In order to compare the word level POS and the character level POS, we also insert into the list the most frequent POS of each word. To construct the character level POS dictionary, we combine all the Chinese treebanks in the SUD project, forming a corpus of 299 895 words in total, and we apply the following strategy : If the character has appeared in this corpus as a single-character word, we simply select the most frequent POS of this character alone in the treebanks; on the other hand if the character appears only in multi-character words, we will select the most frequent POS of all the words that contain this character. However, since one character can have multiple POS in different words, the dictionary created by this method can cause plenty errors during the tagging. Therefore we manually

[1] The word 现代xiandai 'modern' is itself a compound word that can be analyzed as 现(xian, 'present') >**m:mod** 代(dai, 'era, generation'), giving the complete analysis (现xian>**m:mod**代dai)<**m:arg**化hua

corrected the character POS of the 1000 most frequent multi-character words in the dictionary. Here are some examples of what we obtain in our dictionary in **Table 1**.

FORM	POS:char1	POS:char2	POS:char3	POS:char4	...	POS:word	Frequency
电影 dian-ying '*film*'	NOUN	NOUN	-	-	...	(NOUN)	96
发展 fa-zhan '*developmen*t'	VERB	VERB	-	-	...	(VERB)	95
平方公里 ping-fang-gong-li '*square kilometer*'	NOUN	NOUN	NOUN	NOUN	...	(NOUN)	90

Table 1 Character POS Dictionary

To train the character level POS tagger, we divide the SUD Chinese corpus into 3 sets: a training set of 151 954 words, a developing set of 4 469 words, and a testing set of 4 232 words. We then convert these 3 sets of treebanks from word level to a character level by splitting all the complex words. And by using the dictionary that we obtain from the last step, we insert into these treebanks the character level POS, and we can thus train a POS tagger on the characters with these 3 sets using a proper deep learning algorithm such as LSTM. This approche give us a 91% accuracy of the characters POS tagging when we used the tagger of the Dozat parser (Dozat 2016) to train our character level tagger.

3 Experiments

We have worked on the four Chinese UD treebanks converted into SUD format and simplified characters when necessary: The Traditional Chinese Universal Dependencies Treebank annotated by Google (GSD), the Parallel Universal Dependencies treebanks created for the CoNLL 2017 shared task on Multilingual Parsing (PUD), the Traditional Chinese treebank of film subtitles and of legislative proceedings of Hong Kong (HK), and the essays written by learners of Mandarin Chinese as a foreign language (CFL), also proposed by the City University of Hong Kong.

To train the character-based POS tagger and SUD parser, we choose the Graph-based Neural Dependency Parser developed by Timothy Dozat at Stanford University for its character-based LSTM word representation. This parser contains a tagger training network and a dependency parser training network, but unfortunately these two training processes are separated, meaning that to obtain a corpus tagged and parsed, first we have to train a tagger, use it to tag our corpus, then train a parser and use it to parse our tagged corpus. Before the training process, we have also prepared a character vector file which is trained by BERT, a word embedding model developed by Google with a pre-trained character based Chinese model.

Our experiments consist of using Dozat Parser to train the word-based (WB) tagger and parser, as well as the character-based (CB) tagger and parser. Then by applying them to tag and parse our test corpus we can obtain two versions of our treebank: a word-based and a character-based treebank (see **Annex 2**), so that we can perform systematic tests of comparison on the combined Chinese SUD treebanks and evaluate the performance of our character-based tagger and parser. To sum up, we need to go through at least four training processes: WB tagger training, WB parser training, CB tagger training and CB parser training. Therefore, we have prepared our training data as following: for WB tagger and parser training, we extract the last 10% of the four former mentioned Chinese SUD treebanks to serve as the testing set and the developing set, and we combine the rest 90% to serve as a training set; for CB tagger and parser training, we carry out the exact same arrangement, except this time all the treebanks are converted from word level to character level.

Concerning the tagger, we compare the F-score of the tagger trained on WB and on CB. The **Table 2** display the direct result of the CB tagging.

