Can MDL Improve Unsupervised Chinese Word Segmentation?

Pierre Magistry, Benoît Sagot

Can MDL Improve Unsupervised Chinese Word Segmentation?

Pierre Magistry
Alpage, INRIA & Univ. Paris 7,
75013 Paris, France
pierre.magistry@inria.fr

Benoît Sagot
Alpage, INRIA & Univ. Paris 7,
75013 Paris, France
benoit.sagot@inria.fr

Abstract

It is often assumed that Minimum Description Length (MDL) is a good criterion for unsupervised word segmentation. In this paper, we introduce a new approach to unsupervised word segmentation of Mandarin Chinese, that leads to segmentations whose Description Length is lower than what can be obtained using other algorithms previously proposed in the literature. Suprisingly, we show that this lower Description Length does not necessarily corresponds to better segmentation results. Finally, we show that we can use very basic linguistic knowledge to coerce the MDL towards a linguistically plausible hypothesis and obtain better results than any previously proposed method for unsupervised Chinese word segmentation with minimal human effort.

1 Introduction

In Chinese script, very few symbols can be considered as word boundary markers. The only easily identifiable boundaries are sentence beginnings and endings, as well as positions before and after punctuation marks. Although the script doesn't rely on typography to define (orthographic) "words", a word-level segmentation is often required for further natural language processing. This level corresponds to minimal syntactic units that can be POS-tagged or used as input for parsing.

Without word-boundary characters, like whitespace in Latin script, there is no trivial tokenization method that can yield a good enough approximation for further processing. Therefore, the first step of many NLP systems for written Chinese is the Chinese word segmentation task.

A great variety of methods have been proposed in the literature, mostly in supervised machine learning settings. Our work addresses the question of unsupervised segmentation, i.e., without any manually segmented training data. Although supervised learning typically performs better than unsupervised learning, we believe that unsupervised systems are worth investigating as they require less human labour and are likely to be more easily adaptable to various genres, domains and time periods. They can also provide more valuable insight for linguistic studies.

Amongst the unsupervised segmentation systems described in the literature, two paradigms are often used: Branching Entropy (BE) and Minimum Description Length (MDL). The system we describe in this paper relies on both. We introduce a new algorithm which searches in a larger hypothesis space using the MDL criterion, thus leading to lower Description Lengths than other previously published systems. Still, this improvement concerning the Description Length does not come with better results on the Chinese word segmentation task, which raises interesting issues. However, it turns out that it is possible to add very simple constraints to our algorithm in order to adapt it to the specificities of Mandarin Chinese in a way that leads to results better than the state-of-the-art on the Chinese word segmentation task.

This paper is organized as follows. Section 2 describes the role of Branching Entropy in various previous works on Chinese word segmentation, including the algorithm we use as an initialisation step in this paper. In Section 3 we explain how the MDL paradigm is used amongst different Chinese word segmentation systems in the literature. We describe in Section 4 the way we use MDL for trying and improving the results of the initialisation step. A first evaluation and the error analysis given in Section 5 allow us to refine the algorithm and achieve our best results, as shown in Section 6. Finally, we discuss our findings and their implications for our futur work in Section 7.

2 Branching Entropy and Word Segmentation

2.1 The Harrissian hypothesis

Branching Entropy and its discrete counterpart, Accessor Variety are commonly used indicators of linguistically relevant boundaries.

Accessors Variety (hereafter AV) is simply the number of distinct contexts (right or left) in which a given string occurs in a corpus. Branching Entropy (hereafter BE) can be seen as a continuous version of AV that takes into account the probability distribution of cooccurrences. It is the entropy of the probability distribution of the contexts occurring on the right or on the left of a given string. Both measure the diversity of the contexts in which a string can occur.

