A Boundary-Oriented Chinese Segmentation Method Using N-Gram Mutual Information

Ling-Xiang Tang[1], **Shlomo Geva**[1], **Andrew Trotman**[2], **Yue Xu**[1]

[1]Faculty of Science and Technology
Queensland University of Technology
Brisbane, Australia
{l4.tang,s.geva,yue.xu}@qut.edu.au

[2]Department of Computer Science
University of Otago
Dunedin, New Zealand
andrew@cs.otago.ac.nz

Abstract

This paper describes our participation in the Chinese word segmentation task of CIPS-SIGHAN 2010. We implemented an n-gram mutual information (NGMI) based segmentation algorithm with the mixed-up features from unsupervised, supervised and dictionary-based segmentation methods. This algorithm is also combined with a simple strategy for out-of-vocabulary (OOV) word recognition. The evaluation for both open and closed training shows encouraging results of our system. The results for OOV word recognition in closed training evaluation were however found unsatisfactory.

1 Introduction

Chinese segmentation has been an interesting research topic for decades. Lots of delicate methods are used for providing good Chinese segmentation. In general, on the basis of the required human effort, Chinese word segmentation approaches can be classified into two categories: supervised and unsupervised.

Particularly, supervised segmentation methods can achieve a very high precision on the targeted knowledge domain with the help of training corpus—the manually segmented text collection. On the other hand, unsupervised methods are suitable for more general Chinese segmentation where there is no or limited training data available. The resulting segmentation accuracy with unsupervised methods may not be very satisfying, but the human effort for creating the training data set is not absolutely required.

In the Chinese word segmentation task of CIPS-SIGHAN 2010, the focus is on the performance of Chinese segmentation on cross-domain text. There are in total two types of evaluations: closed and open. We participated in both closed and open training evaluation tasks and both simplified and traditional Chinse segmentation subtasks. For the closed training evaluation, the provided resource for system training is limited and using external resources such as trained segmentation software, corpus, dictionaries, lexicons, etc are forbidden; especially, human-encoded rules specified in the segmentation algorithm are not allowed.

For the bakeoff of this year, we implemented a boundary-oriented NGMI-based algorithm with the mixed-up features from supervised, unsupervised and dictionary-based methods for the segmentation of cross-domain text. In order to detect words not in the training corpus, we also used a simple strategy for out-of-vocabulary word recognition.

2 A Boundary-Oriented Segmentation Method

2.1 N-Gram Mutual Information

It is a challenge to segment text that is out-of-domain for supervised methods which are good at the segmentation for the text that has been seen segmented before. On the other hand, unsupervised segmentation methods could help to discover words even if they are not in vocabularies. To conquer the goal of segmenting text that is out-of-domain and to take advantage of the training corpus, we use n-gram mutual information (NGMI)(Tang et al., 2009) — an unsupervised boundary-oriented segmentation method and make it trainable for cross-domain text segmentation.

As an unsupervised segmentation approach, NGMI is derived from the character-based mutual information(Sproat & Shih, 1990), but unlike its ancestor it can additionally recognise words longer than two characters. Generally, mutual information is used to measure the association strength of two adjoining entities (characters or words) in a given corpus. The stronger association, the more likely it is that they should be together. The association score MI for the adjacent two entities (x and y) is calculated as:

$$MI(x,y) = \log \frac{\frac{freq(xy)}{N}}{\frac{freq(x)}{N}\ \frac{freq(x)}{N}} \approx \log \frac{p(xy)}{p(x)p(y)} \tag{1}$$

where *freq(x)* is the frequency of entity *x* occurring the given corpus; *freq(xy)* is the frequency of entity *xy* (*x* followed by *y*) occurring in the corpus; *N* is the size of entities in the given corpus; *p(x)* is an estimate of the probability of entity *x* occurring in corpus, calculated as *freq(x)/N.*

NGMI separates words by choosing the most probable boundaries in the unsegmented text with the help of a frequency table of n-gram string patterns. Such a frequency table can be built from any selected text.

The main concept of NGMI is to find the boundaries between words by combining contextual information rather than looking for just words. Any place between two Chinese characters could be a possible boundary. To find the most rightful ones, boundary confidence (BC) is introduced to measure the confidence of having words separated correctly. In other words, BC measures the association level of the left and right characters around each possible boundary to decide whether the boundary should be actually placed.

