Segmenter for Chinese

Kalle Benéus
Department of Computer Science
Lund University, Sweden
d04kb@student.lth.se

Johan Fänge
Department of Computer Science
Lund University, Sweden
d04jfa@student.lth.se

Abstract

This article examines different methods of word segmentation for Chinese. First we look at some of the difficulties that need to be overcome, as well as how current research is attacking the problem. We then examine a few simple methods in order to establish their efficiency and where there is room for improvement.

1 Introduction

Our goal is to investigate different methods for performing segmentation of Chinese. This includes studying their effectiveness in simple implementations as well as looking to their potential for expansion. By *segmentation*, we mean breaking up a text into individual words. In the European languages, among others, this is not something we need to worry about as segmentation is built into the language itself. However, this is not the case with for example Chinese, Japanese and Thai. Here the text is presented as an unbroken string of Chinese characters, separated only by commas and full stops, etc.

Being able to break up a string into individual words is valuable for future processing of the text. This would allow evaluation of for example parts-of-speech. It should be pointed out however that the rules of segmentation are by no means easily defined, or for that matter commonly agreed upon.

We have investigated several methods for performing segmentation of Chinese, and each will be presented below. First we intend to give some background to the problem, along with a short explanation of the Chinese written language. Here we also intend to examine the possible problems and explain why this field is complicated.

We will then take a look at all the different methods we have used, along with a breakdown of their advantages and disadvantages. This will be followed with a section detailing our results with each of these methods.

Finally we will discuss possible improvements to the segmentation process and end with our conclusions.

2 Background

Chinese uses a method of writing where one Chinese character can either represent a word all by itself, or it can join with other characters to form compounds that are of lengths two or more characters. A sentence is presented as an unbroken string of characters, with no spaces separating words.

To determine what a word is in Chinese is no easy task as there exists several definitions. Even allowing native speakers to manually segment a text might yield quite different results. As a related example consider “Post Office” in English, which in the context of Chinese could be seen as either one or two words. For these reasons previous segmentation efforts have devoted much time to the task of defining a word according to their implementation. This also means that this definition may vary between different pre-segmented corpuses of Chinese.

At the First International Chinese Word Segmentation Bakeoff, a competition in the field, a point was made of providing corpuses with different standards of segmentation, in order to ascertain how well the contenders’ implementations could conform to different standards. Here it is also argued that a single unified standard would not be ideal, as different standards are more suitable for different purposes (Emerson et al. 2003).

Putting the definition aside, we will now present some more concrete problems that will face an implementation of a segmenter. First of all, names in Chinese are written with the same characters used for common words. Most commonly two or three characters are used for a full name. This is a problem as names are difficult to predict and might therefore be indistinguishable from an unknown word. This in turn leads to the bigger problem that the characters used in a name might be able to form a compound with characters appearing immediately before and after the name. As the name itself will likely not be processed as a word, this heightens the likelihood of an implementation that instead chooses the erroneous compounds. Names are only one example where this problem occurs. Naturally the same goes for other unknown words as well, and given the rate at which new words appear in every language it is almost inconceivable that unknown words will not be found at some point.

A similar problem is caused by the Chinese characters' propensity for forming compounds with several other characters, as well as forming a word all by itself. This means that given three Chinese characters, it is possible for the character in the middle to form a compound both with the first and the last character. Since whichever character is left out of the compound is likely to be able to stand as a word by itself, catching an error in this context is not trivial.

Both of these types of errors require a good grasp on the context of the sentence in order to ensure a correct solution, calling for complex solutions to the segmentation process. There are many other challenges, but hopefully this has provided a small insight to the types of problems that is inherent in the language, and we will for now not delve further into this subject matter.

3 Previous research

Traditionally, methods of segmentation are divided into two main categories. The first relies heavily on the use of a dictionary. This approach is inherently quite simple while giving quite good results, which has led to it gaining popularity. The other category mainly involves the use of statistics and linguistic knowledge. This usually means that the possible segmentations are identified, whereupon different rules are applied in order to determine which segmentation yields the best result (Gao et al. 2006).

For the dictionary method, several different strategies exist, starting at the simple yet surprisingly effective method of walking through the text from left to right, and repeatedly choosing as the next word the longest one found in the dictionary. In a few ambiguous situations and with words not in the dictionary (called out-of-dictionary words, or OOD), however, this does not work, calling for further systems designed for handling personal and place names. This normally involves a large database dedicated to proper nouns, as well as relatively complex algorithms for finding them in a text (Dai et al. 2006). The process of handling ambiguous situations is usually quite disconnected from the main segmentation. However, there are those that argue that these should not be separated, but handled as a single process (Gao et al. 2006).

Regarding dictionary sizes, a case has been made for the size not being the deciding factor for the effectiveness of a dictionary approach. What with the constant addition of new words no dictionary, however large, can hope to cover everything. Rather than the size, the appropriateness of the used dictionary would then have a higher significance (Li et al. 1998).

High-end results vary quite strongly depending on the corpus and the standard used, but they can be found to lie in the region of 90-95% for precision and recall (Emerson et al. 2003).

4 Segmentation methods

In this section we will describe the methods that we have used. Only the methods in themselves will be described, with results left for their own section later.

4.1 Dictionary approach

The dictionary method of segmentation consists of taking a Chinese sentence and trying to match the characters in the sentence with the contents of a dictionary. This can be done in several different ways.

The benefit of the dictionary approach is that it is very simple and easy to understand. Given a well-adjusted dictionary it also has a fairly high rate of precision.