The main idea behind the use of AV for unsupervised word segmentation was first introduced by Harris (1955) as a procedure from morpheme segmentation in phonemic transcription of speech. In 1955, Harris did not use a corpus to estimate the AV but asked native speakers of various languages how many phonemes they can think of that can follow or precede a given phoneme sequence. Harris made the hypothesis that linguistic boundaries relate with the *variation* of the AV and proposed algorithms to perform segmentation based on the data collected from native speakers. The underlying idea is the following: when given a prefix of a morpheme as input, we have a certain knowledge of what may be the next phoneme; the variety of possible continuations decreases as we add phonemes to the input string, but when reaching a linguistic boundary, the variety of what may come next suddenly increase.

2.2 Variation of Branching Entropy

Kempe (1999) adapted the method proposed by Harris to corpus linguistics and did the switch from variation of AV to variation of BE (hereafter VBE) which is a better estimation of uncertainty.

Branching Entropy (Right and Left) can be defined as follows: given an n-gram $x_{0..n} = x_{0..1}\, x_{1..2} \dots x_{n-1..n}$ with a left context $\chi_{\rightarrow}$, its *Right Branching Entropy* (RBE) $h_{\rightarrow}(x_{0..n})$ writes as

$$\begin{aligned} h_{\rightarrow}(x_{0..n}) &= H(\chi_{\rightarrow} \mid x_{0..n}) \\ &= -\sum_{x \in \chi_{\rightarrow}} P(x \mid x_{0..n}) \log P(x \mid x_{0..n}). \end{aligned}$$

The *Left Branching Entropy* (LBE) is defined symetrically: if we call $\chi_{\leftarrow}$ the right context of $x_{0..n}$, its LBE is defined as:

$$h_{\leftarrow}(x_{0..n}) = H(\chi_{\leftarrow} \mid x_{0..n}).$$

From $h_{\rightarrow}(x_{0..n})$ and $h_{\rightarrow}(x_{0..n-1})$ on the one hand, and from $h_{\leftarrow}(x_{0..n})$ and $h_{\leftarrow}(x_{1..n})$ on the other hand, we can defined the *Variation of Branching Entropy* (VBE) in both directions:

$$\begin{aligned} \delta h_{\rightarrow}(x_{0..n}) &= h_{\rightarrow}(x_{0..n}) - h_{\rightarrow}(x_{0..n-1}) \\ \delta h_{\leftarrow}(x_{0..n}) &= h_{\leftarrow}(x_{0..n}) - h_{\leftarrow}(x_{1..n}). \end{aligned}$$

2.3 Previous work on VBE-based segmentation

Several unsupervised segmentation algorithms and systems in the literature are based on BE or VBE.

Cohen et al. (2002) use BE as an indicator in their Voting Experts system. They point the need for normalisation but use BE directly, not VBE.

Jin and Tanaka-Ishii (2006) propose a system for unsupervised Chinese word segmentation based on the VBE and evaluate it against a manually segmented corpus in Mandarin Chinese.

Zhikov et al. (2010) use BE to get an initial segmentation. They put a boundary at each position that exceeds a threshold. This threshold is determined by an unsupervised procedure based on MDL. They refine this initial segmentation using two different procedures, also based on BE, which aim at minimizing the Description Length (see next section).

Wang et al. (2011) propose ESA (*Evaluation, Selection, and Adjustment*), a more complex system combining two measures of cohesion and noncohesion iteratively. The Branching Entropy is also at the root of their calculations. They achieve best published results but rely on a parameter used to balance the two measures that can be difficult to set without training data.

In Magistry and Sagot (2012), we use a normalized VBE to define a measure of the *autonomy* of a string (word candidate). The autonomy of a word candidate x is defined as $a(x) = \tilde{\delta}h_{\leftarrow}(x) + \tilde{\delta}h_{\rightarrow}(x)$ where $\tilde{\delta}h(x)$ denotes VBE normalized in order to reduce the bias related to the variation of word lengths. This autonomy function is then used in a segmentation algorithm that maximize the autonomy of all the words in a sentence. The segmentation choosen for a given sentence s

is then chosen among all possible segmentations $w \in Seg(s)$ as being

$$\arg\max_{W \in Seg(s)} \sum_{w_i \in W} a(w_i) \cdot len(w_i),$$

Our results were slightly below ESA, but the system is simpler to implement and improve on; moreover, it does not rely on any parameter for which a value must be chosen.[1]

The system presented in this paper extends both the work of Zhikov et al. (2010) and of Magistry and Sagot (2012): we rely on the notion of autonomy introduced by the latter and use it both for computing an initial segmentation and for guiding the MDL in a way inspired by the former.

