Chinese Word Segmentation in MSR-NLP

Andi Wu
Microsoft Research
One Microsoft Way, Redmond, WA 98052
andiwu@microsoft.com

Abstract

Word segmentation in MSR-NLP is an integral part of a sentence analyzer which includes basic segmentation, derivational morphology, named entity recognition, new word identification, word lattice pruning and parsing. The final segmentation is produced from the leaves of parse trees. The output can be customized to meet different segmentation standards through the value combinations of a set of parameters. The system participated in four tracks of the segmentation bakeoff -- PK-open, PK-close, CTB-open and CTB-closed – and ranked #1, #2, #2 and #3 respectively in those tracks. Analysis of the results shows that each component of the system contributed to the scores.

1 System Description

The MSR-NLP Chinese system that participated in the current segmentation bakeoff is not a standalone word segmenter. It is a Chinese sentence analyzer where the leaves of parse trees are displayed as the output of word segmentation. The components of this system are described below.

1.1 Basic segmentation

Each input sentence is first segmented into individual characters.[1] These characters and their combinations are then looked up in a dictionary[2] and a word lattice containing lexicalized words only is formed. Each node in the lattice is a feature matrix that contains the part of speech and other grammatical attributes. Multiple-character words may also have information for resolving segmentation ambiguities. In general, multiple-character words are assigned higher scores than the words they subsume, but words like "才能" are exceptions and such exceptional cases are usually marked in the dictionary. For some of the words that tend to overlap with other words, there is also information as to what the preferred segmentation is. For instance, the preferred segmentation for "会议员" is "会+议员" rather than "会议+员". Such information was collected from segmented corpora and stored in the dictionary. The scores are later used in word lattice pruning and parse ranking (Wu and Jiang 1998).

1.2 Derivational morphology and named entity recognition

After basic segmentation, a set of augmented phrase structure rules are applied to the word lattice to form larger word units which include:

- Words derived from morphological processes such as reduplication, affixation, compounding, merging, splitting, etc.
- Named entities such as person names, place names, company names, product names, numbers, dates, monetary units, etc.

Each of these units is a tree that reflects the history of rule application. They are added to the existing word lattice as single nodes and treated as single words by the parser. The internal structures are useful for various purposes, one of which is the customization of word segmentation: words with such structures can all be displayed as single words or multiple words depending on where the "cuts" are made in the word tree (Wu 2003).

1.3 New word identification

The expanded word lattice built in 1.2 is inspected to detect spots of possible OOV new words. Typical spots of this kind are sequences of single char-

[1] If an input line contains more than one sentence, a sentence separator is applied to break the line into individual sentences, which are then processed one by one and the results are concatenated to form a single output.

[2] The lookup is optimized so that not all possible combinations are tried.

acters that are not subsumed by longer words. We then use the following information to propose new words (Wu and Jiang, 2000).

- The probability of the character string being a sequence of independent words;
- The morphological and syntactic properties of the characters;
- Word formation rules;
- Behavior of each character in existing words (e.g. how likely is this character to be used as the second character of a two-character verb).
- The context in which the characters appear.

The proposed new words are added to the word lattice and they will get used if no successful parse can be obtained without them. When a new word proposed this way has been verified by the parser (i.e. used in a successful parse) more than *n* times, it will automatically become an entry in the dictionary. From then on, this word can be looked up directly from the dictionary instead of being proposed online. This kind of dynamic lexical acquisition has been presented in Wu *et al* (2002).

1.4 Word lattice pruning

Now that all the possible words are in the word lattice, both statistical and linguistic methods are applied to eliminate certain paths. For instance, those paths that contain one or more bound morphemes are pruned away. Single characters that are subsumed by longer words are also thrown out if their independent word probabilities are very low. The result is a much smaller lattice that resembles the n-best paths produced by a statistical word segmenter. Because the final resolution of ambiguities is expected to be done during parsing, the lattice pruning is non-greedy so that no plausible path will be excluded prematurely. Many of the ambiguities that are eliminated here can also be resolved by the parser, but the pruning greatly reduces the complexity of the parsing process, making the parser much faster and more accurate.

