Fully Unsupervised Word Segmentation with BVE and MDL

Daniel Hewlett and **Paul Cohen**
Department of Computer Science
University of Arizona
Tucson, AZ 85721
{dhewlett,cohen}@cs.arizona.edu

Abstract

Several results in the word segmentation literature suggest that description length provides a useful estimate of segmentation quality in fully unsupervised settings. However, since the space of potential segmentations grows exponentially with the length of the corpus, no tractable algorithm follows directly from the Minimum Description Length (MDL) principle. Therefore, it is necessary to generate a set of candidate segmentations and select between them according to the MDL principle. We evaluate several algorithms for generating these candidate segmentations on a range of natural language corpora, and show that the Bootstrapped Voting Experts algorithm consistently outperforms other methods when paired with MDL.

1 Introduction

The goal of unsupervised word segmentation is to discover correct word boundaries in natural language corpora where explicit boundaries are absent. Often, unsupervised word segmentation algorithms rely heavily on parameterization to produce the correct segmentation for a given language. The goal of fully unsupervised word segmentation, then, is to recover the correct boundaries for arbitrary natural language corpora without explicit human parameterization. This means that a fully unsupervised algorithm would have to set its own parameters based only on the corpus provided to it.

In principle, this goal can be achieved by creating a function that measures the quality of a segmentation in a language-independent way, and applying this function to all possible segmentations of the corpora to select the best one. Evidence from the word segmentation literature suggests that description length provides a good approximation to this segmentation quality function. We discuss the Minimum Description Length (MDL) principle in more detail in the next section. Unfortunately, evaluating all possible segmentations is intractable, since a corpus of length n has 2^{n-1} possible segmentations. As a result, MDL methods have to rely on an efficient algorithm to generate a relatively small number of candidate segmentations to choose between. It is an empirical question which algorithm will generate the most effective set of candidate segmentations. In this work, we compare a variety of unsupervised word segmentation algorithms operating in conjunction with MDL for fully unsupervised segmentation, and find that the Bootstrapped Voting Experts (BVE) algorithm generally achieves the best performance.

2 Minimum Description Length

At a formal level, a segmentation algorithm is a function $\textsc{Segment}(c, \theta)$ that maps a corpus c and a vector of parameters $\theta \in \Theta$ to one of the possible segmentations of that corpus. The goal of fully unsupervised segmentation is to reduce $\textsc{Segment}(c, \theta)$ to $\textsc{Segment}(c)$ by removing the need for a human to specify a particular θ. One way to achieve this goal is to generate a set of candidate segmentations by evaluating the algorithm for multiple values of θ, and then choose the segmentation that minimizes some cost function. Thus, we can define $\textsc{Segment}(c)$ in terms of $\textsc{Segment}(c, \theta)$:

$$\textsc{Segment}(c) = \underset{\theta \in \Theta}{\operatorname{argmin}} \textsc{Cost}(\textsc{Segment}(c, \theta)) \quad (1)$$

Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics:shortpapers, pages 540–545,
Portland, Oregon, June 19-24, 2011.

Now, selecting the best segmentation is treated as a model selection problem, where each segmentation provides a different model of the corpus. Intuitively, a general approach is to choose the simplest model that explains the data, a principle known as Occam's Razor. In information theory, this intuitive principle of simplicity or parsimony has been formalized as the Minimum Description Length (MDL) principle, which states that the most likely model of the data is the one that requires the fewest bits to encode (Rissanen, 1983). The number of bits required to represent a model is called its *description length*. Previous work applying the MDL principle to segmentation (Yu, 2000; Argamon et al., 2004; Zhikov et al., 2010) is motivated by the observation that every segmentation of a corpus implicitly defines a *lexicon*, or set of words.