3 MDL and Word Segmentation

The Minimum Description Length was introduced by Rissanen (1978). It can be considered as an approximation of the Kolmogorov complexity or as the formalisation of the principle of least effort (Zipf, 1949) by a compression model. The underlying idea behind the use of MDL for Word Segmentation is the following: once a corpus is segmented, it can be recoded as a lexicon and a sequence of references to the lexicon. A good segmentation should result in a more compact representation of the data. Probability distributions of lexical items in the corpus and Shannon entropy are used to determine the theoretically optimal compression rate we could achieve with a given segmentation.

A segmented corpus is therefore considered as a sequence of words encoded using a lexicon, or word model, M_w, which represent each word using a code that depends on its frequency: a frequent word is to be represented by a shorter code. The description length $L(C)$ of a corpus C can then be computed as the length $L(M_w)$ of the lexicon plus the length $L(D|M_w)$ of the sequence of word codes:

$$L(C) = L(D, M_w) = L(M_w) + L(D|M_w).$$

[1]With the current impletation of our algorithm presented in (Magistry and Sagot, 2012), the results are not as good as those from the previous paper. This is due to a bug in normalisation which used to include values of sentence initial and final dummy tokens. This was creating a biais in favor of one-character units and yields better scores. Our latest version of the system, which is used in this paper sticks to the definitions and is thus cleaner but does not perform as well.

The content of the lexicon can be further encoded as a sequence of characters, using a model M_c accounting for characters probability distributions in the lexicon. As a result,

$$L(M_w) = L(D_w, M_c) = L(M_c) + L(D_w|M_c).$$

$L(D|M)$ is given by:

$$L(D|M) = -\sum_{i=1}^{|M|} \#w_i \log \frac{\#w_i}{N}$$

As shown for example by Zhikov et al. (2010), it is possible to decompose this formula to allow fast update of the DL value when we change the segmentation and avoid the total computation at each step of the minimization.

MDL is often used in unsupervised segmentation systems, where it mostly plays one of the two following roles: (i) it can help selecting an optimal parameter value in an unsupervised way (Hewlett and Cohen, 2011), and (ii) it can drive the search for a more compact solution in the set of all possible segmentations.

When an unsupervised segmentation model relies on parameters, one needs a way to assign them adequate values. In a fully unsupervised setup, we cannot make use of a manually segmented corpus to compute these values. Hewlett and Cohen (2011) address this issue by choosing the set of parameters that yields the segmentation associated with the smallest DL. They show that the output corresponding to the smallest DL almost always corresponds to the best segmentation in terms of word-based f-score. In the system by Zhikov et al. (2010), the initial segmentation algorithm requires to chose a threshold: for a given position in the corpus, they mark the position as a word boundary if the BE is greater than the threshold. The value of this threshold is unsupervisingly discovered with a bisection search algorithm that looks for the smallest DL.

However, the main issue with MDL is that there is no tractable search algorithm for the whole hypothesis space. One has to rely on heuristic procedures to generate hypotheses before checking their DL. (Zhikov et al., 2010) propose two distinct procedures that they combine sequentially. The first one operate on the whole corpus. They begin by ordering all possible word-boundary positions using BE and then try to add word boundaries checking each position sorted by decreasing BE, and to

