Unsupervised Learning of Word Boundary with Description Length Gain

Chunyu Kit[†‡]
Dept. of Chinese, Translation and Linguistics
City University of Hong Kong[†]
ctckit@cityu.edu.hk

Yorick Wilks[‡]
Department of Computer Science
University of Sheffield[‡]
yorick@dcs.shef.ac.uk

Abstract

This paper presents an unsupervised approach to lexical acquisition with the goodness measure *description length gain* (DLG) formulated following classic information theory within the *minimum description length* (MDL) paradigm. The learning algorithm seeks for an optimal segmentation of an utterance that maximises the description length gain from the individual segments. The resultant segments show a nice correspondence to lexical items (in particular, words) in a natural language like English. Learning experiments on large-scale corpora (e.g., the Brown corpus) have shown the effectiveness of both the learning algorithm and the goodness measure that guides that learning.

1. Introduction

Detecting and handling unknown words properly has become a crucial issue in today's practical natural language processing (NLP) technology. No matter how large the dictionary that is used in a NLP system, there can be many new words in running/real texts, e.g., in scientific articles, newspapers and Web pages, that the dictionary does not include. Many such words are proper names and special terminology that provide critical information. It is unreliable to rest on delimiters such as white spaces to detect new lexical units, because many basic lexical items contain one or more spaces, e.g., as in "New York", "Hong Kong" and "hot dog". It appears that unsupervised learning techniques are necessary in order to alleviate the problem of unknown words in the NLP domain.

There have been a number of studies on lexical acquisition from language data of different types. Wolff attempts to infer word boundaries from artificially-generated natural language sentences, heavily relying on the co-occurrence frequency of adjacent characters [Wolff 1975, Wolff 1977]. Nevill-Manning's text compression program Sequitur can also identify word boundaries and gives a binary tree structure for an identified word [Nevill-Manning 1996]. de Marcken explores unsupervised lexical acquisition from English spoken and written corpora and from a Chinese written corpus [de Marken 1995, de Marken 1996].

In this paper, we present an unsupervised approach to lexical acquisition within the *minimum description length* (MDL) paradigm [Rissanen 1978, Rissanen 1982] [Rissanen 1989], with a goodness measure, namely, the *description length gain* (DLG), which is formulated in [Kit 1998] following classic information theory [Shannon 1948, Cover and Thomas 1991]. This measure is used, following the MDL principle, to evaluate the goodness of identifying a (sub)sequence of characters in a corpus as a lexical item. In order to rigorously evaluate the effectiveness of this unsupervised learning approach, we do not limit ourselves to the detection of unknown words with respect to any given dictionary. Rather, we use it to perform unsupervised lexical acquisition from large-scale English text corpora. Since it is a learning-via-compression approach, the algorithm can be further extended to deal with text compression and, very likely, other data sequencing problems.

The rest of the paper is organised as follows: Section 2 presents the formulation of the DLG measure in terms of classic information theory; Section 3 formulates the learning algorithm within the MDL framework, which aims to achieve an optimal segmentation of the given corpus into lexical items with regard to the DLG measure; Section 4 presents experiments and discusses experimental results with respect to previous studies; and finally, the conclusions of the paper are given in Section 5.

2. Description Length Gain

Kit defines the *description length* of a corpus $X = x_1 x_2 \cdots x_n$, a sequence of linguistic tokens (e.g., characters, words, POS tags), as the Shannon-Fano code length for the corpus [Kit 1998]. Following classic information theory [Shannon 1948, Cover and Thomas 1991], it can be formulated in terms of token counts in the corpus as below for empirical calculation:

$$\begin{aligned} DL(X) &= n\hat{H}(X) \\ &= -n\sum_{x\in V}\hat{p}(x)\log\hat{p}(x) \\ &= -\sum_{x\in V}c(x)\log\frac{c(x)}{|X|} \end{aligned} \qquad (1)$$

where V is the set of distinct tokens (i.e., the vocabulary) in X and $c(x)$ is the count of x in X.

