# TBL-Improved Non-Deterministic Segmentation and POS Tagging for a Chinese Parser

**Martin Forst & Ji Fang**
Intelligent Systems Laboratory
Palo Alto Research Center
Palo Alto, CA 94304, USA
{mforst|fang}@parc.com

## Abstract

Although a lot of progress has been made recently in word segmentation and POS tagging for Chinese, the output of current state-of-the-art systems is too inaccurate to allow for syntactic analysis based on it. We present an experiment in improving the output of an off-the-shelf module that performs segmentation and tagging, the tokenizer-tagger from Beijing University (PKU). Our approach is based on transformation-based learning (TBL). Unlike in other TBL-based approaches to the problem, however, both obligatory and optional transformation rules are learned, so that the final system can output multiple segmentation and POS tagging analyses for a given input. By allowing for a small amount of ambiguity in the output of the tokenizer-tagger, we achieve a very considerable improvement in accuracy. Compared to the PKU tokenizer-tagger, we improve segmentation F-score from 94.18% to 96.74%, tagged word F-score from 84.63% to 92.44%, segmented sentence accuracy from 47.15% to 65.06% and tagged sentence accuracy from 14.07% to 31.47%.

## 1 Introduction

Word segmentation and tagging are the necessary initial steps for almost any language processing system, and Chinese parsers are no exception. However, automatic Chinese word segmentation and tagging has been recognized as a very difficult task (Sproat and Emerson, 2003), for the following reasons:

First, Chinese text provides few cues for word boundaries (Xia, 2000; Wu, 2003) and part-of-speech (POS) information. With the exception of punctuation marks, Chinese does not have word delimiters such as the whitespace used in English text, and unlike other languages without whitespaces such as Japanese, Chinese lacks morphological inflections that could provide cues for word boundaries and POS information. In fact, the lack of word boundary marks and morphological inflection contributes not only to mistakes in machine processing of Chinese; it has also been identified as a factor for parsing miscues in Chinese children's reading behavior (Chang et al., 1992).

Second, in addition to the two problems described above, segmentation and tagging also suffer from the fact that the notion of a word is very unclear in Chinese (Xu, 1997; Packard, 2000; Hsu, 2002). While the word is an intuitive and salient notion in English, it is by no means a clear notion in Chinese. Instead, for historical reasons, the intuitive and clear notion in Chinese language and culture is the character rather than the word. Classical Chinese is in general monosyllabic, with each syllable corresponding to an independent morpheme that can be visually rendered with a written character. In other words, characters did represent the basic syntactic unit in Classical Chinese, and thus became the sociologically intuitive notion. However, although colloquial Chinese quickly evolved throughout Chinese history to be disyllabic or multi-syllabic, monosyllabic Classical Chinese has been considered more elegant and proper and was commonly used in written text until the early 20th century in China. Even in Modern Chinese written text, Classical Chinese elements are not rare. Consequently, even if a morpheme represented by a character is no

*Proceedings of the 12th Conference of the European Chapter of the ACL*, pages 264–272,
Athens, Greece, 30 March – 3 April 2009.

longer used independently in Modern colloquial Chinese, it might still appear to be a free morpheme in modern written text, because it contains Classical Chinese elements. This fact leads to a phenomenon in which Chinese speakers have difficulty differentiating whether a character represents a bound or free morpheme, which in turn affects their judgment regarding where the word boundaries should be. As pointed out by Hoosain (Hoosain, 1992), the varying knowledge of Classical Chinese among native Chinese speakers in fact affects their judgments about what is or is not a word. In summary, due to the influence of Classical Chinese, the notion of a word and the boundary between a bound and free morpheme is very unclear for Chinese speakers, which in turn leads to a fuzzy perception of where word boundaries should be.

Consequently, automatic segmentation and tagging in Chinese faces a serious challenge from prevalent ambiguities. For example [1], the string “有意见” can be segmented as (1a) or (1b), depending on the context.

| | | | |
|---|---|---|---|
| (1) | a. | 有 | 意见 |
| | | yǒu | yìjian |
| | | have | disagreement |
| | b. | 有意 | 见 |
| | | yǒuyì | jiàn |
| | | have the intention | meet |

The contrast shown in (2) illustrates that even a string that is not ambiguous in terms of segmentation can still be ambiguous in terms of tagging.

| | | | |
|---|---|---|---|
| (2) | a. | 白/a | 花/n |
| | | bái | huā |
| | | white | flower |
| | b. | 白/d | 花/v |
| | | bái | huā |
| | | in vain | spend |
| | | ‘spend (money, time, energy etc.) in vain’ | |

Even Chinese speakers cannot resolve such ambiguities without using further information from a bigger context, which suggests that resolving segmentation and tagging ambiguities probably should not be a task or goal at the word level. Instead, we should preserve such ambiguities in this level and leave them to be resolved in a later stage, when more information is available.