remove word boundaries checking each position by increasing order of BE. They accept any modification that will result in a smaller DL. The rationale behind this strategy is simple: for a given position, the higher the BE, the more likely it is to be a word-boundary. They process the more likely cases first. The main limitation of this procedure is that it is unable to change more than one position at a time. It will miss any optimisation that would require to change many occurrences of the same string, e.g., if the same mistake is repeated in many similar places, which is likely to happen given their initial segmentation algorithm.
To overcome this limitation, Zhikov et al. (2010) propose a second procedure that focuses on the lexicon rather than on the corpus. This procedure algorithm tries (i) to split each word of the lexicon (at each position within each word type) and reproduce this split on all occurrences of the word, and (ii) to merge all occurrences of each bi-gram in the corpus provided the merge results in an already existing word type. This strategy allows them to change multiple positions at the same time but their merging procedure is unable to discover new long types that are absent from the initial lexicon.

4 A new segmentation Algorithm based on MDL and nVBE

We propose a new strategy to reduce the DL. We use the algorithm introduced in Magistry and Sagot (2012) as an initialisation procedure followed by a DL reduction step. This step relies on an *autonomy*-driven algorithm that explores a larger part of the hypothesis space, which we shall now describe.

Given an initial segmentation of the corpus, we define a scoring function for boundary positions. As our initial procedure is based on the maximization of autonomy, any change at any position will result in a lower autonomy of the sequence. Our scoring function evaluates this loss of autonomy whenever a segmentation decision is changed. This can be viewed as similar to the ordered n-best solutions from Magistry's procedure.

The context of a boundary position is defined as a triple containing:

a position state between two characters, i.e., a boolean set to *true* if the position is a word boundary,

a prefix which is the sequence of characters running from the previous word boundary to the position,

a suffix which is the sequence of characters running from the position to the next word boundary.

When scoring a position, there are two possibilities:

- the position is currently a word boundary (we evaluate a merge),
- the position is currently not a word boundary (we evaluate a split).

In order to compute the difference in autonomy scores between the current segmentation and the one which is obtained only by performing a merge at one particular position, we simply have to subtract the autonomy of the prefix and suffix and to add the autonomy of the concatenation of the two strings.

Similarly, to evaluate a splitting decision we have to add the autonomy of the prefix and suffix and to subtract the *autonomy* of the concatenation of the two strings.

Note that with this scoring method and this definition of a context as a tuple, all occurrences of a context type will have the same score, and can therefore be grouped. We can thus evaluate the effect of changing the segmentation decision for a set of identical positions in the corpus in just one step.

Like the lexicon cleaning procedure by Zhikov et al. (2010), we can evaluate the effect of a large number of changes at the same time. But contratrily to Zhikov et al. (2010), because we process the whole corpus and not the lexicon, we have a broader search space which allows for the creation of large words even if they were previously absent from the lexicon.

A remaining issue is that changing a segmentation decision at a particular position should result in a change of the scores of all the neighbouring positions inside its *prefix* and its *suffix* and require to rebuild the whole agenda, which is a costly operation. To make our algorithm faster, we use a simplified treatment that freezes the affected positions and prevent further modification (they are simply removed from the agenda). As the agenda is sorted to test the more promising positions first (in terms of autonomy), this trade-off between exhaustiveness for speed is acceptable. Indeed, it turns out

Algorithm 4.1: algorithm1($Corpus$)

```
seg ← MagistrySagot2012(Corpus)
DL ← DescriptionLength(seg)
MinDL ← ∞
Agenda ← SortBoundaries(Corpus, seg)
while DL < MinDL
do { MinDL ← DL
     for each changes ∈ Agenda
     do { changes ← removeFrozen(changes)
          newDL = Score(changes)
          if newDL < MinDL
          then
               do { seg ← ApplyChange(changes)
                    freeze(changes)
                    DL ← newDL
                    break
```

Figure 1: DL minimization

that we still reach lower description length than Zhikov et al. (2010).

The details of our minimization of DL algorithm using this scoring method are presented in figure 4. As we shall see, this system can be further improved. We shall therefore refer to it as the *base system*.