Accordingly, the *description length gain* (DLG) from identifying a (sub)sequence $s = s_1 s_2 \cdots s_k$ in the corpus X as a *segment* or *chunk*, which is expected to have a nice correspondence to a linguistically significant unit (e.g., a lexical item such as a word, or a syntactic phrase), is formulated as

$$DLG(s \in X) = DL(X) - DL(X[r \rightarrow s] \oplus s) \qquad (2)$$

where r is an index, $X[r \rightarrow s]$ represents the resultant corpus by the operation of replacing all occurrences of s with r through out X (in other words, we extract a rule $r \rightarrow s$ from X) and $\oplus$ represents the concatenation of two strings (e.g., $X[r \rightarrow s]$ and s) with a delimiter inserted in between. It is straightforward that the average DLG for extracting an individual s from X is

$$aDLG(s) = \frac{DLG(s)}{c(s)} \qquad (3)$$

This average DLG is an estimation of the *compression effect* of extracting an individual instance of s from X. As an extracted s is supposed to be appended to the modified corpus by a string concatenation, as shown in (2), the original corpus can be easily recovered by a transformation that reverses the extraction, i.e., replacing all r's in $X[r \rightarrow s]$ with the string s.

It is worth noting that we can achieve the purpose of calculating $DL(X[r \rightarrow s] \oplus s)$ without carrying out the string substitution operations throughout the original corpus. The calculation can be based on the token count change involved in the substitution operations to derive the new corpus $X[r \rightarrow s] \oplus s$, as follows:

$$DL(X[r \rightarrow s] \oplus s) = \sum_{x\in V\cup\{r\}} x'(x)\log\frac{c'(x)}{n'} \qquad (4)$$

where $c'(x)$ is the new count of x in the new corpus and n' is the new corpus length. The new counts and the new length are, straightforwardly,

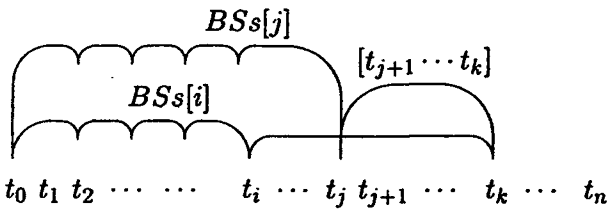
$$\begin{aligned} c'(x) &= \begin{cases} c(s) & \text{if } x = r; \\ c(x) - c(s)c_s(x) + c_s(x) & \text{otherwise.} \end{cases} \\ n' &= n - c(s)|s| + c(s) + |s| + 1 \end{aligned} \qquad (5)$$

where $c(x)$ and $c_s(x)$ are the counts of x in the original corpus X and in the string s, respectively.

A key problem in this straightforward calculation is that we need to derive the count $c(s)$ for all possible string s's in the original corpus X, because during the lexical learning process it is necessary to consider all fragments (i.e., all n-grams) in the corpus in order to select a set of good candidates for lexical items. Kit and Wilks provide an efficient method for deriving n-grams of any length and their counts from large-scale corpora [Kit and Wilks 1998]. It has been adopted as the operational basis for the implementation of the unsupervised lexical acquisition algorithm that is to be reported in the next sections.

3. Learning Algorithm

Given an utterance $U = t_0 t_1 \cdots t_n$ as a string of some linguistic tokens (e.g., characters, words, POS tags), the unsupervised lexical acquisition algorithm seeks for an optimal segmentation $OS(U)$ over the string U such that the sum of the compression effect over the segments is maximal. Formally


$BSs[j]$
$[t_{j+1} \cdots t_k]$
$BSs[i]$
$t_0\ t_1\ t_2 \cdots \cdots\ t_i \cdots t_j\ t_{j+1} \cdots\ t_k \cdots\ t_n$

(A) An illustration for the Viterbi segmentation

```
OpSeg(U = t1 t2 ··· tn)
  For k = 0,1,2,···,n do
    Initialise OS[k] = φ;
    For j = k − 1,···,0 do
      If c([tj+1 ··· tk]) < 2, break;
      If DLG(OS[j] ⊎ {[tj+1 ··· tk]}) > DLG(OS[k])
        then OS[k] = OS[j] ⊎ {[tj+1 ··· tk]}
  The final result: OS[n].
```

(B) The Viterbi segmentation algorithm

Figure 1: The Viterbi algorithm for optimal segmentation, with an illustration

put, it looks for

$$OS(U) = \underset{s_1 \cdots s_k \ s.t.\ U = s_1 + \cdots + s_k}{\arg\max} \sum_{i=1}^{k} aDLG(s_i) \quad (6)$$

where $0 < k \leq n$, $+$ represents a string concatenation and $aDLG(s_i)$ is the average DLG for each instance of the string s_i in the original corpus, as defined in (3) above.