Category	**Precision**	**Recall**	**F-score**
ADJ	89.37%	87.98%	88.67%
ADP	88.55%	81.38%	84.81%
ADV	89.33%	90.17%	89.75%
AUX	75.46%	89.96%	82.07%
CCONJ	95.92%	63.51%	76.42%
DET	89.36%	77.78%	83.17%
INTJ	66.67%	66.67%	66.67%
NOUN	93.20%	94.10%	93.65%
NUM	93.53%	100.00%	96.65%
PART	96.43%	96.72%	96.57%
PRON	96.06%	97.99%	97.01%
PROPN	73.37%	82.12%	77.50%
PUNCT	100.00%	100.00%	100.00%
SCONJ	0.00%	0.00%	0.00%
SYM	100.00%	100.00%	100.00%
VERB	92.22%	89.89%	91.04%
TOTAL	**91.99%**	**91.87%**	**91.93%**

Table 2 F-score of character level POS for our character-based tagg

As we can see, the Dozat parser achieved a rather high score on CB tagging. Some POS, because of it's absence on a character level, doesn't have a remarkable score, like SCONJ, but regardless of that we believe this tagger can satisfy our basic need in CB tagging. However, we can not compare directly this result with the result of WB tagging, since the words in the treebank for CB tagging has been split into characters, and thus we don't have the exact same number of POS in the WB and CB tagged treebanks. Therefore a recombination of the CB treebank after the tagging is necessary. To facilitate the recombination, we use the XPOS column in our treebank (under Conll-U format) to record the word level POS. When we split a word into characters during the preparation of treebanks for tagger training, we insert the character level POS into the UPOS column, and copy the word's original POS to the XPOS column of each character. And since in Dozat Parser the prediction of XPOS is dependent on the prediction of UPOS, we can thus train a tagger that can tag WB POS based on the CB POS. The following are the results of WB tagging (**Table 3**) and CB tagging after the recombination (**Table 4**)

Category	**Precision**	**Recall**	**F-score**
ADJ	65.69%	50.00%	56.78%
ADP	63.48%	69.75%	66.47%
ADV	80.08%	76.40%	78.20%
AUX	59.84%	81.56%	69.03%
CCONJ	92.68%	58.46%	71.70%
DET	96.81%	68.94%	80.53%
INTJ	100.00%	0.00%	0.00%
NOUN	88.17%	82.27%	85.12%
NUM	63.92%	98.41%	77.50%
PART	84.03%	91.74%	87.72%
PRON	94.06%	93.14%	93.60%
PROPN	38.17%	89.29%	53.48%
PUNCT	99.84%	99.84%	99.84%
SCONJ	100.00%	0.00%	0.00%
SYM	100.00%	0.00%	0.00%
VERB	76.29%	77.56%	76.92%
TOTAL	**81.85%**	**81.62%**	**81.74%**

Table 3 F-score of word level POS (UPOS) for our word-based tagger

Category	**Precision**	**Recall**	**F-score**
ADJ	65.52%	42.54%	51.58%
ADP	60.11%	87.90%	71.40%
ADV	75.00%	70.80%	72.84%
AUX	64.71%	86.03%	73.86%
CCONJ	92.68%	58.46%	71.70%
DET	91.22%	86.45%	88.77%
INTJ	100.00%	20.00%	33.33%
NOUN	77.87%	85.56%	81.54%
NUM	65.14%	93.65%	76.84%
PART	91.56%	94.50%	93.00%
PRON	92.47%	88.24%	90.30%
PROPN	54.05%	71.43%	61.54%
PUNCT	99.84%	100.00%	99.92%
SCONJ	20.00%	4.35%	7.14%
SYM	100.00%	100.00%	100.00%
VERB	83.31%	76.41%	79.71%
TOTAL	**88.85%**	**88.70%**	**88.78%**

Table 4 F-score of word level POS (XPOS) for our character-based tagger after the recombination

As we can see from these two tables above, the training on a character base has greatly improved the performance of the tagger. However for some most common POS, like ADJ and NOUN, there's an obvious decline of f-score. One of the possible reasons is that there's an inconsistency between the word level POS and character level POS in Chinese. For example, 活动 (NOUN, '*activity*') is composed by two verbal character "活" (VERB, '*living*') and "动" (VERB, '*moving*'). But by reviewing our data, we noticed that there's also an inconsistency on the POS annotation of the same words between different treebanks, even if in a similar context. This problem may have a bigger influence on both tagger and parser.