To summarize, the word as a notion and hence word boundaries are very unclear; segmentation and tagging are prevalently ambiguous in Chinese. These facts suggest that Chinese segmentation and part-of-speech identification are probably inherently non-deterministic at the word level. However most of the current segmentation and/or tagging systems output a single result.

While a deterministic approach to Chinese segmentation and POS tagging might be appropriate and necessary for certain tasks or applications, it has been shown to suffer from a problem of low accuracy. As pointed out by Yu (Yu et al., 2004), although the segmentation and tagging accuracy for certain types of text can reach as high as 95%, the accuracy for open domain text is only slightly higher than 80%. Furthermore, Chinese segmentation (SIGHAN) bakeoff results also show that the performance of the Chinese segmentation systems has not improved a whole lot since 2003. This fact also indicates that deterministic approaches to Chinese segmentation have hit a bottleneck in terms of accuracy.

The system for which we improved the output of the Beijing tokenizer-tagger is a hand-crafted Chinese grammar. For such a system, as probably for any parsing system that presupposes segmented (and tagged) input, the accuracy of the segmentation and POS tagging analyses is critical. However, as described in detail in the following section, even current state-of-art systems cannot provide satisfactory results for our application. Based on the experiments presented in section 3, we believe that a proper amount of non-deterministic results can significantly improve the Chinese segmentation and tagging accuracy, which in turn improves the performance of the grammar.

## 2 Background

The improved tokenizer-tagger we developed is part of a larger system, namely a deep Chinese grammar (Fang and King, 2007). The system is hybrid in that it uses probability estimates for parse pruning (and it is planned to use trained weights for parse ranking), but the “core” grammar is rule-based. It is written within the framework of Lexical Functional Grammar (LFG) and implemented on the XLE system (Crouch et al., 2006; Maxwell and Kaplan, 1996). The input to our system is a raw Chinese string such as (3).

[1] (1) and (2) are cited from (Fang and King, 2007)

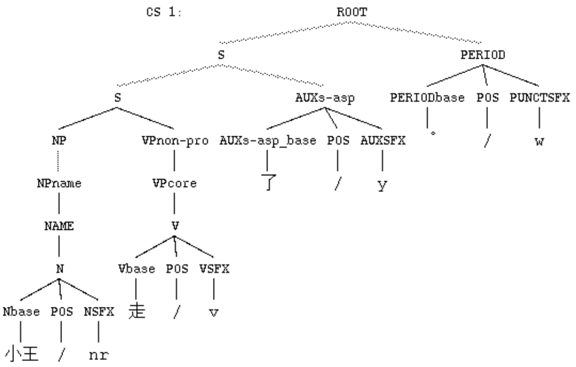
(3)

| | | | |
|---|---|---|---|
| 小王 | 走 | 了 | 。 |
| xiǎowáng | zǒu | le | . |
| XiaoWang | leave | ASP [2] | . |

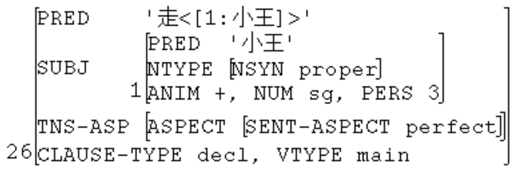
‘XiaoWang left.’

The output of the Chinese LFG consists of a Constituent Structure (c-structure) and a Functional Structure (f-structure) for each sentence. While c-structure represents phrasal structure and linear word order, f-structure represents various functional relations between parts of sentences. For example, (4) and (5) are the c-structure and f-structure that the grammar produces for (3). Both c-structure and f-structure information are carried in syntactic rules in the grammar.

(4) c-structure of (3)



(5) f-structure of (3)



```
  PRED      '走<[1:小王]>'
            [PRED   '小王'
  SUBJ       NTYPE [NSYN proper]
           1 ANIM +, NUM sg, PERS 3]
  TNS-ASP [ASPECT [SENT-ASPECT perfect]]
26 CLAUSE-TYPE decl, VTYPE main
```

To parse a sentence, the Chinese LFG minimally requires three components: a tokenizer-tagger, a lexicon, and syntactic rules. The tokenizer-tagger that is currently used in the grammar is developed by Beijing University (PKU)[3] and is incorporated as a library transducer (Crouch et al., 2006).