For any input string, suppose that we have:

$$s = c_1c_2c_3 \dots c_ic_{i+1} \dots c_{n-2}c_{n-1}c_n \tag{2}$$

The *boundary confidence* of a possible boundary (|) is defined as:

$$BC(L|R) = NGMI_{min}(L|R) \tag{3}$$

where L and R are the adjoining left and right segments with up to two characters from each side of the boundary (|) and $L = c_{i-2}c_{i-1}$, $R = c_ic_{i+1}$; and $NGMI_{min}$ is calculated as:

$$\begin{aligned} NGMI_{min}(L|R) = \min\big(&MI(c_{i-1},c_i), \\ &MI(c_{i-2}c_{i-1},c_i), \\ &MI(c_{i-1},c_ic_{i+1}), \\ &MI(c_{i-2}c_{i-1},c_ic_{i+1})\big) \end{aligned} \tag{4}$$

Basically, $NGMI_{min}$ considers mutual information of *k* (*k=2, or k= 4*) pairs of segments around the boundary; the one with the lowest value is used as the score of boundary confidence. Those segment pairs used in $NGMI_{min}$ calculation are named adjoining segment pairs (ASPs). Each ASP consists of a pair of adjoining segments.

For the boundary confidence of the boundaries at the beginning or end of an input string, we can retrieve only one character from one side of the boundary. So for these two kinds of boundaries differently we have:

$$\begin{aligned} &NGMI_{min}(c_1|c_2c_3) \\ &= \min\big(MI(c_1,c_2),\ MI(c_1,c_2c_3)\big) \end{aligned} \tag{5}$$

$$\begin{aligned} &NGMI_{min}(c_{n-2}c_{n-1}|c_n) \\ &= min\big(MI(c_{n-1},c_n),\ MI(c_{n-2}c_{n-1},c_n)\big) \end{aligned} \tag{6}$$

For any possible boundary the lower confidence score it has, the more likely it is an actual boundary. A threshold then can be set to decide whether a boundary should be placed. So even without a lexicon, it is still probable to segment text with a certain precision which just simply means the suggested words are all out-of-vocabulary. Hence, NGMI can be subsequently used for OOV word recognition.

2.2 Supervised NGMI

The idea of making NGMI trainable is to turn the segmented text into a word based frequency table. It is a table that records only words, adjoining word pairs and their frequencies. For example, given a piece of training text – "A B C E B C A B" (where A, B, C and E are n-gram Chinese words), its frequency table should look like the following:

A|B	2
B|C	2
C|E	1
C|A	1
A	2
B	3
C	2
E	1

Also, when doing the boundary confidence computation, any substrings (children) of the words (parents) in this table are set to have the same frequency as their parents'.

3 Segmentation System Design

3.1 Frequency Table and Its Alignment

In order to resolve ambiguity and also recognise OOV terms, statistical information of n-gram string patterns in test files should be collected. There are in total two groups of frequency information used in the segmentation. One is from the training data, recording the frequency information of the actual words and the adjoining word pairs; the other is from the unsegmented text, containing frequency information of all possible n-gram patterns.

However, the statistical data collected from the unsegmented test file contains many noise patterns. It is necessary to remove those noise patterns from the table to avoid negative impact on the final BC computation. Therefore, an alignment of the pattern frequencies obtained from the test file is performed to reduce noise.

The frequency alignment is conducted in a few steps. First, build a frequency table of all string patterns for the unsegmented text including those having a frequency of one. Second, the frequency table is sorted by the frequency and the length of the patterns. Longer patterns have a higher ranking than the shorter ones; for patterns of same length the ones having higher frequency are ranked higher than those having lower. Next, starting from the beginning of the table where the longest and the most frequent pattern have the highest ranking, retrieve one record each time and remove from the table all its sub-patterns which have the same frequency as its parent's.

After such a frequency alignment is done, two frequency tables are merged into one and ready for the final boundary confidence calculation.

3.2 Segmentation

In the training and the system testing stages, the segmentation results using boundary confidence alone for word disambiguation were found unsatisfactory. Trying to achieve as high performance as possible, the overall word segmentation for the bakeoff is done by using a hybrid algorithm which is a combination of NGMI for general word segmentation, and the backward maximum match (BMM) method for the final word disambiguation.

Since it is common for a Chinese document containing various types of characters: Chinese, digit, alphabet and characters from other languages, segmentation needs to be considered for two particular forms of Chinese words: 1) words containing non-Chinese characters such as numbers or letters; and 2) words containing purely Chinese characters.

In order to simplify the process of overall segmentation, boundaries are automatically added to the places in which Chinese characters precede non-Chinese characters. Additionally, for words containing numbers or letters, we only search those begin with numbers or letters and end with Chinese character(s) against the given lexicons. If the search fails, the part with all non-Chinese characters remains the same and a boundary is added between the non-Chinese character and the Chinese character.

For example, to segment a sentence "一万多人喜迎１９９８年新春佳节", it consists of there following main steps:

- First, because of …迎 | １…, only "一万多人喜迎" requires initial segmentation.
- Next, find a matched word "１９９８年" in a given lexicon.