More formally, the segmented corpus S is a list of words $s_1 s_2 \ldots s_N$. $L(S)$, the lexicon implicitly defined by S, is simply the set of unique words in S. The description length of S can then be broken into two components, the description length of the lexicon and the description length of the corpus given the lexicon. If we consider S as being generated by sampling words from a probability distribution over words in the lexicon, the number of bits required to represent each word s_i in S is simply its surprisal, $-\log P(s_i)$. The information cost of the corpus given the lexicon is then computed by summing the surprisal of each word s_i in the corpus:

$$\textsc{Code}(S|L(S)) \quad = \quad -\sum_{i=1}^{N} \log P(s_i) \qquad (2)$$

To properly compute the description length of the segmentation, we must also include the cost of the lexicon. Adding in the description length of the lexicon forces a trade-off between the lexicon size and the size of the compressed corpus. For purposes of the description length calculation, the lexicon is simply treated as a separate corpus consisting of characters rather than words. The description length can then be computed in the usual manner, by summing the surprisal of each character in each word in the lexicon:

$$\textsc{Code}(L(S)) = -\sum_{w \in L(S)} \sum_{k \in w} \log P(k) \qquad (3)$$

where $k \in w$ refers to the characters in word w in the lexicon. As noted by Zhikov et al. (Zhikov et al., 2010), an additional term is needed for the information required to encode the parameters of the lexicon model. This quantity is normally estimated by $(k/2) \log n$, where k is the degrees of freedom in the model and n is the length of the data (Rissanen, 1983). Substituting the appropriate values for the lexicon model yields:

$$\frac{|L(S)| - 1}{2} * \log N \qquad (4)$$

The full description length calculation is simply the sum of three terms shown in 2, 3, and 4. From this definition, it follows that a low description length will be achieved by a segmentation that defines a small lexicon, which nonetheless reduces the corpus to a short series of mostly high-frequency words.

3 Generating Candidate Segmentations

Recent unsupervised MDL algorithms rely on heuristic methods to generate candidate segmentations. Yu (2000) makes simplifying assumptions about the nature of the lexicon, and then performs an Expectation-Maximization (EM) search over this reduced hypothesis space. Zhikov et al. (2010) present an algorithm called EntropyMDL that generates a candidate segmentation based on branching entropy, and then iteratively refines the segmentation in an attempt to greedily minimize description length.

We selected three entropy-based algorithms for generating candidate segmentations, because such algorithms do not depend on the details of any particular language. By "unsupervised," we mean operating on a single unbroken sequence of characters without any boundary information; Excluded from consideration are a class of algorithms that are semi-supervised because they require sentence boundaries to be provided. Such algorithms include MBDP-1 (Brent, 1999), HDP (Goldwater et al., 2009), and WordEnds (Fleck, 2008), each of which is discussed in Section 5.

3.1 Phoneme to Morpheme

Tanaka-Ishii and Jin (2006) developed Phoneme to Morpheme (PtM) to implement ideas originally developed by Harris (1955). Harris noticed that if one proceeds incrementally through a sequence of phonemes and asks speakers of the language to count the letters that could appear next in the sequence (today called the *successor count*), the points where the number *increases* often correspond to morpheme boundaries. Tanaka-Ishii and Jin cor-

rectly recognized that this idea was an early version of branching entropy, given by $H_B(seq) = -\sum_{c \in S} P(c|seq) \log P(c|seq)$, where S is the set of successors to seq. They designed their PtM algorithm based on branching entropy in both directions, and it was able to achieve scores near the state of the art on word segmentation in phonetically-encoded English and Chinese. PtM posits a boundary whenever the increase in the branching entropy exceeds a threshold. This threshold provides an adjustable parameter for PtM, which we exploit to generate 41 candidate segmentations by trying every threshold in the range $[0.0, 2.0]$, in steps of 0.05.

3.2 Voting Experts

The Voting Experts (VE) algorithm (Cohen and Adams, 2001) is based on the premise that words may be identified by an information theoretic signature: Entropy within a word is relatively low, entropy at word boundaries is relatively high. The name *Voting Experts* refers to the "experts" that vote on possible boundary locations. VE has two experts: One votes to place boundaries after sequences that have low internal entropy (surprisal), given by $H_I(seq) = -\log P(seq)$, the other votes after sequences that have high branching entropy. All sequences are evaluated locally, within a sliding window, so the algorithm is very efficient. A boundary is generated whenever the vote total at a given location exceeds a threshold, and in some cases only if the vote total is a local maximum. VE thus has three parameters that can be manipulated to generate potential segmentations: Window size, threshold, and local maximum. Pairing VE with MDL was first examined by Hewlett and Cohen (2009). We generated a set of 104 segmentations by trying every viable threshold and local max setting for each window size between 2 and 9.