Because the grammar’s syntactic rules are applied based upon the results produced by the tokenizer-tagger, the performance of the latter is critical to overall quality of the system’s output. However, even though PKU’s tokenizer-tagger is one of the state-of-art systems, its performance is not satisfactory for the Chinese LFG. This becomes clear from a small-scale evaluation in which the system was tested on a set of 101 gold sentences chosen from the Chinese Treebank 5 (CTB5) (Xue et al., 2002; Xue et al., 2005). These 101 sentences are 10-20 words long and all of them are chosen from Xinhua sources [4]. Based on the deterministic segmentation and tagging results produced by PKU’s tokenizer-tagger, the Chinese LFG can only parse 80 out of the 101 sentences. Among the 80 sentences that are parsed, 66 received full parses and 14 received fragmented parses. Among the 21 completely failed sentences, 20 sentences failed due to segmentation and tagging mistakes.

This simple test shows that in order for the deep Chinese grammar to be practically useful, the performance of the tokenizer-tagger must be improved. One way to improve the segmentation and tagging accuracy is to allow non-deterministic segmentation and tagging for Chinese for the reasons stated in Section 1. Therefore, our goal is to find a way to transform PKU’s tokenizer-tagger into a system that produces a proper amount of non-deterministic segmentation and tagging results, one that can significantly improve the system’s accuracy without a substantial sacrifice in terms of efficiency. Our approach is described in the following section.

## 3 FST[5] Rules for the Improvement of Segmentation and Tagging Output

For grammars of other languages implemented on the XLE grammar development platform, the input is usually preprocessed by a cascade of generally non-deterministic finite state transducers that perform tokenization, morphological analysis etc. Since word segmentation and POS tagging are such hard problems in Chinese, this traditional setup is not an option for the Chinese grammar. However, finite state rules seem a quite natural approach to improving in XLE the output of a sep-

[2] ASP stands for aspect marker.

[3] http://www.icl.pku.edu.cn/icl_res/

[4] The reason why only sentences from Xinhua sources were chosen is because the version of PKU’s tokenizer-tagger that was integrated into the system was not designed to handle data from Hong Kong and Taiwan.

[5] We use the abbreviation “FST” for “finite-state transducer”. *fst* is used to refer to the finite-state tool called *fst*, which was developed by Beesley and Karttunen (2003).

arate segmentation and POS tagging module like PKU's tokenizer-tagger.

### 3.1 Hand-Crafted FST Rules for Concept Proving

Although the grammar developer had identified PKU's tokenizer-tagger as the most suitable for the preprocessing of Chinese raw text that is to be parsed with the Chinese LFG, she noticed in the process of development that (i) certain segmentation and/or tagging decisions taken by the tokenizer-tagger systematically go counter her morphosyntactic judgment and that (ii) the tokenizer-tagger (as any software of its kind) makes mistakes. She therefore decided to develop a set of finite-state rules that transform the output of the module; a set of mostly obligatory rewrite rules adapts the POS-tagged word sequence to the grammar's standard, and another set of mostly optional rules tries to offer alternative segment and tag sequences for sequences that are frequently processed erroneously by PKU's tokenizer-tagger.

Given the absence of data segmented and tagged according to the standard the LFG grammar developer desired, the technique of hand-crafting FST rules to postprocess the output of PKU's tokenizer-tagger worked surprisingly well. Recall that based on the deterministic segmentation and tagging results produced by PKU's tokenizer-tagger, our system can only parse 80 out of the 101 sentences, and among the 21 completely failed sentences, 20 sentences failed due to segmentation and tagging mistakes. In contrast, after the application of the hand-crafted FST rules for postprocessing, 100 out of the 101 sentences can be parsed. However, this approach involved a lot of manual development work (about 3-4 person months) and has reached a stage where it is difficult to systematically work on further improvements.

### 3.2 Machine-Learned FST Rules

Since there are large amounts of training data that are close to the segmentation and tagging standard the grammar developer wants to use, the idea of inducing FST rules rather than hand-crafting them comes quite naturally. The easiest way to do this is to apply transformation-based learning (TBL) to the combined problem of Chinese segmentation and POS tagging, since the cascade of transformational rules learned in a TBL training run can straightforwardly be translated into a cascade of FST rules.

#### 3.2.1 Transformation-Based Learning and μ-TBL

TBL is a machine learning approach that has been employed to solve a number of problems in natural language processing; most famously, it has been used for part-of-speech tagging (Brill, 1995). TBL is a supervised learning approach, since it relies on gold-annotated training data. In addition, it relies on a set of templates of transformational rules; learning consists in finding a sequence of instantiations of these templates that minimizes the number of errors in a more or less naive base-line output with respect to the gold-annotated training data.