5 Evaluation of the base system

5.1 Reference corpora

The evaluation presented here uses the corpora from the 2005 Chinese Word Segmentation Bakeoff (Emerson, 2005). These corpora are available from the bakeoff website and many previous works use them for evaluation, results are therefore easily comparable. This dataset also has the advantage of providing corpora that are segmented manually following four different guidelines. Given the lack of consensus on the definition of the minimal segmentation unit, it is interesting to evaluate unsupervised systems against multiple guidelines and data sources: since an unsupervised system is not trained to mimic a specific guideline, its output may be closer to one or another. The dataset includes data from the Peking University Corpus (PKU), from the LIVAC Corpus by Hong-Kong City-University (City-U), from Microsoft Research (MSR) and from the Balanced Corpus of the Academia Sinica (AS). It was initially intended for supervised segmentation so each corpus is divided between a training and a test set, the latter being smaller. We retain these splits in order to provide results comparable with other studies and to have an idea of the effect of the size of the training data. All the scores we provide are computed on the test set of each corpus. As our task is unsupervised segmentation, all whitespaces were of course removed from the training sets. Details about the size of the various corpora are given in Table 1.

Corpus	Words		Characters	
	Tokens	Types	Tokens	Types
AS	5 449 698	141 340	8 368 050	6 117
CITYU	1 455 629	69 085	2 403 355	4 923
PKU	1 109 947	55 303	1 826 448	4 698
MSR	2 368 391	88 119	4 050 469	5 167

Table 1: Size of the different corpora

5.2 Evaluation Metrics

The metric used for all following evaluations is a standard f-score on words. It is the harmonic mean of the word recall

$$R_w = \frac{\text{\#correct words in the results}}{\text{\#words in the gold corpus}}$$

and the word precision

$$P_w = \frac{\text{\#correct words in the result}}{\text{\#words in the result}},$$

which leads to the following:

$$F_w = \frac{2 \times R_w \times P_w}{R_w + P_w}$$

For each corpus and method, we also present the Description Length of each segmentation.

Note that, as mentioned by several studies (Huang and Zhao, 2007; Magistry and Sagot, 2012; Sproat and Shih, 1990), the agreement between the different guidelines and even between untrained native speakers is not high. Using cross-trained supervized systems or inter-human agreement, these studies suggest that the topline for unsupervised segmentation is beetween 0.76 and 0.85. As a result, not only the output of an unsupervised system cannot be expected to perfectly mimic a given "gold" segmented corpus, but performances around 0.80 against multiple "gold" segmented corpora using different guidelines can be regarded as satisfying.

5.3 Results

The results of our base system, without and with our MDL step, are presented in Table 2. We also