Concerning the parser, we have the usual UAS and LAS, but in addition the Orthogonal Label Unattached Score (OLS) that simply measures whether the word is connected to its governor with the right relation, independently whether the governor is correct (**Table 5**)

	WB	**CB**
UAS	78.96%	81.72%
OLS	81.29%	85.93%
LAS	66.65%	72.99%

Table 5 Comparison between the results of WB and CB parser

By comparing the UAS, OLS and LAS between the WB and CB parser, we can see that although the CB parser can correctly recognise more heads and dependency relations, the score is still relatively low, especially for the recognition of the dependency tree (LAS)

This is due to several possible reasons, including the incomplete character POS annotation and word structure annotation. Since we haven't totally finished the pretreatment process, there's a problem of inconsistency in our data, with the same word in the same context but having different POS or different internal structure annotated.

We can also separately measure the performance on the syntactic and morphological dependencies (**Table 6**). This method has a special function, that is the performance of the segmentation can be evaluated by concerning only about the two main groups of dependency relations: Morphe (relations annotated with m: at the beginning) and Deprel (the original dependency relations in SUD).

	Morph (Gold)	**Deprel (Gold)**	**TOTAL**
Morphe	2099	2	2101
Deprel	0	3128	3128
Wrong Head	4	1092	1096
TOTAL	2103	4222	6325

Table 6 Binary Confusion Matrix for Relations at Word/Character-level

The parsing error analysis has shown that the comparatively inferior recall scores for almost all types of relations are largely caused by the great quantity of false annotation of head-dependent arcs, while the morphe relations is the only one with a high recall (above 99%). Some relations with especially high head-dependency arc errors include clf, conj, dep, flat and punct. In contrast, the precision scores of most of dependency relations have passed 80% or close to it, with the exception of obl (62%, confusing with various types) and parataxis (47%, confusing frequentitly with comp) type relations. See **Annex 3** and **Annex 4** for more details about our evaluation data.

One possible reason behind these errors is the annotation error at previous tagger step, which also involve the dismatch of word POS annotations between different original Chinese Treebanks (e.g. the ordinal numbers are annotated as ADJ in certain corpus and as NUM in others). This the lack of equivalence may later lead the neural parser to some incorrect intuitions from statistics.

The f-score of the morphe relation is about 99.85% (**Table 6**) . The low annotation error (around 0.15%) shows an outstanding capability of the parser to distinguish character-level and word-level relations, and thus has the potential to serve as a decent word segmenter.

4 Conclusion

In this paper, we have presented a character-level annotation schema for modern Chinese and evaluated the state-of-the-art parser trained to annotate character level POS and dependency relations based on this schema. By comparing it with the word-based tagger and parser, we have witnessed a progress in the accuracy of this annotation system. However, after the evaluation we found out that the score for dependency tree annotation are not so satisfying. According to our error analysis, we conclude that there are mainly three reasons: incomplete and incorrect character level POS annotation, incomplete word structure annotation and discorrespondance in annotation between treebanks, all of them causing the irregularity of our data and thus confuse the algorithm to find the pattern. The solution is clear, by normalizing the data we can make further progress at improving the accuracy of our parser. Thus our next step is to establish formal annotation guidelines for this annotation schema in order to refine SUD treebanks so that them can be better adapted to our training system. Also, there's still room for improvement in our character POS annotation and word structure annotation, for example instead of using the most frequent POS for a single character and manually correct the faults, we can use deep learning algorithm to assign the most probable POS to a character judging by its context. And by accomplishing these two tasks we can provide our parser with a more powerful morphological support to achieve a more thorough syntactic analysis.

In spite of a less favorable score, these preliminary results show that it is actually possible to skip the word segmentation task and perform a joint segmentation and parsing. This has been shown to work on the existing Chinese UD dependency treebanks. We expect this to be useful for parsing texts with high rates of neologisms such as technological texts, but we will have to show that the joint parsing performance will not be too negatively affected itself by the unknown words. Yet, intuitively, it seems likely that the new words also show a systematic internal behavior and that many of the head-daughter relations can be correctly predicted because the individual characters have appeared elsewhere in the training corpus even if the combined word is new to the parser. Work is in progress to test this claim on Chinese patent texts.

We consider this work to be a step out of the hen-and-egg problem of tokenization and syntactic analysis: A parser needs tokens and a tokenizer needs syntactic information. Yet, a parser is an optimized tool to predict structure depending on the context. There is no reason that word-internal relations cannot be predicted in the same way as syntactic relations among words, even more so as many of these relations, in particular for compound words, actually correspond and behave very similarly to syntactic relations. This is an interesting result, not only for a scriptua continua on an isolating language such as Chinese but for other languages, too, where a morphological decomposition could be a successful basis for dependency parsing as long as the decomposition is linguistically well-grounded.