The first attempts to employ TBL to solve the problem of Chinese word segmentation go back to Palmer (1997) and Hockenmaier and Brew (1998). In more recent work, TBL was used for the adaption of the output of a statistical "general purpose" segmenter to standards that vary depending on the application that requires sentence segmentation (Gao et al., 2004). TBL approaches to the combined problem of segmenting and POS-tagging Chinese sentences are reported in Florian and Ngai (2001) and Fung et al. (2004).

Several implementations of the TBL approach are freely available on the web, the most well-known being the so-called Brill tagger, fnTBL, which allows for multi-dimensional TBL, and μ-TBL (Lager, 1999). Among these, we chose μ-TBL for our experiments because (like fnTBL) it is completely flexible as to whether a sample is a word, a character or anything else and (unlike fnTBL) it allows for the induction of optional rules. Probably due to its flexibility, μ-TBL has been used (albeit on a small scale for the most part) for tasks as diverse as POS tagging, map tasks, and machine translation.

#### 3.2.2 Experiment Set-up

We started out with a corpus of thirty gold-segmented and -tagged daily editions of the Xinhua Daily, which were provided by the Institute of Computational Linguistics at Beijing University. Three daily editions, which comprise 5,054 sentences with 129,377 words and 213,936 characters, were set aside for testing purposes; the remaining 27 editions were used for training. With the idea of learning both obligatory and optional transformational rules in mind, we then split the training data into two roughly equally sized subsets. All the data were broken into sentences using a very simple method: The end of a paragraph was always considered a sentence boundary. Within paragraphs, sentence-final punctuation marks such as periods (which are unambiguous in Chinese), question marks and exclamation marks, potentially followed by a closing parenthesis, bracket or quote mark, were considered sentence boundaries.

We then had to come up with a way of casting the problem of combined segmentation and POS tagging as a TBL problem. Following a strategy widely used in Chinese word segmentation, we did this by regarding the problem as a character tagging problem. However, since we intended to learn rules that deal with segmentation and POS tagging simultaneously, we could not adopt the BIO-coding approach.[6] Also, since the TBL-induced transformational rules were to be converted into FST rules, we had to keep our character tagging scheme one-dimensional, unlike Florian and Ngai (2001), who used a multi-dimensional TBL approach to solve the problem of combined segmentation and POS tagging.

The character tagging scheme that we finally chose is illustrated in (6), where a. and b. show the character tags that we used for the analyses in (1a) and (1b) respectively. The scheme consists in tagging the last character of a word with the part-of-speech of the entire word; all non-final characters are tagged with '-'. The main advantages of this character tagging scheme are that it expresses both word boundaries and parts-of-speech and that, at the same time, it is always consistent; inconsistencies between BIO tags indicating word boundaries and part-of-speech tags, which Florian and Ngai (2001), for example, have to resolve, can simply not arise.

(6)

|  | 有 | 意 | 见 |
|---|---|---|---|
| a. | v | - | n |
| b. | - | v | v |

Both of the training data subsets were tagged according to our character tagging scheme and converted to the data format expected by μ-TBL. The first training data subset was used for learning obligatory resegmentation and retagging rules. The corresponding rule templates, which define the space of possible rules to be explored, are given in Figure 1. The training parameters of μ-TBL, which are an accuracy threshold and a score threshold, were set to 0.75 and 5 respectively; this means that a potential rule was only retained if at least 75% of the samples to which it would have applied were actually modified in the sense of the gold standard and not in some other way and that the learning process was terminated when no more rule could be found that applied to at least 5 samples in the first training data subset. With these training parameters, 3,319 obligatory rules were learned by μ-TBL.

Once the obligatory rules had been learned on the first training data subset, they were applied to the second training data subset. Then, optional rules were learned on this second training data subset. The rule templates used for optional rules are very similar to the ones used for obligatory rules; a few templates of optional rules are given in Figure 2. The difference between obligatory rules and optional rules is that the former replace one character tag by another, whereas the latter add character tags. They hence introduce ambiguity, which is why we call them optional rules. Like in the learning of the obligatory rules, the accuracy threshold used was 0.75; the score theshold was set to 7 because the training software seemed to hit a bug below that threshold. 753 optional rules were learned. We did not experiment with the adjustment of the training parameters on a separate held-out set.

Finally, the rule sets learned were converted into the *fst* (Beesley and Karttunen, 2003) notation for transformational rules, so that they could be tested and used in the FST cascade used for preprocessing the input of the Chinese LFG. For evaluation, the converted rules were applied to our test data set of 5,054 sentences. A few example rules learned by μ-TBL with the set-up described above are given in Figure 3; we show them both in μ-TBL notation and in *fst* notation.