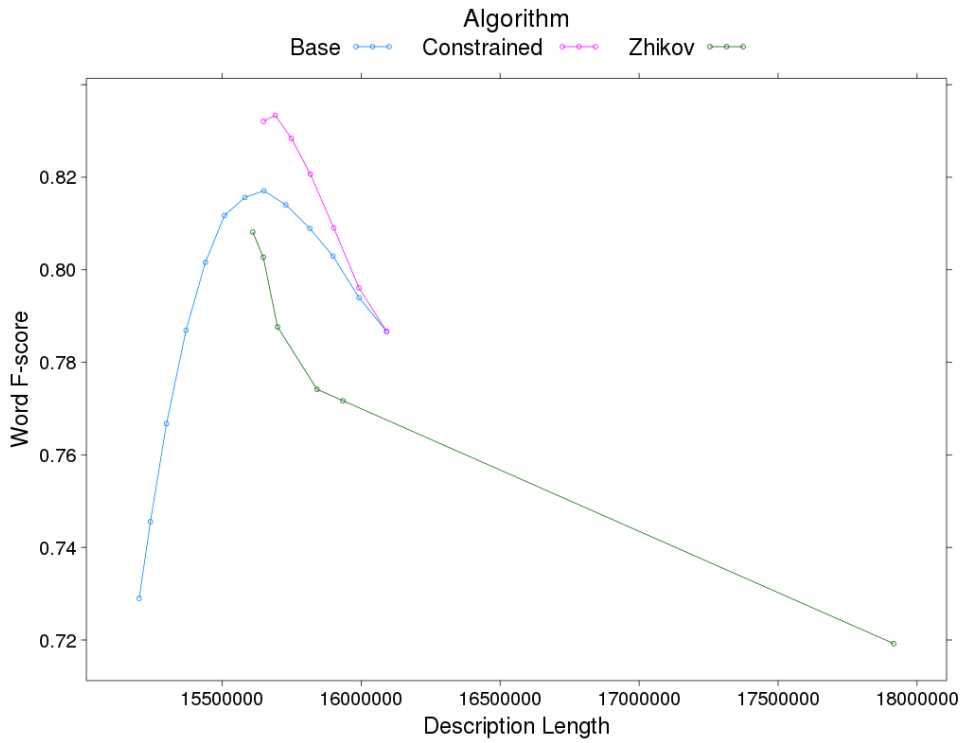

Figure 2: f-score on words as a function of description length for the three algorithms

provide results for our re-implementation of the algorithm by Zhikov et al. (2010), without and with their own MDL step. Our initialisation (without our MDL step) obtains very good results; on the MSR corpus, they are even as high as the results of Zhikov *et al.*'s full algorithm, including their MDL step. However, at a first sight, the results we get when using our MDL procedure are disappointing: it sometimes worsen the results of the initisalisation step. However, we observe that our MDL step succesfully decreases the Description Lengths obtained after the initialisation step, and leads to Description Lengths lower than Zhikov *et al.*'s system although with lower f-scores. This tackles the common idea that lower Description Length yields better segmentation, and calls for further analysis.

5.4 Step-by-step MDL results

In both systems, ours and Zhikov *et al.*'s, the MDL algorithm is iterative. We therefore decided to dump intermediary results at each iteration to observe the evolution of the segmentation quality as the DL gets smaller. Figure 5.3 shows the resulting f-scores as a function of the DL at different stages, on the PKU corpus (results on other corpora behave similarly). Each iteration of one MDL algorithm or the other reduces the DL, which means that a given curve on this graphic are followed by the corresponding system step after step from right to left. The leftmost dot on each curve corresponds to the point when the corresponding system decides to stop and produce its final output.

This graphic shows that our system produces better segmentation at some point, outperforming Zhikov *et al.*'s system. But it doesn't stop at that point and the f-score drops as the DL continue to decrease. This seems to mean that our algorithm, because it explores a larger search space, manages to find segmentations that are optimal as far as DL is concerned, but that do not constitute optimal word-level segmentation.

In order to better understand what is going on, we have added a logging functionality to our implementations, so we can check which operations are made when the f-score decreases. We shall now discuss several typical examples thereof.

5.5 Error analysis

A sample of the latest modifications made by our system while the f-score is falling is given in Table 3. We show the modification that are applied to the largest numbers of occurrences. The type of operation is either a merge (suppression of a boundary) or a split (adding a boundary). We pro-

Method	f-score	DL (Mb)
PKU corpus		
Zhikov *et al.* (no MDL)	0.719	17.9
Zhikov *et al.* (with their MDL)	**0.808**	15.6
This paper (no MDL)	0.786	16.1
This paper (with our base MDL)	0.729	**15.2**
Gold	*1.0*	*15.0*
City-U corpus		
Zhikov *et al.* (no MDL)	0.652	23.2
Zhikov *et al.* (with their MDL)	**0.787**	19.8
This paper (no MDL)	0.744	20.3
This paper (with our base MDL)	0.754	**19.3**
Gold	*1.0*	*19.0*
MSR corpus		
Zhikov *et al.* (no MDL)	0.690	37.1
Zhikov *et al.* (with their MDL)	**0.782**	31.9
This paper (no MDL)	**0.782**	33.0
This paper (with our base MDL)	0.690	**31.1**
Gold	*1.0*	*30.8*
AS Corpus		
Zhikov *et al.* (no MDL)	0.614	80.8
Zhikov *et al.* (with their MDL)	**0.762**	67.1
This paper (no MDL)	0.758	68.9
This paper (with our base MDL)	0.711	**65.7**
Gold	*1.0*	*65.3*

Table 2: Scores on different Corpora for Zhikov *et al.*'s (2010) algorithm (without and with their MDL-based improvement step) and for our base system (without MDL and with our base MDL algorithm). Final results are displayed in Table 6

vide the prefix and suffix, whether the merge or split is an error or not, as well as English glosses.

