Non-Deterministic Segmentation for Chinese Lattice Parsing

Hai Hu
Indiana University
huhai@indiana.edu

Daniel Dakota
Indiana University
ddakota@indiana.edu

Sandra Kübler
Indiana University
skuebler@indiana.edu

Abstract

Parsing Chinese critically depends on correct word segmentation for the parser since incorrect segmentation inevitably causes incorrect parses. We investigate a pipeline approach to segmentation and parsing using word lattices as parser input. We compare CRF-based and lexicon-based approaches to word segmentation. Our results show that the lattice parser is capable of selecting the correction segmentation from thousands of options, thus drastically reducing the number of unparsed sentence. Lexicon-based parsing models have a better coverage than the CRF-based approach, but the many options are more difficult to handle. We reach our best result by using a lexicon from the n-best CRF analyses, combined with highly probable words.

1 Introduction

Many Asian languages, such as Chinese, Korean, and Burmese, do not mark word boundaries with spaces, in contrast to Indo-European languages such as English. Traditionally, parsing is preceded by word segmentation in a pipeline model. That is, the segmenter provides the most likely segmentation, which is subsequently passed to the parser, resulting in a propagation of errors from any initial incorrect segmentation. Previous work has demonstrated that performing segmentation and POS tagging jointly improves results (Ng and Low, 2004; Zhang and Clark, 2008; Forst and Fang, 2009), but results in a standard pipeline approach to segmentation and POS tagging have been mixed at best (Jiang et al., 2009).

The interaction between segmentation and unlexicalized constituent parsing for Chinese has not been fully explored. Whereas segmentation is performed on a character level, unlexicalized parsing is based on POS tags. Consequently, there can be a disconnect between the most likely character segmentation and the optimal POS sequence to fit the grammar. If the parser is given multiple segmentations from which to select, it is unclear how consistently and accurately it is able to select the correct segmentation combined with the correct POS sequence. One inherent difficulty is that the most probable segmentation may not actually be the optimal segmentation for the parser, particularly for an unlexicalized parser, since segmentation is done on the character level. Different segmentations may result in completely different sequences of POS tags, resulting in an alteration of the syntactic structure of the sentence that the parser must fit within its grammar.

We investigate a pipeline model where the segmenter provides n-best solutions, and the constituent parser decides on the best segmentation for POS tagging and parsing. I.e., we approach Chinese parsing as similar to morphologically rich languages (MRLs) of Hebrew and Arabic, in which lattice inputs have been used to provide the parser with options from which it chooses the best possible segmentation and morphological analysis. All experiments are based on the Penn Chinese Treebank `CTB5` (Xue et al., 2005).

The paper is structured as follows: We present an overview of related work in sec. 2 and a description of the non-deterministic segmenters in sec. 3. We discuss the experimental setup and results plus error analysis in sec. 4 and 5.

2 Related Work

2.1 Word Segmentation

Various approaches to word segmentation have been developed, often during the ACL-SIGHAN

Proceedings of Recent Advances in Natural Language Processing, pages 316–324,
Varna, Bulgaria, Sep 4–6 2017.
https://doi.org/10.26615/978-954-452-049-6_043

segmentation bake-offs (e.g. Sproat and Emerson, 2003; Emerson, 2005)[1]. In the bake-offs, variants of the maximum-length matching algorithm have traditionally been used to establish a baseline for segmentation (Levow, 2006), but more recent approaches have implemented various machine learning algorithms, treating word segmentation as a character sequence labeling task, where each character is given a tag that indicates the position of the character in a word (Xue, 2003; Tseng et al., 2005; Zhao and Kit, 2008, among others). Xue (2003) first employed a Maximum Entropy model to perform character labeling, with character unigrams and bigrams and previous labels as features. Later models also used other machine learning tools, most commonly Conditional Random Field (CRF) (e.g. Zhao et al., 2010; Qian and Liu, 2012). Common features include character types (Zhao et al., 2010), morphological information (Tseng et al., 2005), etc. Word-based F-measures for segmentation of state of the art systems are very high, ranging from 95% to 98%.