### 3.2.3 Results

The results achieved by PKU's tokenizer-tagger on its own and in combination with the transformational rules learned in our experiments are given in Table 1. We compare the output of PKU's

[6] In this character tagging approach to word segmentation, characters are tagged as the beginning of a word (B), inside (or at the end) of a multi-character word (I) or a word of their own (O). Their are numerous variations of this approach.

```
tag:m> - <- wd:'一'@[0] & wd:'个'@[1] &
            tag:q@[1,2,3,4] & {\+q=(-)}.
tag:r>n <- wd:'我'@[-1] & wd:'国'@[0].
tag:add nr <- tag:(-)@[0] & wd:'铸'@[1].
...
```

```
"/" m WS @-> 0 || 一 _ 个 [ ( TAG )
                    CHAR ]^{0,3} "/" q WS
"/" r WS @-> "/" n TB || 我 ( TAG ) 国 _
[..] (@->) "/" n r TB || CHAR _ 铸
...
```

Figure 3: Sample rules learned in our experiments in μ-TBL notation on the left and in *fst* notation on the right[8]

```
tag:A>B <- ch:C@[0].
tag:A>B <- ch:C@[1].
tag:A>B <- ch:C@[-1] & ch:D@[0].
tag:A>B <- ch:C@[0] & ch:D@[1].
tag:A>B <- ch:C@[1] & ch:D@[2].
tag:A>B <- ch:C@[-2] & ch:D@[-1] &
           ch:E@[0].
tag:A>B <- ch:C@[-1] & ch:D@[0] &
           ch:E@[1].
tag:A>B <- ch:C@[0] & ch:D@[1] & ch:E@[2].
tag:A>B <- ch:C@[1] & ch:D@[2] & ch:E@[3].
tag:A>B <- tag:C@[-1].
tag:A>B <- tag:C@[1].
tag:A>B <- tag:C@[1] & tag:D@[2].
tag:A>B <- tag:C@[-2] & tag:D@[-1].
tag:A>B <- tag:C@[-1] & tag:D@[1].
tag:A>B <- tag:C@[1] & tag:D@[2].
tag:A>B <- tag:C@[1] & tag:D@[2] &
           tag:E@[3].
tag:A>B <- tag:C@[-1] & ch:W@[0].
tag:A>B <- tag:C@[1] & ch:W@[0].
tag:A>B <- tag:C@[1] & tag:D@[2] &
           ch:W@[0].
tag:A>B <- tag:C@[-2] & tag:D@[-1] &
           ch:W@[0].
tag:A>B <- tag:C@[-1] & tag:D@[1] &
           ch:W@[0].
tag:A>B <- tag:C@[1] & tag:D@[2] &
           ch:W@[0].
tag:A>B <- tag:C@[1] & tag:D@[2] &
           tag:E@[3] & ch:W@[0].
tag:A>B <- tag:C@[-1] & ch:W@[1].
tag:A>B <- tag:C@[1] & ch:W@[1].
tag:A>B <- tag:C@[1] & tag:D@[2] &
           ch:W@[1].
tag:A>B <- tag:C@[-2] & tag:D@[-1] &
           ch:W@[1].
tag:A>B <- tag:C@[-1] & ch:D@[0] &
           ch:E@[1].
tag:A>B <- tag:C@[-1] & tag:D@[1] &
           ch:W@[1].
tag:A>B <- tag:C@[1] & tag:D@[2] &
           ch:W@[1].
tag:A>B <- tag:C@[1] & tag:D@[2] &
           tag:E@[3] & ch:W@[1].
tag:A>B <- tag:C@[1,2,3,4] & {\+C='-'}.
tag:A>B <- ch:C@[0] & tag:D@[1,2,3,4] &
           {\+D='-'}.
tag:A>B <- tag:C@[-1] & ch:D@[0] &
           tag:E@[1,2,3,4] & {\+E='-'}.
tag:A>B <- ch:C@[0] & ch:D@[1] &
           tag:E@[1,2,3,4] & {\+E='-'}.
```

Figure 1: Templates of obligatory rules used in our experiments

```
tag:add B <- tag:A@[0] & ch:C@[0].
tag:add B <- tag:A@[0] & ch:C@[1].
tag:add B <- tag:A@[0] & ch:C@[-1] &
             ch:D@[0].
...
```

Figure 2: Sample templates of optional rules used in our experiments

tokenizer-tagger run in the mode where it returns only the most probable tag for each word (PKU one tag), of PKU's tokenizer-tagger run in the mode where it returns all possible tags for a given word (PKU all tags), of PKU's tokenizer-tagger in one-tag mode augmented with the obligatory transformational rules learned on the first part of our training data (PKU one tag + deterministic rule set), and of PKU's tokenizer-tagger augmented with both the obligatory and optional rules learned on the first and second parts of our training data respectively (PKU one tag + non-deterministic rule set). We give results in terms of character tag accuracy and ambiguity according to our character tagging scheme. Then we provide evaluation figures for the word level. Finally, we give results referring to the sentence level in order to make clear how serious a problem Chinese segmentation and POS tagging still are for parsers, which obviously operate at the sentence level.

