CIS 526 Final Project: Chinese Word Segmentation

Mitchell Stern

1 Problem Definition

Unlike English text, in which words are separated by spaces, Chinese text is written as a continuous string of characters and punctuation, without spaces or similar delimiters to indicate separations between words. As such, a common first step in Chinese natural language processing is word segmentation, wherein characters are grouped into minimal linguistic units corresponding to words or short phrases.

More formally, the goal of word segmentation is to partition a sequence of Chinese characters $\langle c_1 c_2 \cdots c_m \rangle$ into contiguous, disjoint subsequences $\langle c_1 \cdots c_{i_1} \rangle, \langle c_{i_1+1} \cdots c_{i_2} \rangle, \cdots, \langle c_{i_{n-1}+1} \cdots c_m \rangle$, where each subsequence corresponds to a distinct Chinese word.

As an example, consider the following Chinese sentence:

我喜欢计算机科学。(I like computer science.)

After passing this sentence through a word segmentation system, we obtain the following output:

我 (I) 喜欢 (like) 计算机 (computer) 科学 (science) 。 (.)

Note that the words we obtain range in length from one to three characters, and that the final punctuation mark is treated separately from the character it follows.

Your goal for this assignment is to build a Chinese word segmenter, which produces a list of word segmentations for a given input sentence. However, since even human annotators often agree less than 80% of the time [2], you will not be required to produce an exact partition of the input sentence. Instead, you may produce any collection of word segmentations, including ones that do not fully cover the sentence, or ones that contain overlapping subsequences. Your output will be evaluated by computing its F-score relative to a reference segmentation, as described in the evaluation section below.

2 Evaluation

Two common evaluation metrics for tasks in natural language processing are precision and recall. For Chinese word segmentation, precision and recall can be computed as follows:

$$\text{Precision} = \frac{\text{number of correctly segmented words}}{\text{total number of segmented words}}$$

$$\text{Recall} = \frac{\text{number of correctly segmented words}}{\text{total number of words in gold data}}$$

However, high precision can easily be obtained by outputting a small number of high-probability segmentations, and high recall can easily be obtained by outputting all possible segmentations of the text into one-character words, two-character words, etc.

For this reason, scores will be assigned to outputs using the F-score metric, which is the harmonic mean of precision and recall:

$$\text{F-Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

This metric provides a quantitative measure of the tradeoff between precision and recall, requiring high-scoring systems to perform well from both perspectives.

3 Data

The data for this assignment comes from the Second International Chinese Word Segmentation Bakeoff [6], organized by Tom Emerson of Basis Technology. Training and test sentences have been sampled from the Microsoft Research corpus, and evaluations will be made with respect to the gold-standard segmentations provided by the competition organizer.

4 Default System

As a simple baseline, we could imagine outputting each character as its own word. More effective, perhaps, would be to output each pair of adjacent characters as a separate word.

This class of solutions has been implemented as the default system, and is included in the provided code. Given a set of lengths $\{\ell_1, \cdots, \ell_n\}$ as command line arguments, the default system will produce all segmentations of lengths ℓ_1, ℓ_2, $\cdots$, and ℓ_n.

Note that we can achieve very high recall by running the default system with all lengths of a reasonable size, e.g. one through ten, but precision will suffer as more incorrect segmentations are produced. You should experiment with the default system to gain some intuition for the precision-recall tradeoff inherent in Chinese word segmentation.

5 Baseline System and Beyond

A slightly more sophisticated baseline is the maximum matching strategy. Given a lexicon of Chinese words, we can greedily segment a sentence by scanning the characters from left to right, finding the longest sequence of characters present in the lexicon, and repeating this process until the whole sentence has been processed. A slight variant is the reverse maximum matching strategy, in which greedy segmentation is performed while scanning the sentence backwards from right to left.

Of course, these simple heuristics often fail, and so you might consider looking into one of the following more sophisticated approaches for inspiration:

- Conditional Random Fields
  - Optimizing Chinese Word Segmentation for Machine Translation Performance [1]
  - A Conditional Random Field Word Segmenter [4]
- Linear Mixture Models
  - Chinese Word Segmentation and Named Entity Recognition: A Pragmatic Approach [2]
- Compression-Based Algorithms
  - A Compression-Based Algorithm for Chinese Word Segmentation [3]

But the sky's the limit! There are many, many ways that you can improve the performance of the baseline segmentation system, and you can try anything you want as long as you follow the ground rules of the previous homework assignments.

6 Potential Pitfalls

Two of the main challenges in Chinese word segmentation include the handling of ambiguous sentences and the detection of out-of-vocabulary words.

It is often the case that a sentence has many valid segmentations, but typically only a small number of these are deemed acceptable by a native speaker. Therefore, pruning the search space and identifying plausible segmentations are key barriers that any segmentation system will have to overcome.

In addition, although the set of characters in modern Chinese is fixed, many mechanisms exist for the creation of new words, such as compounding, abbreviation, and transliteration of foreign words. Hence, the detection of previously unencountered words is another issue that developers of segmentation systems must address.

References

[1] Pi-Chuan Chang, Michel Galley, and Christopher D. Manning. 2008. Optimizing Chinese Word Segmentation for Machine Translation Performance. In Proceedings of the Third Workshop on Statistical Machine Translation (StatMT '08). Association for Computational Linguistics, Stroudsburg, PA, USA, 224-232. http://aclweb.org/anthology/W/W08/W08-0336.pdf

[2] Jianfeng Gao, Mu Li, Andi Wu, and Chang-Ning Huang. 2005. Chinese Word Segmentation and Named Entity Recognition: A Pragmatic Approach. Computational Linguistics 31, 4 (December 2005), 531-574. http://acl.ldc.upenn.edu/J/J05/J05-4005.pdf

[3] W. J. Teahan, Rodger McNab, Yingying Wen, and Ian H. Witten. 2000. A Compression-based Algorithm for Chinese Word Segmentation. Computational Linguistics 26, 3 (September 2000), 375-393. http://acl.ldc.upenn.edu/J/J00/J00-3004.pdf

[4] Huihsin Tseng, Pichuan Chang, Galen Andrew, Daniel Jurafsky, and Christopher Manning. 2005. A Conditional Random Field Word Segmenter. In Fourth SIGHAN Workshop on Chinese Language Processing. Association for Computational Linguistics. http://acl.ldc.upenn.edu/I/I05/I05-3027.pdf

[5] Kam-Fai Wong, Wenjie Li, Ruifeng Xu, Zheng-sheng Zhang. 2009. Introduction to Chinese Natural Language Processing. Morgan and Claypool Publishers. http://www.morganclaypool.com/doi/abs/10.2200/S00211ED1V01Y200909HLT004

[6] Second International Chinese Word Segmentation Bakeoff. http://www.sighan.org/bakeoff2005/