References

Cai D., Zhao H. 2016. Neural Word Segmentation Learning for Chinese. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (Vol. 1, pp. 409-420).

Chang S. 2003. On the Study of Compounds : A Contrastive Analysis of Chinese, English and Japanese. *In Proceedings of the 7th World Symposium On Chinese Language Teaching*. 張淑敏，〈漢英日複合詞的對

比分析：分類、結構與衍生〉，《第七屆世界華語文教學研討會論文集》，世界華語文教育學會，2003年12月。

Chen X., Qiu X., Zhu C., Liu P., Huang X. 2015. Long short-term memory neural networks for chinese word segmentation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 1197-1206.

Dong X. 2011. *Lexicalization: The Origin and Evolution of Chinese Disyllabic Words*. Sichuan: Sichuan Minorities Press.

Dozat T., Manning C. D. 2016. Deep Biaffine Attention for Neural Dependency Parsing. arXiv preprint arXiv:1611.01734.

Emerson T. 2005. The second international chinese word segmentation bakeoff. In *Proceedings of the fourth SIGHAN workshop on Chinese language Processing*.

Feng S. 1997. *Interactions between Prosody, Morphology and Syntax in Chinese*. Beijing: Peking University Press.

Gerdes K., Guillaume B., Kahane S., Perrier G. 2018. SUD or Surface-Syntactic Universal Dependencies: An annotation scheme near-isomorphic to UD. *In Proceedings of the Universal Dependencies Workshop* (UDW), EMNLP, Bruxelles.

Li H., Zhang Z., Ju Y., Zhao H. 2018. Neural character-level dependency parsing for Chinese. In The Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18).

Li Y., Gerdes K. 2019. In *Proceedings of the 13th TOTh International Conference* (TOTh 2019).

Li Z., Zhou G. 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 (EMNLP-CoNLL)*, 1445–1454.

Li Z. 2011. Parsing the internal structure of words: a new paradigm for Chinese word segmentation. In *Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL)*, 1405–1414.

Zhang M., Zhang Y., Che W., Liu T. 2013. Chinese Parsing Exploiting Characters. In *Proceedings of the 51th Annual Meeting of the Association for Computational Linguistics (ACL 2013)*.

Zhang M., Zhang Y., Che W., Liu T. 2014. Character-Level Chinese Dependency Parsing. *In Proceedings of the 52th Annual Meeting of the Association for Computational Linguistics (ACL 2014).*

Packard J.L. 2000. *The morphology of Chinese. A linguistic and cognitive approach*. Cambridge University Press, Cambridge.

Pan W., Ye B., Han Y. 2004. *The research on word formation in Chinese*. Shanghai: Huadong Shifan Daxue Chubanshe. 潘文国，叶步青,《汉语的构词法研究》，上海：华东师范大学出版社，2004。

Zhao H., Kit C., Song, Y. 2009. Character dependency tree based lexical and syntactic all-in-one parsing for chinese. In *The 10th Chinese National Conference on Computational Linguistics (CNCCL-2009)*, 82–88.

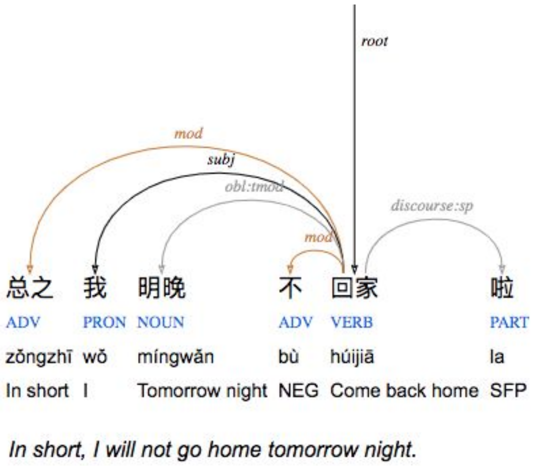
Zhao H. 2009. Character-level dependencies in Chinese: Usefulness and learning. *In Proceedings of the 12th Conference of the European Chapter of the ACL (EACL)*, 879–887.