These results show that simply switching from the one-tag mode of PKU's tokenizer-tagger to its all-tags mode is not a solution. First of all, since the tokenizer-tagger always produces only one segmentation regardless of the mode it is used in, segmentation accuracy would stay completely unaffected by this change, which is particularly serious because there is no way for the grammar to recover from segmentation errors and the tokenizer-tagger produces an entirely correct segmentation only for 47.15% of the sentences. Second, the improved tagging accuracy would come at a very heavy price in terms of ambiguity; the median number of combined segmentation and POS tagging analyses per sentence would be 1,440.

In contrast, machine-learned transformation rules are an effective means to improve the output of PKU's tokenizer-tagger. Applying only the obligatory rules that were learned already improves segmented sentence accuracy from 47.15% to 63.14% and tagged sentence accuracy from 14.07% to 27.21%, and this at no cost in terms of ambiguity. Adding the optional rules that were learned and hence making the rule set used for post-processing the output of PKU's tokenizer-tagger non-deterministic makes it possible to improve segmented sentence accuracy and tagged sentence accuracy further to 65.06% and 31.47% respectively, i.e. tagged sentence accuracy is more than doubled with respect to the baseline. While this last improvement does come at a price in terms of ambiguity, the ambiguity resulting from the application of the non-deterministic rule set is very low in comparison to the ambiguity of the output of PKU's tokenizer-tagger in all-tags mode; the median number of analyses per sentences only increases to 2. Finally, it should be noted that the transformational rules provide entirely correct segmentation and POS tagging analyses not only for more sentences, but also for longer sentences. They increase the average length of a correctly segmented sentence from 18.22 words to 21.94 words and the average length of a correctly segmented and POS-tagged sentence from 9.58 words to 16.33 words.

## 4 Comparison to related work and Discussion

Comparing our results to other results in the literature is not an easy task because segmentation and POS tagging standards vary, and our test data have not been used for a final evaluation before. Nevertheless, there are of course systems that perform word segmentation and POS tagging for Chinese and have been evaluated on data similar to our test data.

Published results also vary as to the evaluation measures used, in particular when it comes to combined word segmentation and POS tagging. For word segmentation considered separately, the consensus is to use the (segmentation) F-score (SF). The quality of systems that perform both segmentation and POS tagging is often expressed in terms of (character) tag accuracy (TA), but this obviously depends on the character tagging scheme adopted. An alternative measure is POS tagging F-score (TF), which is the geometric mean of precision and recall of correctly segmented and POS-tagged words. Evaluation measures for the sentence level have not been given in any publication that we are aware of, probably because segmenters and POS taggers are rarely considered as pre-processing modules for parsers, but also because the figures for measures like sentence accuracy are strikingly low.

For systems that perform only word segmentation, we find the following results in the literature: (Gao et al., 2004), who use TBL to adapt a "general purpose" segmenter to varying standards, report an SF of 95.5% on PKU data and an SF of 90.4% on CTB data. (Tseng et al., 2005) achieve an SF of 95.0%, 95.3% and 86.3% on PKU data from the Sighan Bakeoff 2005, PKU data from the Sighan Bakeoff 2003 and CTB data from the Sighan Bakeoff 2003 respectively. Finally, (Zhang et al., 2006) report an SF of 94.8% on PKU data.

For systems that perform both word segmentation and POS tagging, the following results were published: Florian and Ngai (2001) report an SF of 93.55% and a TA of 88.86% on CTB data. Ng and Low (2004) report an SF of 95.2% and a TA of 91.9% on CTB data. Finally, Zhang and Clark (2008) achieve an SF of 95.90% and a TF of 91.34% by 10-fold cross validation using CTB data.

Last but not least, there are parsers that operate on characters rather than words and who perform segmentation and POS tagging as part of the parsing process. Among these, we would like to mention Luo (2003), who reports an SF 96.0% on Chinese Treebank (CTB) data, and (Fung et al., 2004), who achieve "a word segmentation precision/recall performance of 93/94%". Both the SF and the TF results achieved by our "PKU one tag + non-deterministic rule set" setup, whose output is slightly ambiguous, compare favorably with all the results mentioned, and even the results achieved by our "PKU one tag + deterministic rule set" setup are competitive.