3.3 Bootstrapped Voting Experts

The Bootstrapped Voting Experts (BVE) algorithm (Hewlett and Cohen, 2009) is an extension to VE. BVE works by segmenting the corpus repeatedly, with each new segmentation incorporating knowledge gained from previous segmentations. As with many bootstrapping methods, three essential components are required: some initial seed knowledge, a way to represent knowledge, and a way to leverage that knowledge to improve future performance. For BVE, the seed knowledge consists of a high-precision segmentation generated by VE. After this seed segmentation, BVE segments the corpus repeatedly, lowering the vote threshold with each iteration. Knowledge gained from prior segmentations is represented in a data structure called the *knowledge trie*. During voting, this knowledge trie provides statistics for a third expert that places votes in contexts where boundaries were most frequently observed during the previous iteration. Each iteration of BVE provides a candidate segmentation, and executing BVE for window sizes 2-8 and both local max settings generated a total of 126 segmentations.

4 Experiments

There are two ways to evaluate the quality of a segmentation algorithm in the MDL framework. The first is to directly measure the quantity of the segmentation chosen by MDL. For word segmentation, this is typically done by computing the F-score, where $F = (2 * \text{Precision} * \text{Recall})/(\text{Precision} + \text{Recall})$, for both boundaries (BF) and words (WF) found by the algorithm. The second is to compare the minimal description length among the candidates to the true description length of the corpus.

4.1 Results

We chose a diverse set of natural language corpora, including some widely-used corpora to facilitate comparison. For each corpus, we generated a set of candidate segmentations with PtM, VE, and BVE, as described in the previous section. From each set of candidates, results for the segmentation with minimal description length are presented in the tables below. Where possible, results for other algorithms are presented in italics, with semi-supervised algorithms set apart. Source code for all algorithms evaluated here, as well as data files for all corpora, are available online[1].

One of the most commonly-used benchmark corpora for unsupervised word segmentation is the BR87 corpus. This corpus is a phonemic encoding of the Bernstein Ratner corpus (Bernstein Ratner, 1987) from the CHILDES database of child-directed speech (MacWhinney, 2000). The perfor-

[1] http://code.google.com/p/voting-experts

mance of the algorithms on BR87 is shown in Table 1 below. As with all experiments in this work, the input was presented as one continuous sequence of characters with no word or sentence boundaries. Published results for two unsupervised algorithms, the MDL-based algorithm of Yu (2000) and the EntropyMDL (EMDL) algorithm of Zhikov et al. (2010), on this widely-used benchmark corpus are shown in italics. Set apart in the table are published results for three semi-supervised algorithms, MBDP-1 (Brent, 1999), HDP (Goldwater, 2007), and WordEnds (Fleck, 2008), described in Section 5. These algorithms operate on a version of the corpus that includes sentence boundaries.

Algorithm	BP	BR	BF	WP	WR	WF
PtM+MDL	0.861	**0.897**	0.879	0.676	0.704	0.690
VE+MDL	0.875	0.803	0.838	0.614	0.563	0.587
BVE+MDL	**0.949**	0.879	**0.913**	**0.793**	**0.734**	**0.762**
Yu	0.722	0.724	0.723	NR	NR	NR
EMDL	NR	NR	0.907	NR	NR	0.750
MBDP-1	0.803	0.843	0.823	0.670	0.694	0.682
HDP	0.903	0.808	0.852	0.752	0.696	0.723
WordEnds	0.946	0.737	0.829	NR	NR	0.707

Table 1: Results for the BR87 corpus.

Results for one corpus, the first 50,000 characters of George Orwell's *1984*, have been reported in nearly every VE-related paper. It thus provides a good opportunity to compare to the other VE-derived algorithms: Hierarchical Voting Experts – 3 Experts (Miller and Stoytchev, 2008) and Markov Experts (Cheng and Mitzenmacher, 2005). Table 2 shows the results for candidate algorithms as well as the two other VE-derived algorithms, HVE-3E and ME.

Algorithm	BP	BR	BF	WP	WR	WF
PtM+MDL	0.694	**0.833**	0.758	0.421	0.505	0.459
VE+MDL	0.788	0.774	0.781	0.498	0.489	0.493
BVE+MDL	**0.841**	0.828	**0.834**	**0.585**	**0.577**	**0.581**
HVE-3E	0.796	0.771	0.784	0.512	0.496	0.504
ME	0.809	0.787	0.798	NR	0.542	NR

Table 2: Results for the first 50,000 characters of *1984*.