The first observation we make is that amongst highly frequent items, our system only performs merges. Splits are indeed performed on a large number of rare types for which both the prefix and the suffix exist in the lexicon. We note that for bigrams, such splits are almost always an erroneous decision.

Merge operations include valid decisions, erroneous decisions producing multi-word expression units (MWE), and erroneous decisions that merge a grammatical word to one of its collocations.

6 Description and evaluation of our constrained system

Given this error analysis, there are three main types of common mistakes that we would like to avoid:

- merging MWEs such as named entities;
- merging function words with content words when the co-occurrence is frequent;

Operation	String	Evaluation
merge	的 . 发展 DE - development	error
merge	据 . 新华社 According to - Xinhua Agency	error
merge	新华社 . 北京 Xinhua Agency - Peking	error
merge	经济 . 发展 economic - growth	error (MWE)
merge	进行 . 了 conduct - LE (-ed)	error
merge	和 . 发展 AND - development	error
merge	在 . 北京 AT - Peking	error
merge	邓小平 . 理论 Deng Xiaoping - Theories	error (MWE)
merge	领导 . 干部 leading - cadre	error (MWE)
merge	常 . 委会 standing - committee	error (MWE)
merge	改革 . 开放 reform and opening	error (MWE)
merge	反 . 腐败 anti - corruption	correct
merge	节 . 日 holi-day	correct
merge	党 . 中央 central committee	correct
merge	金融 . 危机 finance - crisis	error (MWE)
merge	新 . 世纪 new - century	error
merge	副 . 总理 vice - premier	correct
merge	国民 . 经济 national - economy	error (MWE)
merge	北京 . 市 Peking - city	no
merge	基础 . 上 basis - postposition (=basically)	error
merge	副 . 主席 vice-chairman	correct
merge	结构 . 调整 structural adjustment	error (MWE)
merge	产业 . 化 industrial - ize	correct
merge	现代化 . 建设 modernization - drive	error (MWE)
merge	人 . 大 Acronym for Renmin University	correct

Table 3: Modification made (sorted by number of occurrences)

Method	f-score	DL (Mb)
PKU corpus		
Zhikov *et al.* (with their MDL)	0.808	15.6
This paper (with constrained MDL)	**0.832**	15.6
Gold	*1.0*	*15.0*
City-U corpus		
Zhikov *et al.* (with their MDL)	0.787	19.8
This paper (with constrained MDL)	**0.801**	19.8
Gold	*1.0*	*19.0*
MSR corpus		
Zhikov *et al.* (with their MDL)	0.782	31.9
This paper (with constrained MDL)	**0.809**	32.1
Gold	*1.0*	*30.8*
AS Corpus		
Zhikov *et al.* (with their MDL)	0.762	67.1
This paper (with constrained MDL)	**0.795**	67.3
Gold	*1.0*	*65.3*

Table 4: Final results

- splitting bigrams that were correct in the initial segmentation.

If we give up on having a strictly language-independent system and focus on Mandarin Chinese segmentation, these three issues are easy to address with a fairly low amount of human work to add some basic linguistic knowledge about Chinese to the system.

The first issue can be dealt with by limiting the length of a merge's output. A MWE will be larger than a typical Chinese word that very rarely exceeds 3 characters. With the exception of phonetic loans for foreign languages, larger units typically correspond to MWE that are segmented in the various gold corpora.[2] The question whether it is a good thing to do or not will be raised in the discussion section, but for a higher f-score on word segmentation, leaving them segmented does help.