2.2 Chinese Parsing

Statistical parsing of Chinese has been approached in many different ways, yielding numerous systems, some Chinese specific. The highest achieved results, to our knowledge, on the Chinese treebank using standard PARSEVAL metrics is 86.6_F achieved by (Wang and Xue, 2014) using a joint POS tagging transition-based constituency parser that incorporates non-local and semi-supervised features using gold segmentation.

Qian and Liu (2012) use a joint system that is an extension of the CYK algorithm achieving 84.13_F using gold segmentation of words, 81.76_F in a pipeline, and 82.85_F for their joint system that includes: segmentation, POS tagging, and parsing. Brackets were only counted as correct if boundaries, label, and segmentation were correct, but this is not directly comparable to standard PARSEVAL metrics, but akin to CParseval (Harper and Huang, 2011).

Successful parsing in a pipeline hinges on the accuracy of the predicted segmentation. Unless the segmentation accuracy is almost 100% (99.9% as suggested by Sun (1999)), passing several segmentations to a downstream application may help resolve ambiguities. Forst and Fang (2009) showed that by applying non-deterministic segmentation and POS tagging, sentence level segmentation accuracy increases from 47.15% to 65.06%, and passing multiple analyses to an LFG parser increased the accuracy of parseable sentences.

Although Chinese lacks substantial morphology, the problem of identifying words is similar to the need to segment words into syntactic units in morphologically rich languages, which has improved parser performance (Tsarfaty, 2006). Lattice parsing (Chappelier et al., 1999) has been utilized in PCFG parsing; it allows the parser to determine the optimal path through all possible analyses to produce a tree (Goldberg and Tsarfaty, 2008). This technique has been applied to both Hebrew (Cohen and Smith, 2007) and Arabic (Green and Manning, 2010) with significant improvements noted for Hebrew, as well as to recover empty categories for both English and Chinese (Cai et al., 2011).

Directly related work by Wang et al. (2013) used the `blatt` parser, a modified PCFG-LA parser that allows a lattice input, in a pipeline approach. They concluded that non-weighted lattices are not effective for parsing Chinese. They developed a completely lattice-based system that uses a lattice to pass information between analyses (e.g. segmentation to POS tagging), improving results over standard pipeline approaches in all steps.

3 Non-Deterministic Segmentation

3.1 CRF Segmentation

We train a CRF model (crf++[2]) due to its ability to provide the n-best segmentations. We use a standard feature template (see Table 1). Character types are numbers, time (year, month, day, etc.), English letters, punctuation, and other Chinese characters. We use the 6-tag IOB scheme that performed best in a comparison by Zhao et al. (2010): S denotes a single-character word, B and E denote characters at the beginning and end of a multi-character word respectively. B2, B3 and M denote characters in the middle of a multi-character word. For example, the characters in 进出口|总值|达|一千零九十八点二亿|美元 (Eng.: The value of import and export reaches 109.82 billion USD.) are assigned the labels 'B B1 E|B E|S|B B1 B2 M M M M M E'.

[1] http://sighan.cs.uchicago.edu/

[2] http://taku910.github.io/crfpp/

	Features		
Unigram	C_{-1}	C_0	C_{+1}
Bigram	C_{-1} C_0	C_0 C_{+1}	C_{-1} C_{+1}
Char. type	type(C_{-1})	type(C_0)	type(C_{+1})

Table 1: CRF Features (C_{-1}: previous character, C_0: current character, C_{+1}: next character).

3.2 Lexicon-Based Segmentation

The second approach to segmentation is lexicon-based, using a Chinese word lexicon. Segmentation is approached as a search that finds all character sequences that occur in the dictionary, returning an unweighted lattice of all possible segmentations. We experiment with different types of input for the parser:

Upper bound: In order to investigate the feasibility of having the parser choose the correct segmentation from the lattice, we first use the lexicon extracted from the *test set*. This ensures full coverage with a minimal lexicon size, but is unrealistic.

Upper bound+Train: In this setting, we add the words from the training set.