- Last, segment “新春佳节”.

So the critical part of the segmentation algorithm is to segment strings with purely Chinese characters.

By already knowing the actual word information (i.e. a vocabulary from the labelled training data), it can be set in our algorithm that when computing BCs each possible boundary is assigned with a score falling in one of the following four BC categories:

- INSEPARATABLE
- THRESHOLD
- normal boundary confidence score
- ABSOLUTE-BOUNDARY

INSEPARATABLE means the characters around the possible boundary are a part of an actual word; ABSOLUTE-BOUNDARY means the adjoining segments pairs are not seen in any words or string patterns. THRESHOLD is a threshold value that is given to a possible boundary for which only one of ASPs can be found in the word pair table, and its length is greater than two.

After finishing all BC computations for an input string, it then can be broken down into segments separated by the boundaries having a BC score that is lower than or equals to the threshold value. For each segment, it can be checked if it is a word in the vocabulary or if it is an OOV term using an OOV judgement formula that will be discussed in Section 3.3. If a segment is not a word or an OOV term, it means there is an ambiguity in that segment. For example, given a sentence “…送行李…”, the substring “送行李” inside the sentence can be either segmented into “送行 | 李” or “送 | 行李”.

To disambiguate it, a segment is divided into two chunks at the place having the lowest BC score. If one of the chunks is a word or OOV term, this two-chunk breaking-down operation continues on the remaining non-word chunk until both divided chunks are words, or none of them is a word or an OOV term. After this recursive operation is finished, if there are still non-word chunks left they will be further segmented using the BMM method.

The overall segmentation algorithm for an all-Chinese string can be summarised as follows:

1) Compute BC for each possible boundary.
2) Input string becomes segments that are separated by the boundaries having a low BC score (not higher than the threshold).
3) For each remaining non-word segment resulting from step 2, it gets recursively broken down into two chunks at the place having the lowest BC among this segment based on the scores from step 1. This breaking-down-into-two-chunk loop continues on the non-word chunk if the other is a word or an OOV term; otherwise, all the remaining non-word chucks are further segmented using the backward maximum match method.

3.3 OOV Word Recognition

We use a simple strategy for OOV word detection. It is assumed that an n-gram string pattern can be qualified as an OOV word if it repeats frequently within only a short span of text or a few documents. So to recognise OOV words, the statistical data extracted from the unsegmented text needs to contain not only pattern frequency information but also document frequency information. However, the documents in the test data are boundary-less. To obtain document frequencies for string patterns, we separate test files into a set of virtual documents by splitting them according size. The size of the virtual document (VDS) is adjustable.

For a given non-word string pattern *S*, we then can compute its probability of being an OOV term by using:

$$OOV\_P(S) = \frac{tf}{df} \tag{7}$$

where *tf* is the term frequency of the string pattern *S*; *df* is the virtual document frequency of the string pattern. Then S is considered an OOV candidate, if it satisfies:

$$OOV_{P(S)} > OOV\_THRES \tag{8}$$

where *OOV_THRES* is an adjustable threshold value used to filter out the patterns with lower probability of being OOV words. However, using this strategy could have side effects on the segmentation performance because not all the suggested OOV words could be correct.

4 Experiments

4.1 Experimental Environment

OS	GNU/Linux 2.6.32.11-99.fc12.x86_64
CPU	Intel(R) Core(TM)2 Duo CPU E6550 @ 2.33GHz
MEM	8G memory
BUILD	DEBUG build without optimisation

Table 1. Software and Hardware Environment.

The information of operating system and hardware used in the experiments is given in Table 1.

4.2 Parameters Settings

Parameter	Value
N	# of words in training corpus
THRESHOLD	log(1/N)
VDS	10,000bytes
OOV_THRES	2.3

Table 2. System settings used in both closed and open training evaluation.

Table 2 shows the parameters used in the system for segmentation and OOV recognition.

4.3 Closed and Open Training

For both closed and open training evaluations, the algorithm and parameters used for segmentation and OOV detection are exactly the same. This is true except for an extra dictionary - cc-cedict (MDBG) being used in the open training evaluation.

4.4 Segmentation Efficiency

SUBTASK	DOMAIN	TIME
simplified (closed)	A	2m19.841s
	B	2m1.405s
	C	1m57.819s
	D	1m54.375s
simplified (open)	A	3m52.726s
	B	3m20.907s
	C	3m10.398s
	D	3m22.866s
traditional (closed)	A	2m33.448s
	B	2m56.056s
	C	3m7.103s
	D	3m14.286s
traditional (open)	A	3m14.595s
	B	3m41.634s
	C	3m53.839s
	D	4m10.099s

Table 3. The execution time of segmentations for four different domains in both simplified and traditional Chinese subtasks.