Annex 1

The annotation of the head-dependency relation follows the CoNLL-U Format for UD and SUD (https://universaldependencies.org/format.html), in which every line for a single token including its annotation in 10 fields (ID, FORM, LEMMA, UPOS, XPOS, FEATS, HEAD, DEPREL, DEPS, MISC) separated by single tab characters. In our retokenized Chinese sentences, each line is devoted to a single character. Based on the dictionary of all Chinese words in the SUD corpus annotated with its head position and internal dependency relation type, we automatically integrate these character-level information into the converted CoNLL file with a Python script.
In the actual annotation process, we only indicate the index of the head character in the field of HEAD, as it is done for the syntactic dependencies.

Annex 2

And here is a comparison between the word-based (WB) treebank (Figure 1) and the character-based (CB) treebank (Figure 2) of the same sentence in Chinese.


root
mod
subj
obl:tmod
mod
discourse:sp
总之 我 明晚 不 回家 啦
ADV PRON NOUN ADV VERB PART
zǒngzhī wǒ míngwǎn bù húijiā la
In short I Tomorrow night NEG Come back home SFP
In short, I will not go home tomorrow night.

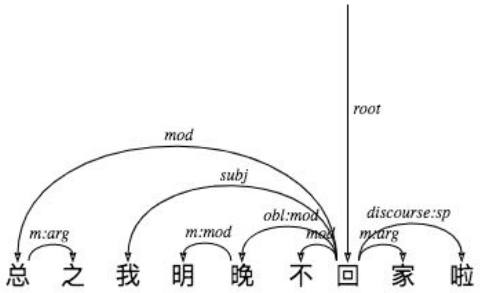
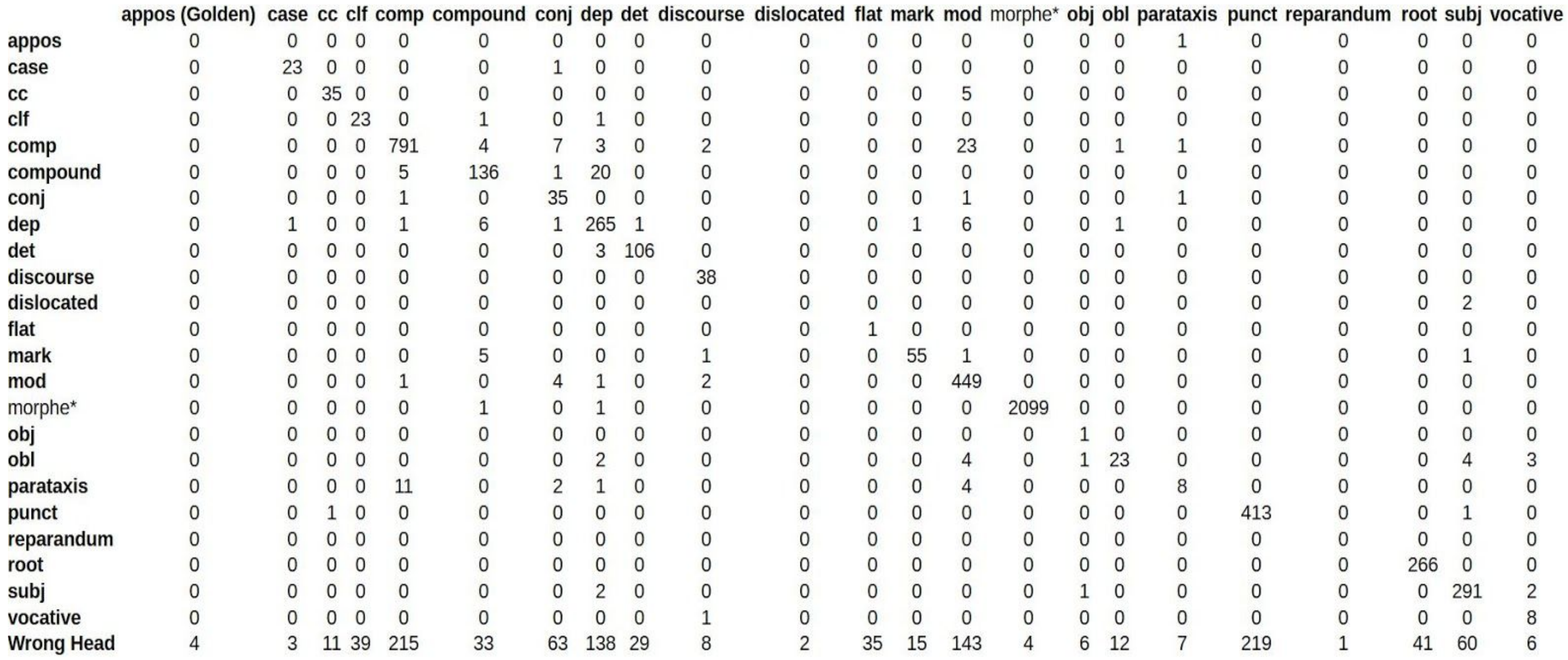