## 5 Conclusions and Future Work

The idea of carrying some ambiguity from one processing step into the next in order not to prune good solutions is not new. E.g., Prins and van Noord (2003) use a probabilistic part-of-speech tagger that keeps multiple tags in certain cases for a hand-crafted HPSG-inspired parser for Dutch,

|  | PKU | PKU | PKU one tag + | PKU one tag + |
|---|---|---|---|---|
|  | one tag | all tags | det. rule set | non-det. rule set |
| Character tag accuracy (in %) | 89.98 | 92.79 | 94.69 | 95.27 |
| Avg. number of tags per char. | 1.00 | 1.39 | 1.00 | 1.03 |
| Avg. number of words per sent. | 26.26 | 26.26 | 25.77 | 25.75 |
| Segmented word precision (in %) | 93.00 | 93.00 | 96.18 | 96.46 |
| Segmented word recall (in %) | 95.39 | 95.39 | 96.84 | 97.02 |
| Segmented word F-score (in %) | 94.18 | 94.18 | 96.51 | 96.74 |
| Tagged word precision (in %) | 83.57 | 87.87 | 91.27 | 92.17 |
| Tagged word recall (in %) | 85.72 | 90.23 | 91.89 | 92.71 |
| Tagged word F-score (in %) | 84.63 | 89.03 | 91.58 | 92.44 |
| Segmented sentence accuracy (in %) | 47.15 | 47.15 | 63.14 | 65.06 |
| Avg. nmb. of words per correctly segm. sent. | 18.22 | 18.22 | 21.69 | 21.94 |
| Tagged sentence accuracy (in %) | 14.07 | 21.09 | 27.21 | 31.47 |
| Avg. number of analyses per sent. | 1.00 | 4.61e18 | 1.00 | 12.84 |
| Median nmb. of analyses per sent. | 1 | 1,440 | 1 | 2 |
| Avg. nmb. of words per corr. tagged sent. | 9.58 | 13.20 | 15.11 | 16.33 |

Table 1: Evaluation figures achieved by four different systems on the 5,054 sentences of our test set

and Curran et al. (2006) show the benefits of using a multi-tagger rather than a single-tagger for an induced CCG for English. However, to our knowledge, this idea has not made its way into the field of Chinese parsing so far. Chinese parsing systems either pass on a single segmentation and POS tagging analysis to the parser proper or they are character-based, i.e. segmentation and tagging are part of the parsing process. Although several treebank-induced character-based parsers for Chinese have achieved promising results, this approach is impractical in the development of a hand-crafted deep grammar like the Chinese LFG. We therefore believe that the development of a "multi-tokenizer-tagger" is the way to go for this sort of system (and all systems that can handle a certain amount of ambiguity that may or may not be resolved at later processing stages). Our results show that we have made an important first step in this direction.

As to future work, we hope to resolve the problem of not having a gold standard that is segmented and tagged exactly according to the guidelines established by the Chinese LFG developer by semi-automatically applying the hand-crafted transformational rules that were developed to the PKU gold standard. We will then induce obligatory and optional FST rules from this "grammar-compliant" gold standard and hope that these will be able to replace the hand-crafted transformation rules currently used in the grammar. Finally, we plan to carry out more training runs; in particular, we intend to experiment with lower accuracy (and score) thresholds for optional rules. The idea is to find the optimal balance between ambiguity, which can probably be higher than with our current set of induced rules without affecting efficiency too adversely, and accuracy, which still needs further improvement, as can easily be seen from the sentence accuracy figures.

## References

Kenneth R. Beesley and Lauri Karttunen. 2003. *Finite State Morphology*. CSLI Publications, Stanford, CA.

Eric Brill. 1995. Transformation-based error-driven learning and natural language processing: a case study in part-of-speech tagging. *Computational Linguistics*, 21(4):543–565.

J.M Chang, D.L. Hung, and O.J.L. Tzeng. 1992. Miscue analysis of chinese children's reading behavior at the entry level. *Journal of Chinese Linguistics*, 20(1).

Dick Crouch, Mary Dalrymple, Ron Kaplan, Tracy Holloway King, John Maxwell, and Paula Newman. 2006. XLE documentation. http://www2.parc.com/isl/groups/nltt/xle/doc/.

James R. Curran, Stephen Clark, and David Vadas. 2006. Multi-Tagging for Lexicalized-Grammar Parsing. In *In Proceedings of COLING/ACL-06*, pages 697–704, Sydney, Australia.