Train$n$: In a more realistic setting, we extract the lexicon from the training set. Thus, the lexicon is incomplete with regard to the test set. We use heuristics to handle unknown words: For every unknown segment in a test sentence, we add the segment and a larger sequence of n (1–6) characters to the left and right to the lexicon. The maximal length of the context corresponds to the longest word in the training data. For example, if the sentence is 知识信息网络通讯技术和脱氧核糖核酸生物技术(Eng.: information and web technology and DNA biological technology) and the character 氧 is not present in the lexicon, we add 氧, 脱氧, 氧核, 和脱氧, 脱氧核 and 氧核糖 to the lexicon when $n = 3$. Here the unknown word 脱氧核糖核酸 (DNA) is of length 6, thus we cover this unknown word only when $n = 6$.

Train$n$+Names: Here, we add all person and geographic names, as well as number and time related words from the test data, as gazetteers are fairly easy to gather (c.f. e.g. Yu et al., 2008).

CRFn: Here we create a unique lexicon for each sentence by extracting all words from the n-best CRF analyses for that sentence ($1 < n < 5$).

We also experiment with extracting a lexicon from the CRF analyses for *all* test sentences:

CRF$n$lex: We extract these lexicons from the n-best analyses of the CRF segmenter ($1 < n < 5$).

CRF$n$lex+Train: We add all the words from the training data to CRF$n$lex to increase coverage.

CRF1lex+HiProb: We take advantage of the probability for any segmentation given by the CRF segmenter. Recall that the CRF segmenter provides the sentence probability for each of the n-best options. If the probability of a segmentation is greater than a threshold, we add all the words in that segmentation to the lexicon. By doing so, we add a range of word hypotheses that the CRF segmenter considers probable even though they may not appear in the best segmentation. Non-exhaustive experiments show that the probability threshold 0.30 yields a balance between adding new words to gain coverage and the parser's ability to select the correct segmentation. This setting results in 1901 words in the lexicon, and a reduction of unknown words.

CRF1lex+HiProb+Names+Single: We add names, numbers, times, and all single characters to the lexicon since some single-character words are not captured by the above lexicon. For long sentences (>50 segments) whose best segmentation has a probability <0.35, we extract words from all 5 segmentations. Note that we create individual lexicons for such low probability analyses, by adding a few words that are relevant for this specific sentence to the standard lexicon.

CRF1lex+HiProb+Names+Single+PKU: Since we still have unknown words, we additionally use the Peking University data (PKU) from the 2nd International Bakeoff in Chinese Word Segmentation (Emerson, 2005), which covers a broader lexicon and thus may increase coverage, but also increases the size of the lexicon, thus making the parser's task more difficult.

We add all words that only appear in the PKU data. Since the segmentation decisions differ between the PKU data and the CTB5, we use a simple filtering method to include only the words for which there is no annotation conflict. For example, the sequence 事实上(事实=fact, 上=grammatical particle, Eng.: in fact) occurs in the CRF analyses only as 事实|上, but in the test data, the only occurrence is segmented as one word, thus adding it from the PKU data reduces the number of unknown words.

System	F
Jiang et al. (2009)	97.58
Jiang et al. (2009) w/ adaptation	98.23
Qian and Liu (2012)	97.85
Zhang and Clark (2011)	97.78
Our CRF model	97.70

Table 2: CRF segmentation results for the 1-best setting.

n-best	# correct sent.	Coverage
1-best	258	74.14%
2-best	296	85.06%
3-best	306	87.93%
4-best	311	89.37%
5-best	318	91.38%

Table 3: Coverage of the CRF n-best analyses.

4 Experimental Setup

We extract the dictionaries and train the CRF model on the Penn Chinese Treebank (CTB5) (Xue et al., 2005), following the split of Qian and Liu (2012): sections 001–270 and 400–1151 for training, and sections 271-300 for testing. We evaluate segmentation using the official evaluation script from the 2nd International Bakeoff (Emerson, 2005). We report coverage and F-score. Coverage is defined as the percentage of sentences with the correct segmentation among the n-best solutions.