Chinese and Thai are both commonly written without spaces between words, though some punctuation is often included. Because of this, these languages provide an excellent real-world challenge for unsupervised segmentation. The results shown in Table 3 were obtained using the first 100,000 words of the Chinese Gigaword corpus (Huang, 2007), written in Chinese characters. The word boundaries specified in the Chinese Gigaword Corpus were used as a gold standard. Table 4 shows results for a roughly 100,000 word subset of a corpus of Thai novels written in the Thai script, taken from a recent Thai word segmentation competition, InterBEST 2009. Working with a similar but much larger corpus of Thai text, Zhikov et al. were able to achieve slightly better performance (BF=0.934, WF=0.822).

Algorithm	BP	BR	BF	WP	WR	WF
PtM+MDL	0.894	0.610	0.725	0.571	0.390	0.463
VE+MDL	**0.871**	0.847	0.859	**0.657**	0.639	0.648
BVE+MDL	0.834	**0.914**	**0.872**	0.654	**0.717**	**0.684**

Table 3: Results for a corpus of orthographic Chinese.

Algorithm	BP	BR	BF	WP	WR	WF
PtM+MDL	0.863	0.934	0.897	0.702	0.760	0.730
VE+MDL	**0.916**	0.837	0.874	0.702	0.642	0.671
BVE+MDL	0.889	**0.969**	**0.927**	**0.767**	**0.836**	**0.800**

Table 4: Results for a corpus of orthographic Thai.

The Switchboard corpus (Godfrey and Holliman, 1993) was created by transcribing spontaneous speech, namely telephone conversations between English speakers. Results in Table 5 are for a roughly 64,000 word section of the corpus, transcribed orthographically.

Algorithm	BP	BR	BF	WP	WR	WF
PtM+MDL	0.761	0.837	0.797	0.499	0.549	0.523
VE+MDL	0.779	**0.855**	0.815	0.530	0.582	0.555
BVE+MDL	**0.890**	0.818	**0.853**	**0.644**	**0.592**	**0.617**
Yu	0.674	0.665	0.669	NR	NR	NR
WordEnds	0.900	0.755	0.821	NR	NR	0.663
HDP	0.731	0.924	0.816	NR	NR	0.636

Table 5: Results for a subset of the Switchboard corpus.

4.2 Description Length

Table 6 shows the best description length achieved by each algorithm for each of the test corpora. In most cases, BVE compressed the corpus more than VE, which in turn achieved better compression than PtM. In Chinese, the two VE-algorithms were able to compress the corpus beyond the gold standard

size, which may mean that these algorithms are sometimes finding repeated units larger than words, such as phrases.

Algorithm	BR87	Orwell	SWB	CGW	Thai
PtM+MDL	3.43e5	6.10e5	8.79e5	1.80e6	1,23e6
VE+MDL	3.41e5	5.75e5	8.24e5	1.54e6	1.23e6
BVE+MDL	3.13e5	5.29e5	7.64e5	1.56e6	1.13e6
Gold Standard	2.99e5	5.07e5	7.06e5	1.62e6	1.11e6

Table 6: Best description length achieved by each algorithm compared to the actual description length of the corpus.

5 Related Work

The algorithms described in Section 3 are all relatively recent algorithms based on entropy. Many algorithms for computational morphology make use of concepts similar to branching entropy, such as successor count. The HubMorph algorithm (Johnson and Martin, 2003) adds all known words to a trie and then performs DFA minimization (Hopcroft and Ullman, 1979) to convert the trie to a finite state machine. In this DFA, it searches for sequences of states (*stretched hubs*) with low branching factor internally and high branching factor at the boundaries, which is analogous to the chunk signature that drives VE and BVE, as well as the role of branching entropy in PtM.

MDL is analogous to Bayesian inference, where the information cost of the model $\textsc{Code}(M)$ acts as the prior distribution over models $P(M)$, and $\textsc{Code}(D|M)$, the information cost of the data given the model, acts as the likelihood function $P(D|M)$. Thus, Bayesian word segmentation methods may be considered related as well. Indeed, one of the early Bayesian methods, MBDP-1 (Brent, 1999) was adapted from an earlier MDL-based method. Venkataraman (2001) simplified MBDP-1, relaxed some of its assumptions while preserving the same level of performance. Recently, Bayesian methods with more sophisticated language models have been developed, including one that models language generation as a hierarchical Dirichlet process (HDP), in order to incorporate the effects of syntax into word segmentation (Goldwater et al., 2009). Another recent algorithm, WordEnds, generalizes information about the distribution of characters near word boundaries to improve segmentation (Fleck, 2008), which is analogous to the role of the knowledge trie in BVE.