1.5 Parsing

The cleaned-up word lattice is then submitted to the parser as the initial entries in the parsing chart. With the assumption that a successful parse of the sentence requires a correct segmentation of the sentence, many segmentation ambiguities are expected to be resolved here. This assumption does not always hold, of course. A sentence can often be parsed in multiple ways and the top-ranking parse is not always the correct one. There are also sentences that are not covered by the grammar and therefore cannot be parsed at all. In this latter case, we back off to partial parsing and use dynamic programming to assemble a tree that consists of the largest sub-trees in the chart.

In most cases, the use of the parser results in better segmentation, but the parser can also mislead us. One of the problems is that the parser treats every input as a sentence and tries to construct an *S* out of it. As a result, even a name like "王爱民" can be analyzed as a sentence with 王 as the subject, 爱 as the verb and 民 as the object, if it appears in the wrong context (or no context).

1.6 Segmentation parameters

Due to the differences in segmentation standards, the leaves of a parse tree do not always correspond to the words in a particular standard. For instance, a Chinese full name is a single leaf in our trees, but it is supposed to be two words (family name + given name) according to the PK standard. Fortunately, most of the words whose segmentation is controversial are built dynamically in our system with their internal structures preserved. A Chinese full name, for example, is a word tree where the top node dominates two nodes: the family name and the given name. Each non-terminal node in a word tree as described in 1.2 is associated with a parameter whose value determines whether the daughters of this node are to be displayed as a singe word or multiple words. Since all the dynamic words are built by phrase structure rules and their word trees reflect the derivational history of rule application, there is a one-to-one correspondence between the types of words and the word-internal structures of those words. A segmentation parameter is associated with each type of words[3] and the value of this parameter determines how the given type of words should be segmented. This makes it possible for the system to quickly adapt to different standards (Wu 2003).

[3] There are about 50 parameters in our system.

1.7 Speed

Our system is not optimized for word segmentation in terms of speed. As we have seen, the system is a sentence analyzer and word segmentation is just the by-product of a parser. The speed we report here is in fact the speed of parsing.

On a single 997 MHz Pentium III machine, the system is able to process 28,740 characters per minute. The speed may vary according to sentence lengths: given texts of the same size, those containing longer sentences will take more time. The number reported here is an average of the time taken to process the test sets of the four tracks we participated in.

We have the option of turning off the parser during word segmentation. When the parser is turned off, segmentation is produced directly from the word lattice with dynamic programming which selects the shortest path. The speed in this case is about 60,000 characters per minute.

2 Evaluation

We participated in the four GB tracks in the first international Chinese word segmentation bakeoff - PK-open, PK-closed, CTB-open and CTB-closed – and ranked #1, #2, #2, and #3 respectively in those tracks. In what follows, we discuss how we got the results: what dictionaries we used, how we used the training data, how much each component contributed to the scores, and the problems that affected our performance.

2.1 Dictionaries

In the open tracks, we used our proprietary dictionary of 89,845 entries, which includes the entries of 7,017 single characters. In the closed tracks, we removed from the dictionary all the words that did not appear in the training data, but kept all the single characters. This resulted in a dictionary of 34,681 entries in the PK track and 18,207 entries in the CTB track. It should be noted that not all the words in the training data are in our dictionary. This explains why the total numbers of entries in those reduced dictionaries are smaller than the vocabulary sizes of the respective training sets even with all the single-character entries included in them.

The dictionary we use in each case is not a simple word list. Every word has one or more parts-of-speech and a number of other grammatical features. No word can be used by the parser unless it has those features. This made it very difficult for us to add all the words in the training data to the dictionary. We did use a semi-automatic process to add as many words as possible, but both the accuracy and coverage of the added grammatical features are questionable due to the lack of manual verification.

2.2 Use of the training data

We used the training data mainly to tune the segmentation parameters of our system. As has been mentioned in 1.6, there are about 50 types of morphologically derived words that are built online in our system and each type has a parameter to determine whether a given unit should be displayed as a single word or separate words. Since our default segmentation is very different from PK or CTB, and PK and CTB also follow different guidelines, we had to try different value combinations of the parameters in each case until we got the optimal settings.