For parsing, CTB5 is preprocessed using standard procedures (Harper and Huang, 2011): Function labels are deleted, unary nodes are collapsed, and empty nodes are removed using the Berkeley Parser Analyser (Kummerfeld et al., 2013). We use the `blatt` parser (Goldberg and Elhadad, 2011), which is a reimplementation of the Berkeley parser (Petrov et al., 2006; Petrov and Klein, 2007), modified to allow lattice input. The parser uses a PCFG-LA (Matsuzaki et al., 2005; Petrov et al., 2006) iterative algorithm that splits each non-terminal category and determines if the split is beneficial. Splits deemed non-beneficial are then merged back together, and smoothing is performed over the non-terminals towards a common ancestor, calculating the EM after each sequence. We train four grammars using four different seeds (1–4) and report averages (unless otherwise noted), using the scorer from the 2013 SPMRL shared task (Seddah et al., 2013), a reimplementation of EVALB (Sekine and Collins, 1997) that allows for the penalization of unparsed sentences by scoring them as completely wrong.

5 Results

5.1 Segmentation Results

5.1.1 CRF Results

The results of our CRF segmenter are compared to other systems in table 2. Our results are similar to state-of-the-art systems, i.e., a CRF segmenter with simple features already works very well. However, table 3 shows that given an F-score of >97%, less than 75% of the test sentences are segmented completely correctly. As the CRF segmenter produces more segmentations, coverage increases to 91.38% given the 5-best analyses.

5.1.2 Dictionary Segmentation Results

Table 4 gives an overview of the coverage and lexicon size of the individual methods. The number of unknown words is the number of words from the gold segmentation of the test set that do not occur in the lexicon. Lexicon size refers to the number of words in the lexicon that occur in the test set. Both numbers give a more general view of the coverage of a lexicon. The training set based methods create much larger lexicons in comparison to the upper bound, but still have a low coverage of 67.53% even with the longest context (6). We reach 76.15% if we include names, etc. The CRF approach, which reaches a segmentation accuracy close to the state of the art (see Table 2), has a similarly low coverage of 74.14%. This shows that a high performance in segmentation does not directly translate into good parsing results. Interestingly, the lexicon extracted from the same 1-best CRF model performs better and reaches a coverage of 79.02%. If we use all 5 segmentations from the CRF to extract a lexicon, we reach a high coverage of 93.07%, at a lexicon size that is similar to the one extracted form the training data.

Combining the CRFn lexicons with the words from the training set gives a good coverage between 85.92% and 91.38%, but also increases the size of the lexicon considerably. Adding highly probable words from the CRF graph to CRF1lex improves coverage by about 8 points, but it does not reach the coverage of CRF5. Adding names and single segments to the lexicon increases coverage by >5.5% absolute. We reach the highest coverage of 94.83% by adding the PKU lexicon. Note that this lexicon only adds 200 words on average,

Lexicon	% coverage	# unk. words	Lex. size
Upper bound	100.00	0	1829
Upper bound+Train	100.00	0	2888
Train1	56.03	263	2755
Train2	58.62	242	2888
Train6	67.53	185	4142
Train1+Names	72.99	162	2785
Train2+Names	75.29	147	2825
Train6+Names	76.15	140	3216
CRF1	74.14	102	1872
CRF2	85.06	50	2164
CRF5	91.38	27	2872
CRF1lex	79.02	102	1872
CRF2lex	89.66	50	2164
CRF5lex	93.97	27	2872
CRF1lex+Train	85.92	57	2926
CRF2lex+Train	91.38	36	3094
CRF5lex+Train	94.83	22	3595
CRF1lex+HiProb	81.90	89	1901
CRF1lex+HiProb+Names+Single	87.64	50	2878
CRF1lex+HiProb+Names+Single+PKU	94.83	19	3028+

Table 4: Coverage of different segmentation methods.

	Wr. seg.	F	Rec.	Prec.
No Penalty				
Gold seg.	0	83.38	82.73	84.04
Upper bound	6.00	83.52	82.87	84.18
Upper+Train	41.25	84.81	84.23	85.39
With Penalty				
Upper bound	6.00	82.07	80.07	84.18
Upper+Train	41.25	75.02	66.90	85.39

Table 5: Initial parsing results

but decreases the number of unknown words by more than half.