Figure 1 word-based treebank

Figure 2 character--based treebank

Annex 3

Confusion matrix of dependency relations annotated by our character-based parser



	appos (Golden)	case	cc	clf	comp	compound	conj	dep	det	discourse	dislocated	flat	mark	mod	morphe*	obj	obl	parataxis	punct	reparandum	root	subj	vocative
appos	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
case	0	23	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
cc	0	0	35	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0
clf	0	0	0	23	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
comp	0	0	0	0	791	4	7	3	0	2	0	0	0	23	0	0	1	1	0	0	0	0	0
compound	0	0	0	0	5	136	1	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
conj	0	0	0	0	1	0	35	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0
dep	0	1	0	0	1	6	1	265	1	0	0	0	1	6	0	0	1	0	0	0	0	0	0
det	0	0	0	0	0	0	0	3	106	0	0	0	0	0	0	0	0	0	0	0	0	0	0
discourse	0	0	0	0	0	0	0	0	0	38	0	0	0	0	0	0	0	0	0	0	0	0	0
dislocated	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0
flat	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
mark	0	0	0	0	0	5	0	0	0	1	0	0	55	1	0	0	0	0	0	0	0	1	0
mod	0	0	0	0	1	0	4	1	0	2	0	0	0	449	0	0	0	0	0	0	0	0	0
morphe*	0	0	0	0	0	1	0	1	0	0	0	0	0	0	2099	0	0	0	0	0	0	0	0
obj	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
obl	0	0	0	0	0	0	0	2	0	0	0	0	0	4	0	1	23	0	0	0	0	4	3
parataxis	0	0	0	0	11	0	2	1	0	0	0	0	0	4	0	0	0	8	0	0	0	0	0
punct	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	413	0	0	1	0
reparandum	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
root	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	266	0	0
subj	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	1	0	0	0	0	0	291	2
vocative	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	8
Wrong Head	4	3	11	39	215	33	63	138	29	8	2	35	15	143	4	6	12	7	219	1	41	60	6

Annex 4

Comparison between the parsing result of our word-based parser and character-based parser on several most frequent relations.

Category	**Precision**	**Recall**	**F-score**
case	89.66%	96.30%	92.86%
cc	70.31%	95.74%	81.08%
clf	89.71%	90.39%	90.04%
comp	80.82%	84.96%	82.83%
compound	66.67%	77.42%	71.64%
conj	56.04%	44.74%	49.76%
det	96.21%	93.38%	94.78%
discourse	93.62%	84.62%	88.89%
mark	76.71%	78.87%	77.78%
mod	90.71%	78.86%	84.37%
obl	45.10%	62.16%	52.27%
parataxis	5.13%	11.11%	7.02%
punct	99.53%	100.00%	99.76%
root	85.34%	85.34%	85.34%
subj	79.27%	84.12%	81.62%
vocative	100.00%	0.00%	0.00%
TOTAL	**81.41%**	**75.49%**	**78.33%**

Table 7 F-score of the most frequent dependency relations of the word-based parser

Category	**Precision**	**Recall**	**F-score**
case	85.19%	85.19%	85.19%
cc	73.77%	95.74%	83.33%
clf	91.94%	91.94%	91.94%
comp	78.74%	85.19%	81.84%
compound	62.93%	78.49%	69.86%
conj	62.32%	37.72%	46.99%
det	96.27%	94.85%	95.56%
discourse	97.78%	84.62%	90.72%
mark	71.43%	84.51%	77.42%
mod	90.94%	78.93%	84.51%
obl	62.00%	70.27%	65.88%
parataxis	47.02%	44.44%	45.69%
punct	99.68%	100.00%	99.84%
root	86.64%	86.64%	86.64%
subj	79.08%	86.35%	82.56%
vocative	81.82%	47.37%	60.00%
TOTAL	**83.67%**	**78.81%**	**81.17%**

Table 8 F-score of the most frequent dependency relations of the character-based parser after the recombination of characters