The first and perhaps biggest challenge with his approach is to construct a dictionary that matches the standard for what constitutes a word. Common dictionaries are usually not intended to reflect grammatical constructions and may not be consistent with its entries. To cite an example, in a dictionary we looked at the equivalent of the English word 'my/mine', consisting of the character for 'I' coupled with another character representing the possessive particle, appears as could be expected. However, the possessive particle does not appear in combination with all possible nouns and pronouns in the dictionary, thus making it inconsistent with what constitutes a word regarding this particular grammatical property.

A major problem for a general purpose segmenter using the dictionary approach is that of OODs words, i.e. when some words in the corpus are not present in the dictionary. In this case not only will the word itself, in all likelihood, not be segmented correctly, but it may also make it more likely that characters of that word are absorbed into other, erroneous compounds. It should be noted however that the problem behind OODs is not unique to the dictionary approach.

The next step is to decide on what basis a word should be matched to those appearing in the dictionary. A very straightforward method is to always match the longest possible string of characters that has an entry in the dictionary. Starting from the beginning and successively eliminating any matches from the current sentence, this would give you a segmentation using the assumption that if in any given case a match exists for a longer compound, this is more likely to be the correct one.

We explored the pure dictionary approeach by using a freely available dictionary (MDBG 2008), and segmenting the text using longest prefix match as described above. Note that this online dictionary also has an online segmenter of its own based on the dictionary. This made us curious as to how well this actually performed, and motivated the choice of this particular dictionary.

4.2 Word frequency

In an attempt to improve the dictionary approach we tried to take into account word frequency. With access to training data, one can use this to determine the frequencies of words. Once these have been established, you can calculate the probability of the appearance of any given word and then, create a set of matches that gives the highest cumulative "score". The score would here be directly linked to word frequency. While this would fall under the category of statistical methods, outlined earlier, it also has elements of the dictionary method, as we in this case use the words appearing in the training data as our dictionary.

In implementing this approach we simply maximized the sum of logs for each sentence, reasoning that a sentence that has words with high frequency would be a more likely sentence. As we shall see this method turned out to perform very bad. Playing around some more with it we found that instead choosing the least frequent word yielded a fairly good score (yet still inferior to a pure dictionary approach).

4.3 Weka

After this we tried another approach completely. We used a classifier based on machine learning techniques. We use the general-purpose system Weka for this, by massaging our data to a format that it accepted, based on tagging characters as Start or In-word, and then building a segmenting program based on its classifiers. As a feature vector we chose a context of two words behind and after, and the last two tags chosen.

However the large number of instances to classify meant our choice for classifiers was quite limited. Among the more successful classifiers we tried Naive Bayes is notable.

5 Results

Below is the data we collected in our investigations. We measured Precision, Recall and calculated the corresponding F-measure.

Method	*P*	*R*	*F*
dict, longest word, online dictionary	83%	87%	85%
dict, longest word, train as dict	98%	97%	98%
dict, "most frequent" sentence, train as dict	39%	58%	47%
dict, "least frequent" sentence, train as dict	93%	94%	94%
weka, naive bayes, train as dict	75%	75%	75%

The dictionary approach performed the best, and was also the simplest one. As can be seen when using the training data on itself (which excludes the possibility of OODs) it performs very well. Even a common dictionary not specialized for the purpose of segmenting performs well.

Our frequency based approach turned out to perform pretty badly. In fact, when we tried the completely opposite our original thinking we got quite acceptable results.

Our attempts based on machine learning wasn't very successful either. Most classifiers didn't manage to process the large amount of data we had (and performed very bad when fed with only a small amount of data). One of the best that also produced a model of reasonable size was Naive Bayes, which still performed worse than simply using a non-specialized dictionary.

6 Improvements and Discussion

The dictionary method has high potential, as seen through our experiments with a dictionary generated from the test corpus itself. If a large enough corpuses is used to create a dictionary, this method could potentially become highly efficient. Alternatively, one could to get high precision take a regular dictionary and manually edit it to fit a given definition of words, which could be a very time consuming task.

Regarding the matching algorithm, we saw how a simple strategy of matching the longest word consistently gave us the best results. This leads us to believe that it is a good strategy for basic matches. However, one could also consider a combination strategy where we also take into account the frequency of words. This could be used for making decisions in ambiguous situations where several (shorter) compounds are possible. This could then be used as a rough strategy in lieu of the more complex version where deciding the context would be necessary.

Regarding our frequency method, some further consideration gives that the frequency in itself is not a good measure for how probable a part of a sentence is. A better estimate would require word length to be considered as well. It would be unlikely for several characters to form a long word by coincidence, given the volume of characters available, meaning that the higher frequency of shorter words is to be quite expected. This may explain why a longest prefix match works as well as it does, and also why a least frequent word method, which typically favors longer words, works so well. As such a probability based method may still work quite well, but is not what we implemented.

Another interesting idea would be to combine a dictionary with a machine learning algorithm for OODs.

References

Shuaixiang Dai and Xin Li. 2006. NetEase Automatic Chinese Word Segmentation. *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing*: 193-196.

Thomas Emerson and Richard Sproat. 2003. The First International Chinese Word Segmentation Bakeoff. http://www.sighan.org/bakeoff2003/paper.pdf, retrieved on 2008-01-12.

Jianfeng Gao, Chang-Ning Huang, Mu Li, and Andi Wu. 2006. Chinese Word Segmentation and Named Entity Recognition, A Pragmatic Approach. *Computational Linguistics*, 31(4):531-574.

Haizhou Li and Baosheng Yuan. 1998. Chinese Word Segmentation *Language, Information and Computation (PACLIC12)*, 18-20 Feb, 212-217.

MDBG Retrieval date: 2008-09-26 MDBG Chinese-English dictionary - CC-CEDICT http://www.mdbg.net/chindict/chindict.php?page=cc-cedict