5.2 Parsing Results

5.2.1 Initial Results

We establish an upper bound by using the gold segmentation of the *test sentences*, i.e., a deterministic input for the parser. We compare this to a setting using gold standard information, where we use the upper bound lexicon (based on gold segmentations of the test sentences), and a more realistic setting that extracts the lexicon from the combined training and test set. The results are shown in table 5. Note that the standard EVALB metric ignores sentences that have no parse or where the words in the parser output do not match the words in the gold standard. In our case, the latter translates into sentence where the parser did not choose the correct segmentation. We also present an analysis where both unparsed sentences and incorrectly segmented sentences are counted as completely incorrect, which is overly harsh. We address this issue in section 5.2.2.

The correct segmentation results in an F-score of 83.38. If we present the parser with the upper bound lexicon, the F-score increases minimally to 83.52. This means that the parser is capable of selecting the correct segmentation from the lattice in most cases. The increase in F is due to six incorrectly segmented sentences per grammar/seed, which are consequently ignored in the parser evaluation. Penalizing the parser (lower half of the table) for incorrectly segmented sentences results in a lower F-score of 82.07. When we use a lexicon based the upper bound+train, we achieve results of 84.81 and 75.02 respectively. Note that neither score is very informative. However, we do note that the number of incorrectly segmented sentences increases dramatically when we use a more realistic lexicon. We can conclude that the creation of the lexicon has a considerable influence on parsing quality: We need to provide good coverage without overwhelming the parser with too many segmentation possibilities.

5.2.2 Corrected Evaluation

Here, we have a closer look at how evaluation results are affected by either ignoring incorrectly

	F	Rec.	Prec.
Upper bound	85.14	84.68	85.60
Upper+penalty	76.23	68.71	85.60
Corrected	83.89	83.36	84.42

Table 6: Corrected results (seed 4).

segmented sentences or counting them as completely incorrect. We manually "correct" incorrectly segmented sentences by replacing the wrong tokens by the correct ones and deleting all nodes that cover these tokens in the parses. Thus, we keep the tree that is not affected by the incorrect segmentation but remove the affected part of the tree. As a consequence, recall should suffer from the wrong segmentations while precision should not be affected. This correction gives us a better picture of how incorrect segmentation affects results. Since it requires manual corrections, the analysis is based on a seed of 4, which results in four incorrectly segmented sentences.

Table 6 shows results for individual experiments and settings. Penalizing the parser for an incorrect segment is overly harsh given that the F-Score drops roughly 9% absolute for only four incorrect sentences. The results on the corrected set show higher results overall, i.e., the syntactic analyses for those sentences are mostly correct.

5.2.3 Parsing based on Realistic Segmentation

We have shown that the parser is able to select the correct segmentation with a high level of accuracy if it is present. Given that the gold lexicon is not representative of realistic data, we determine experimentally whether the parser can still perform at a consistently high accuracy with lexicons created from more realistic data. Results are shown in table 7. In the first setting, where we extract the lexicon directly from the training data and use a heuristic to cover unknown words, the parser has difficulties determining the correct segmentation, as evidenced by the high number of incorrectly segmented sentences. Thus, while the parsing results on correctly segmented sentences (no penalty) are high, theF-scores with the penalty are below 50. Adding names and time expressions reduces the number of wrong segmentations and increases the penalty F-scores by about 10 points. Longer contexts do not seem to be useful.

The CRF results show lower numbers of wrong segmentations and higher F-scores under penalty if we keep the number of lattices low. Creating a lexicon from the best CRF segmentation decreases the number of incorrectly segmented sentence to 82 and increases the F-score slightly. Using the n-best CRF analyses in any form is not useful. These analyses increase the number of wrong segmentations (to 191.75 for CRF5, to 205.25 for CRF5lex and CRF1lex+Train).