Table 3 shows the execution time of all tasks for generating the segmentation outputs. The execution time listed in the table includes the time for loading the training frequency table, building the frequency table from the test file, and producing the actual segmentation results.

5 Evaluation

5.1 Segmentation Results

Simplified Chinese

Task	R	P	F1	R_{OOV}	RR_{OOV}	RR_{IV}
A (c)	0.907	0.862	0.884	0.069	0.206	0.959
A (o)	0.869	0.873	0.871	0.069	0.657	0.885
B (c)	0.876	0.844	0.86	0.152	0.457	0.951
B (o)	0.859	0.878	0.868	0.152	0.668	0.893
C (c)	0.885	0.804	0.842	0.110	0.218	0.967
C (o)	0.865	0.846	0.855	0.110	0.559	0.903
D (c)	0.904	0.865	0.884	0.087	0.321	0.960
D (o)	0.853	0.850	0.851	0.087	0.438	0.893

Traditional Chinese

Task	R	P	F1	R_{OOV}	RR_{OOV}	RR_{IV}
A (c)	0.864	0.789	0.825	0.094	0.105	0.943
A (o)	0.804	0.722	0.761	0.094	0.234	0.863
B (c)	0.868	0.85	0.859	0.094	0.316	0.926
B (o)	0.789	0.736	0.761	0.094	0.35	0.834
C (c)	0.871	0.815	0.842	0.075	0.115	0.932
C (o)	0.811	0.74	0.774	0.075	0.254	0.856
D (c)	0.875	0.834	0.854	0.068	0.169	0.926
D (o)	0.811	0.753	0.781	0.068	0.235	0.853

Table 4. The segmentation results for four domains in both closed and open training evaluations. (c) – closed; (o) – open; A - Literature; B – Computer; C – Medicine; D – Finance. R_{OOV} is the OOV rate in the test file.

In the Chinese word segmentation task of CIPS-SIGHAN 2010, the system performance is measured by five metrics: recall (R), precision (P), F-measure (F1), recall rate of OOV words (RR_{OOV}), and recall rate of words in vocabulary (RR_{IV}).

The official results of our system for both open and closed training evaluation are given in Table 4. The recall rates, precision values, and F1-scores of all tasks show promising results of our system in the segmentation for cross-domain text. However, the gaps between our scores and the bakeoff bests also suggest that there is still plenty of room for performance improvements in our system.

The OOV recall rates (RRoov) showed in Table 4 demonstrate that the OOV recognition strategy used in our system can achieve a certain level of OOV word discovery in closed training evaluation. The overall result for the OOV word recognition is not very satisfactory if comparing it with the best result from other bakeoff participants. But for the open training evaluation the OOV recall rate picked up significantly, which indicates that the extra dictionary - cc-cedict covers a fair amount of terms for various domains.

5.2 Possible Further Improvements

Due to finishing the implementation of our segmentation system in a short time, we believe that there might be many program bugs which had negative effects on our system and leaded to producing results not as expected. In an analysis of the segmentation outputs, words starting with numbers were found incorrectly segmented because of the different encodings used in the training and test files for digits. Moreover, the disambiguation in breaking down a non-word segment which contains at least an n-gram word could lead to an all-single-character-word segmentation. This should certainly be avoided.

Also, the current OOV word recognition strategy may detect a few good OOV words, but also introduces incorrect segmentation consistently through the whole input text if OOV words are mistakenly identified. If this OOV word recognition used in our system can be further improved, it can help to alleviate the problem of performance deterioration.

For the open training, if language rules can be encoded in both word segmentation and OOV word recognition, it certainly is another beneficial method to improve the overall precision and recall rate.

6 Conclusions

In this paper, we describe a novel hybrid boundary-oriented NGMI-based segmentation method, which combines a simple strategy for OOV word recognition. The evaluation results show reasonable performance of our system in cross-domain text segmentation even with the negative effects from system bugs and the OOV word recognition strategy. It is believed that the segmentation system can be improved by fixing the existing program bugs, and having a better OOV word recognition strategy. Performance can also be further improved by incorporating language or domain specific knowledge into the system.

References

MDBG. *CC-CEDICT download*. from http://www.mdbg.net/chindict/chindict.php?page=cc-cedict

Sproat, Richard, and Chilin Shih. 1990. A statistical method for finding word boundaries in Chinese text. *Computer Processing of Chinese &amp; Oriental Languages, 4*(4): 336-351.

Tang, Ling-Xiang, Shlomo Geva, Yue Xu, and Andrew Trotman. 2009. *Word Segmentation for Chinese Wikipedia Using N-Gram Mutual Information*. Paper presented at the 14th Australasian Document Computing Symposium (ADCS 2009).