6 Discussion

For the five corpora tested above, BVE achieved the best performance in conjunction with MDL, and also achieved the lowest description length. We have shown that the combination of BVE and MDL provides an effective approach to unsupervised word segmentation, and that it can equal or surpass semi-supervised algorithms such as MBDP-1, HDP, and WordEnds in some cases.

All of the languages tested here have relatively few morphemes per word. One area for future work is a full investigation of the performance of these algorithms in polysynthetic languages such as Inuktitut, where each word contains many morphemes. It is likely that in such languages, the algorithms will find morphs rather than words.

Acknowledgements

This work was supported by the Office of Naval Research under contract ONR N00141010117. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the ONR.

References

Shlomo Argamon, Navot Akiva, Amihood Amir, and Oren Kapah. 2004. Efficient Unsupervised Recursive Word Segmentation Using Minimum Description Length. In *Proceedings of the 20th International Conference on Computational Linguistics*, Morristown, NJ, USA. Association for Computational Linguistics.

Nan Bernstein Ratner, 1987. *The phonology of parent-child speech*, pages 159–174. Erlbaum, Hillsdale, NJ.

Michael R. Brent. 1999. An Efficient, Probabilistically Sound Algorithm for Segmentation and Word Discovery. *Machine Learning*, (34):71–105.

Jimming Cheng and Michael Mitzenmacher. 2005. The Markov Expert for Finding Episodes in Time Series. In *Proceedings of the Data Compression Conference*, pages 454–454. IEEE.

Paul Cohen and Niall Adams. 2001. An algorithm for segmenting categorical time series into meaningful episodes. In *Proceedings of the Fourth Symposium on Intelligent Data Analysis*.

Margaret M. Fleck. 2008. Lexicalized phonotactic word segmentation. In *Proceedings of The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 130–138, Columbus, Ohio, USA. Association for Computational Linguistics.

John J. Godfrey and Ed Holliman. 1993. Switchboard- 1 Transcripts.

Sharon Goldwater, Thomas L Griffiths, and Mark Johnson. 2009. A Bayesian Framework for Word Segmentation: Exploring the Effects of Context. *Cognition*, 112(1):21–54.

Sharon Goldwater. 2007. *Nonparametric Bayesian models of lexical acquisition*. Ph.D. dissertation, Brown University.

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

Daniel Hewlett and Paul Cohen. 2009. Bootstrap Voting Experts. In *Proceedings of the Twenty-first International Joint Conference on Artificial Intelligence*.

J. E. Hopcroft and J. D. Ullman. 1979. *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley.

Chu-Ren Huang. 2007. *Tagged Chinese Gigaword (Catalog LDC2007T03)*. Linguistic Data Consortium, Philadephia.

Howard Johnson and Joel Martin. 2003. Unsupervised learning of morphology for English and Inuktitut. *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology (HLT-NAACL 2003)*, pages 43–45.

Brian MacWhinney. 2000. *The CHILDES project: Tools for analyzing talk*. Lawrence Erlbaum Associates, Mahwah, NJ, 3rd editio edition.

Matthew Miller and Alexander Stoytchev. 2008. Hierarchical Voting Experts: An Unsupervised Algorithm for Hierarchical Sequence Segmentation. In *Proceedings of the 7th IEEE International Conference on Development and Learning*, pages 186–191.

Jorma Rissanen. 1983. A Universal Prior for Integers and Estimation by Minimum Description Length. *The Annals of Statistics*, 11(2):416–431.

Kumiko Tanaka-Ishii and Zhihui Jin. 2006. From Phoneme to Morpheme: Another Verification Using a Corpus. In *Proceedings of the 21st International Conference on Computer Processing of Oriental Languages*, pages 234–244.

Anand Venkataraman. 2001. A procedure for unsupervised lexicon learning. In *Proceedings of the Eighteenth International Conference on Machine Learning*.

Hua Yu. 2000. Unsupervised Word Induction using MDL Criterion. In *Proceedings of the International Symposium of Chinese Spoken Language Processing*, Beijing, China.

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, Cambridge, MA. MIT Press.