When we add the words from the training set to the CRF1 lexicon, we slightly increase the number of incorrectly segmented sentences, which decreased F-scores. Adding the highly probable words decreases the number of incorrectly segmented sentences to 74.50. Also adding names, times, and single characters to the lexicon decreases the number to 65.25, and adding the PKU lexicon reaches the lowest number of 55.25, along with the highest F-score with penalty: 70.69.

These results show clearly that simply increasing the coverage of our lexicon, and thus the input lattice of the parser, does not give us good segmentation and parsing performance. Using the 5-best CRF analyses, the lexicon based on those 5 analyses, and the combination with training words all result in good coverage, but provide unreliable information that does not allow the parser to choose the correct segmentation in many cases. However, adding words from highly likely analyses, and less reliable hypotheses only when necessary, gives the parser a good basis to make correct segmentation decisions. Adding the 200 words from the PKU lexicon helps in another 10 sentences. Thus, we can conclude that the parser is able to select correct segmentations if we have a lexicon that balances quality and good coverage.

5.2.4 Error Analysis

We performed an error analysis for the best setting (CRF1lex+HiProb+Names+Single+PKU), both on the segmentation and the syntax level, using the grammar based on seed 4.

Segmentation. There are 54 incorrectly segmented sentences. For 35 out of these, the correct segmentation is available in the lattice, but the parser did not select it. When analyzing these sentences, we found that in 32 cases, the parser selects a segmentation that has fewer words than the gold segmentation. I.e., the parser prefers analyses with fewer words. In some cases, the wrong segmentation makes sense linguistically, e.g., (NN 全文) (Eng.: full text) instead of the gold segmenta-

Setting		No penalty			With penalty		
	Wrong seg.	F	Rec.	Prec.	F	Rec.	Prec.
Train1	172.25	88.41	87.80	89.04	47.23	32.14	89.04
Train2	163.25	88.17	87.65	88.70	49.30	34.14	88.70
Train6	168.50	88.58	88.01	89.17	48.02	32.87	89.17
Train1+Name	117.25	87.87	87.56	88.18	57.68	42.87	88.18
Train2+Name	109.25	87.63	87.34	87.91	59.21	44.63	87.91
Train6+Name	116.25	87.90	87.61	88.19	57.91	43.12	88.19
CRF1	90.00	85.56	85.32	85.79	63.05	49.83	85.79
CRF2	108.25	85.74	85.05	86.44	61.02	47.16	86.44
CRF5	191.75	85.93	85.32	86.53	41.48	27.28	86.53
CRF1lex	82.00	85.95	85.69	86.21	65.21	52.44	86.21
CRF2lex	110.00	85.64	84.90	86.40	60.59	46.66	86.40
CRF5lex	205.25	86.38	85.75	87.01	35.56	22.34	87.01
CRF1lex+Train	86.50	86.25	85.87	86.63	64.63	51.54	86.63
CRF2lex+Train	117.25	85.66	84.91	86.42	58.73	44.48	86.42
CRF5lex+Train	205.25	86.38	85.75	87.01	35.56	22.34	87.01
CRF1lex+HiProb	74.50	85.84	85.55	86.13	66.51	54.17	86.13
CRF1lex+HiProb+Names+Single	65.25	85.63	85.36	85.91	68.58	57.07	85.91
CRF1lex+HiProb+Names+Single+PKU	55.25	85.73	85.24	86.23	70.69	59.90	86.23

Table 7: Parsing results for the different input lattices.

Error type	Count
NP → NP	20
non-NP → NP	20
non-NP → non-NP	10
NP → non-NP	15

Table 8: Top phrase errors in the best performing setting (CRF1lex+HiProb+Names+Single+PKU).

tion (DP (DT 全))(NP (NN 文)), or (NP (NN 交流会)) (Eng.: a meeting to exchange ideas) instead of (NN 交流) (NN 会).

Syntax. An analysis of the parses based on the upper bound lexicon shows that the most common mistakes made on the 344 correctly segmented sentences consists of frequently overgenerated nouns (NN), leading to NP-rich analyses. The same pattern can be found in the correctly segmented sentences from the best setting (CRF1lex+HiProb+Names+Single+PKU). The distribution of parsing errors is shown in table 8. The analysis shows that we have 20 errors of non-NP phrases becoming NPs. For example, a VV retagged as NN causes a VP to become an NP. We also find 20 cases where the parsed NP has the wrong structure.