The second issue can be addressed using a closed list of function words such as aspectual markers and pre/post-positions. As those are a closed list of items, listing all of them is an easily manually tractable task. Here is the list we used in our experiments:
的、了、上、在、下、中、是、有、和、与、和、就、多、于、很、才、跟

As for the third issue, since Chinese is known to favour bigram words, we simply prevent our system to split those.

[2]A noticeable exception are the 4-characters idioms (chengyu) but they seem less frequent than 2+2 multiword expressions.

We implemented these three constraints to restrict the search space for our minimization of the Description Length an re-run the experiments. Results are presented in the next section.

6.1 Evaluation of the constrained system

The scores obtained by our second system are given in Table 6. They show a large improvement on our initial segmentation and outperform previously reported results.

7 Discussion and futur work

The results presented in this paper invite for discussion. It is well accepted in the literature that MDL is a good indicator to find better segmentation but our results show that it is possible to reach a lower description length without improving the segmentation score. However, this paper also demonstrates that MDL can still be a relevant criterion when its application is constrained using very simple and almost zero-cost linguistic information.

The constraints we use reflect two underlying linguistic phenomena. The first one is related to what would be called "multi-word expressions" (MWE) in other scripts. It is unclear whether it is a limitation of the segmentation system or a problem with the definition of the task. There is a growing interest for MWE in the NLP community. Their detection is still challenging for all languages, but has already been proven useful for deeper analysis such as parsing. It is somewhat frustrating to have to prevent the detection of multi-words expressions to achieve better segmentation results.

The second restriction concerns the distinction between content words and grammatical words. It is not so surprising that open and closed classes of words show different distributions and deserve specific treatments. From a practical point of view, it is worth noting that MDL is useful for open classes where manual annotation or rule-based processing are costly if even possible. On the other hand, rules are helpful for small closed classes and represent a task that is tractable for human, even when facing the need to process a large variety of sources, genres or topics. This division of labour is acceptable for real-world applications when no training data is available for supervised systems.

References

Paul Cohen, Brent Heeringa, and Niall Adams. 2002. An unsupervised algorithm for segmenting categorical timeseries into episodes. *Pattern Detection and Discovery*, page 117–133.

Thomas Emerson. 2005. The second international chinese word segmentation bakeoff. In *Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing*, volume 133.

Zellig S. Harris. 1955. From phoneme to morpheme. *Language*, 31(2):190–222.

Daniel Hewlett and Paul Cohen. 2011. Fully unsupervised word segmentation with bve and mdl. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers*, volume 2, pages 540--545.

Changning. Huang and Hai Zhao. 2007. 中文分词十年回顾 (Chinese word segmentation: A decade review). *Journal of Chinese Information Processing*, 21(3):8–20.

Zhihui Jin and Kumiko Tanaka-Ishii. 2006. Unsupervised segmentation of Chinese text by use of branching entropy. In *Proceedings of the COLING/ACL on Main conference poster sessions*, page 428–435.

André Kempe. 1999. Experiments in unsupervised entropy-based corpus segmentation. In *Workshop of EACL in Computational Natural Language Learning*, page 7–13.

Pierre Magistry and Benoît Sagot. 2012. Unsupervized word segmentation: the case for mandarin chinese. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2*, pages 383--387. Association for Computational Linguistics.

Jorma Rissanen. 1978. Modeling by shortest data description. *Automatica*, 14(5):465--471.

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

Hanshi Wang, Jian Zhu, Shiping Tang, and Xiaozhong Fan. 2011. A new unsupervised approach to word segmentation. *Computational Linguistics*, 37(3): 421–454.

Valentin Zhikov, Hiroya Takamura, and Manabu Okumura. 2010. An efficient algorithm for unsupervised word segmentation with branching entropy and mdl. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 832--842. Association for Computational Linguistics.

George Kingsley Zipf. 1949. *Human Behaviour and the Principle of Least-Effort*. Addison-Wesley.