The main problem in the tuning is that many morphologically derived words have been lexicalized in our dictionary and therefore do not have the word-internal structures that they would have if they had been constructed dynamically. As a result, their segmentation is beyond the control of those parameters. To solve this problem, we used the training data to automatically identify all such cases, create a word-internal structure for each of them, and store the word tree in their lexical entries.[4] This made it possible for the parameter values to apply to both the lexicalized and non-lexicalized words. This process can be fairly automatic if the annotation of the training data is completely consistent. However, as we have discovered, the training data is not as consistent as expected, which made total automation impossible.

2.3 Contribution of each component

After we received our individual scores and the reference testing data, we did some ablation ex-

[4] The work is incomplete, since the trees were created only for those words that are in the training data provided.

periments to find out the contribution of each system component in this competition. We turned off the components one at a time (except basic segmentation) and recorded the scores of each ablated system. The results are summarized in the following table, where "DM-NER" stands for "derivational morphology and named entity recognition", "NW-ID" for "new word identification and lexicalization", "pruning" for "lattice pruning" and "tuning" for "tuning of parameter values". Each cell in the table has two percentages. The top one is the F-measure and the bottom one is the OOV word recall rate.

	PK Open	PK closed	CTB open	CTB closed
Complete System	**95.9 %** **79.9 %**	**94.7 %** **68.0 %**	**90.1 %** **73.8 %**	**83.1 %** **43.1 %**
Without DM-NER	90.2 % 44.4 %	88.9 % 33.9 %	86.6 % 66.6 %	79.2 % 33.5 %
Without NW-ID	95.8 % 77.3 %	94.0 % 61.2 %	88.7 % 69.0 %	79.2 % 28.2 %
Without Pruning	92.0 % 77.5 %	90.9 % 65.9 %	85.5 % 69.0 %	78.8 % 39.5 %
Without Parsing	95.5 % 79.9 %	94.4 % 68.5 %	89.8 % 75.0 %	84.0 % 48.1 %
Without Tuning	84.8 % 43.4 %	83.9 % 33.3 %	84.8 % 72.3 %	78.4 % 43.3 %

Several interesting facts are revealed in this break-down:

- The tuning of parameter values has the biggest impact on the scores across the board.
- Derivational morphology and NE recognition is also a main contributor, especially in the PK sets, which presumably contains more named entities.
- The impact of new word identification is minimal when the OOV word rate is low, such as in the PK-open case, but becomes more and more significant as the OOV rate increases.
- Lattice pruning makes a big difference as well. Apparently it cannot be replaced by the parser in terms of the disambiguating function it performs. Another fact, which is not represented in the table, is that parsing is three times slower when lattice pruning is turned off.
- The parser has very limited impact on the scores. Looking at the data, we find that parsing did help to resolve some of the most difficult cases of ambiguities and we would not be able to get the last few points without it. But it seems that most of the common problems can be solved without the parser. In one case (CTB closed), the score is higher when the parser is turned off. This is because the parser may prefer a structure where those dynamically recognized OOV words are broken up into smaller units. For practical purposes, therefore, we may choose to leave out the parser.

2.4 Problems that affected our performance

The main problem is the definition of new words. While our system is fairly aggressive in recognizing new words, both PK and CTB are quite conservative in this respect. Expressions such as "援藏", "反腐", "耳鼻喉", "屡禁不绝" are considered single words in our system but not so in PK or CTB. This made our new word recognition do more harm than good in many cases, though the overall impact is positive. Consistency in the annotated corpora is another problem, but this affects every participant. We also had a technical problem where some sentences remained unsegmented simply because some characters are not in our dictionary.

References

Wu, Andi. 2003. Customizable segmentation of morphologically derived Words in Chinese, to appear in *Computational Linguistics and Chinese Language Processing.*, 8(2).

Wu, Andi, J. Pentheroudakis and Z. Jiang, 2002. Dynamic lexical acquisition in Chinese sentence analysis. In *Proceedings of the 19th International Conference on Computational Linguistics*, pp. 1308-1312, Taipei, Taiwan.

Wu, Andi, J. and Z. Jiang, 2000. Statistically-enhanced new word identification in a rule-based Chinese system, in *Proceedings of the 2nd Chinese Language Processing Workshop*, pp. 46-51, HKUST, Hong Kong.

Wu, Andi, J. and Z. Jiang, 1998. Word segmentation in sentence analysis, in *Proceedings of 1998 International Conference on Chinese Information Processing*, pp. 46-51.169-180, Beijing, China.