Ji Fang and Tracy Holloway King. 2007. An lfg chinese grammar for machine use. In Tracy Holloway

King and Emily M. Bender, editors, *Proceedings of the GEAF 2007 Workshop*. CSLI Studies in Computational Linguistics ONLINE.

Radu Florian and Grace Ngai. 2001. Multidimensional transformation-based learning. In *CoNLL '01: Proceedings of the 2001 workshop on Computational Natural Language Learning*, pages 1–8, Morristown, NJ, USA. Association for Computational Linguistics.

Pascale Fung, Grace Ngai, Yongsheng Yang, and Benfeng Chen. 2004. A maximum-entropy Chinese parser augmented by transformation-based learning. *ACM Transactions on Asian Language Information Processing (TALIP)*, 3(2):159–168.

Jianfeng Gao, Andi Wu, Mu Li, Chang-Ning Huang, Hongqiao Li, Xinsong Xia, and Haowei Qin. 2004. Adaptive Chinese word segmentation. In *ACL '04: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, page 462, Morristown, NJ, USA. Association for Computational Linguistics.

Julia Hockenmaier and Chris Brew. 1998. Error-Driven Segmentation of Chinese. *International Journal of the Chinese and Oriental Languages Information Processing Society*, 8(1):69?–84.

R. Hoosain. 1992. Psychological reality of the word in chinese. In H.-C. Chen and O.J.L. Tzeng, editors, *Language Processing in Chinese*. North-Holland and Elsevier, Amsterdam.

Kylie Hsu. 2002. *Selected Issues in Mandarin Chinese Word Structure Analysis*. The Edwin Mellen Press, Lewiston, New York, USA.

Torbjörn Lager. 1999. The µ-TBL System: Logic Programming Tools for Transformation-Based Learning. In *Proceedings of the Third International Workshop on Computational Natural Language Learning (CoNLL'99)*, Bergen.

Xiaoqiang Luo. 2003. A Maximum Entropy Chinese Character-Based Parser. In Michael Collins and Mark Steedman, editors, *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, pages 192–199.

John Maxwell and Ron Kaplan. 1996. An efficient parser for LFG. In *Proceedings of the First LFG Conference*. CSLI Publications.

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 Dekang Lin and Dekai Wu, editors, *Proceedings of EMNLP 2004*, pages 277–284, Barcelona, Spain, July. Association for Computational Linguistics.

Jerome L. Packard. 2000. *The Morphology of Chinese*. Cambridge University Press, Cambridge, UK.

David D. Palmer. 1997. A trainable rule-based algorithm for word segmentation. In *Proceedings of the 35th annual meeting on Association for Computational Linguistics*, pages 321–328, Morristown, NJ, USA. Association for Computational Linguistics.

Robbert Prins and Gertjan van Noord. 2003. Reinforcing parser preferences through tagging. *Traitement Automatique des Langues*, 44(3):121–139.

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

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 Fourth SIGHAN Workshop on Chinese Language Processing*.

A.D. Wu. 2003. Customizable segmentation of morphologically derived words in chinese. *International Journal of Computational Linguistics and Chinese Language Processing*, 8(1):1–28.

Fei Xia. 2000. The segmentation guidelines for the penn chinese treebank (3.0). Technical report, University of Pennsylvania.

Nianwen Xue, Fu-Dong Chiou, and Martha Palmer. 2002. Building a large-scale annotated Chinese corpus. In *Proceedings of the 19th. International Conference on Computational Linguistics*.

Nianwen Xue, Fei Xia, Fu-Dong Chiou, and Martha Palmer. 2005. The Penn Chinese treebank: Phrase structure annotation of a large corpus. *Natural Language Engineering*, pages 207–238.

Tongqiang Xu（徐通锵）. 1997. *On Language*（语言论）. Dongbei Normal University Publishing, Changchun, China.

Shiwen Yu（俞士汶）, Baobao Chang（常宝宝）, and Weidong Zhan（詹卫东）. 2004. *An Introduction of Computational Linguistics*（计算语言学概论）. Shangwu Yinshuguan Press, Beijing, China.

Yue Zhang and Stephen Clark. 2008. Joint Word Segmentation and POS Tagging Using a Single Perceptron. In *Proceedings of ACL-08*, Columbus, OH.

Ruiqiang Zhang, Genichiro Kikui, and Eiichiro Sumita. 2006. Subword-based tagging for confidence-dependent Chinese word segmentation. In *Proceedings of the COLING/ACL on Main conference poster sessions*, pages 961–968, Morristown, NJ, USA. Association for Computational Linguistics.