6 Conclusion & Future Work

We have shown that a pipeline approach to Chinese parsing is feasible and beneficial, but it requires a carefully selected lexicon to guide the parser to make reliable segmentation choices. While lattices from a CRF segmenter with state-of-the-art performance do not allow the parser to select good segmentations, using a lexicon carefully extracted from the n-best CRF analyses gives the parser a good basis. The parser successfully selects the correct segmentation when given the option. The best performing lexicon consists of the 1-best CRF analyses, along with highly probable other analyses, names, dates, and words from the PKU corpus. A lexicon extracted from the CRF analyses has a higher coverage than using the corresponding analyses directly, but analyses beyond the best analysis have a detrimental effect on parsing, as the parser is biased towards its internal POS tag preferences, which may not correspond to the most probable segmentation.

We plan to extend our approach of creating individual lexicons per long sentence into a more general approach where the lexicon for each sentence is determined on an individual basis. We will also investigate the interaction of segmentation and parsing when grammatical functions are present. Preliminary experiments show that they can help resolve segmentation and POS tagging ambiguities, thus also increasing parsing accuracy.

Acknowledgments

H. Hu is funded by the China Scholarship Council.

References

Shu Cai, David Chiang, and Yoav Goldberg. 2011. Language-independent parsing with empty elements. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Portland, Oregon, pages 212–216.

Jean-Cédric Chappelier, Martin Rajman, Ramon Aragües, and Antoine Rozenknop. 1999. Lattice parsing for speech recognition. In *Sixth Conference sur le Traitement Automatique du Langage Naturel (TANL'99)*. Cargèse, France, pages 95–104.

Shay Cohen and Noah Smith. 2007. Joint morphological and syntactic disambiguation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Prague, Czech Republic, pages 208–217.

Thomas Emerson. 2005. The second international Chinese word segmentation bakeoff. In *Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing*. pages 123–133.

Martin Forst and Ji Fang. 2009. TBL-improved non-deterministic segmentation and POS tagging for a Chinese parser. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*. pages 264–272.

Yoav Goldberg and Michael Elhadad. 2011. Joint Hebrew segmentation and parsing using a PCFG-LA lattice parser. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Portland, Oregon, pages 704–709.

Yoav Goldberg and Reut Tsarfaty. 2008. A single generative model for joint morphological segmentation and syntactic parsing. In *46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Columbus, Ohio, pages 371–379.

Spence Green and Christopher Manning. 2010. Better Arabic parsing: Baselines, evaluations, and analysis. In *Proceedings of the 23rd International Conference on Computational Linguistics*. Beijing, China, pages 394–402.

Mary Harper and Zhongqiang Huang. 2011. Chinese statistical parsing. In Joseph Olive, Caitlin Christianson, and John McCary, editors, *Handbook of Natural Language Processing and Machine Translation*, Springer Publishing Company.

Wenbin Jiang, Liang Huang, and Qun Liu. 2009. Automatic adaptation of annotation standards: Chinese word segmentation and POS tagging: A case study. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*. Singapore, pages 522–530.

Jonathan K. Kummerfeld, Daniel Tse, James R. Curran, and Dan Klein. 2013. An empirical examination of challenges in Chinese parsing. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*. Sofia, Bulgaria, pages 98–103.

Gina-Anne Levow. 2006. The third international Chinese language processing bakeoff: Word segmentation and named entity recognition. In *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing*. Sydney, Australia, pages 108–117.

Takuya Matsuzaki, Yusuke Miyao, and Jun'ichi Tsujii. 2005. Probabilistic CFG with latent annotations. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*. Ann Arbor, Michigan, pages 75–82.

Hwee Tou Ng and Jin Kiat Low. 2004. Chinese part-of-speech tagging: One-at-a-time or all-at-once? Word-based or character-based? In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Barcelona, Spain, pages 277–284.

Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein. 2006. Learning accurate, compact, and interpretable tree annotation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*. Sydney, Australia, pages 433–440.

Slav Petrov and Dan Klein. 2007. Learning and inference for hierarchically split PCFGs. In *Proceedings of the National Conference on Artificial Intelligence*. Vancouver, Canada, pages 1663–1666.

Xian Qian and Yang Liu. 2012. Joint Chinese word segmentation, POS tagging and parsing. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Jeju Island, Korea, pages 501–511.

Djamé Seddah, Reut Tsarfaty, Sandra Kübler, Marie Candito, Jinho D. Choi, Richárd Farkas, Jennifer Foster, Iakes Goenaga, Koldo Gojenola Galletebeitia, Yoav Goldberg, Spence Green, Nizar Habash, Marco Kuhlmann, Wolfgang Maier, Joakim Nivre, Adam Przepiórkowski, Ryan Roth, Wolfgang Seeker, Yannick Versley, Veronika Vincze, Marcin Woliński, Alina Wróblewska, and Eric Villemonte de la Clergerie. 2013. Overview of the SPMRL 2013 shared task: A cross-framework evaluation of parsing morphologically rich languages. In *Proceedings of the Fourth Workshop on Statistical Parsing of Morphologically-Rich Languages*. Seattle, Washington, pages 146–182.

Satoshi Sekine and Michael Collins. 1997. Evalb bracket scoring program. http://nlp.cs.nyu.edu/evalb/.

Richard Sproat and Thomas Emerson. 2003. The first international Chinese word segmentation bakeoff. In *Proceedings of the Second SIGHAN Workshop on Chinese Language Processing*. Sapporo, Japan, pages 133–143.

Bin Sun. 1999. Methods for handling segmentation ambiguities. http://ccl.pku.edu.cn/doubtfire/NLP/Lexical_Analysis/Word_Segmentation_Tagging/Chinese_Word_Seg_Tag/seg_tag_BSWEN.htm. In Chinese.

Reut Tsarfaty. 2006. Integrated morphological an syntactic disambiguation for Modern Hebrew. In *Proceedings of the COLING/ACL 2006 Student Research Workshop*. Sydney, Australia, pages 49–54.

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*. Jeju Island, Korea, pages 32–39.

Zhiguo Wang and Nianwen Xue. 2014. Joint POS tagging and transition-based constituent parsing in Chinese with non-local features. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*. Baltimore, Maryland, pages 733–742.

Zhiguo Wang, Chengqing Zong, and Nianwen Xue. 2013. A lattice-based framework for joint Chinese word segmentation, POS tagging and parsing. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*. Sofia, Bulgaria, pages 623–627.

Naiwen Xue, Fei Xia, Fu-dong Chiou, and Marta Palmer. 2005. The Penn Chinese TreeBank: Phrase structure annotation of a large corpus. *Natural Language Engineering* 11(2):207–238.

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

Xiaofeng Yu, Wai Lam, Shing-Kit Chan, Yiu Kei Wu, and Bo Chen. 2008. Chinese NER using CRFs and logic for the fourth SIGHAN bakeoff. In *Proceedings of the Sixth SIGHAN Workshop on Chinese Language Processing*. Hyderabad, India, pages 102–105.

Yue Zhang and Stephen Clark. 2008. Joint word segmentation and POS tagging using a single perceptron. In *46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Columbus, Ohio, pages 888–896.

Yue Zhang and Stephen Clark. 2011. Syntactic processing using the generalized perceptron and beam search. *Computational Linguistics* 37(1):105–151.

Hai Zhao, Chang-Ning Huang, Mu Li, and Bao-Liang Lu. 2010. A unified character-based tagging framework for Chinese word segmentation. *ACM Transactions on Asian Language Information Processing (TALIP)* 9(2):5.

Hai Zhao and Chunyu Kit. 2008. Unsupervised segmentation helps supervised learning of character tagging for word segmentation and named entity recognition. In *Proceedings of the Third International Joint Conference on Natural Language Processing*. Hyderabad, India, pages 106–111.