Based on this description length gain calculation, a Viterbi algorithm is formulated to search for the optimal segmentation over an utterance U that fulfils (6). It is presented in Figure 1 with an illustration. The algorithm uses a list of intermediate variables $OS[0], OS[1], \cdots, OS[n]$, each $OSs[i]$ stores the optimal segmentation over $t_0 t_1 \cdots t_i$ (for $i = 0, 1, 2, \cdots, n$). A segmentation is an ordered set (or list) of adjacent segments. The sign $\uplus$ represents an ordered set union operation. The DLG over a list of segments, e.g., $DLG(OS[j])$, is defined as the sum of all segments' DLGs in the set:

$$DLG(OS[j]) = \sum_{s \in OS[j]} DLG(s) \qquad (7)$$

Notice that the algorithm has a bias against the extraction of a single token as a rule, due to the fact that a single token rule bears a negative DLG. When $j = k - 1$, $OS[j] \uplus [t_{j+1} \cdots t_k]$ becomes $OS[k-1] \uplus \{[t_k]\}$, which is less preferable than $OS[k-1] \uplus \{t_k\}$. The difference between the denotations $[t_k]$ and t_k is that the former indicates that the string t_k is extracted from the corpus as the right-hand side of a rule (a deterministic CFG rule), which results in a negative DLG; whereas the latter treats t_k as an individual token instead of a segment, which has a zero DLG.

It is worth noting that the breaking condition $c([\ t_j\ \cdots\ t_k]) < 2$ in the inner loop in the algorithm is an empirical condition. Its main purpose is to speed up the algorithm by avoiding fruitless iterations on strings of count 1. According to our observation in experiments, learning without this breaking condition leads to exactly the same result on large-scale corpora but the speed is many times slower. Strings with a count $c = 1$ can be skipped in the learning, because they are all long strings with a negative DLG[1]and none of them can become a good segment that contributes a positive compression effect to the entire segmentation of the

[1] Since extracting a string $[t_i \cdots t_k]$ of count 1 as a rule does not change any token's count in the new corpus $C[r \to t_i \cdots t_k] \oplus t_i \cdots t_k)$, except the new non-terminal r and the delimiter $\oplus$, whose counts become 1 (i.e., $c(r) = c([t_i \cdots t_k]) = 1$ and $c(\oplus) = 1$) after the extraction. Thus,

$$\begin{aligned}
DLG([t_i \cdots t_k]) &= DL(C) - DL(C[r \to t_i \cdots t_k] \oplus t_i \cdots t_k) \\
&= -\sum_{t \in V} c(t) \log_2 \frac{c(t)}{|C|} - (-\sum_{t \in V \cup \{r, \oplus\}} c(t) \log_2 \frac{c(t)}{|C|+2}) \\
&= -\sum_{t \in V} c(t)(\log_2 \frac{c(t)}{|C|} - \log_2 \frac{c(t)}{|C|+2}) + \sum_{t \in \{r, \oplus\}} c(t) \log_2 \frac{c(t}{|C|+} \\
&= -\sum_{t \in V} c(t) \log_2 \frac{|C|+2}{|C|} + 2 \log_2 \frac{1}{|C|+2} \\
&= -|C| \log_2 \frac{|C|+2}{|C|} - 2 \log_2(|C|+2) \\
&= -(|C|+2) \log_2(|C|+2)) + |C| \log_2 |C| \\
&< 0
\end{aligned}$$

utterance. Rather, they can be broken into shorter segments with a positive DLG.

Time complexity analysis also shows that this breaking condition can speed up the algorithm significantly. Without this condition, the time complexity of the algorithm is $O(n^2)$. With it, the complexity is bounded by $O(mn)$, where m is the maximal common prefix length of sub-strings (i.e., n-grams) in the corpus. Accordingly, the average time complexity of the algorithm is $O(an)$, where a is the average common prefix length in the corpus, which is much smaller than m.

4. Experiments

We have conducted a series of lexical acquisition experiments with the above algorithm on large-scale English corpora, e.g., the Brown corpus [Francis and Kucera 1982] and the PTB WSJ corpus [Marcus *et al.* 1993]. Below is the segmentation result on the first few sentences in the Brown corpus:

```
[the][_fulton_county][_grand_jury][_said_][friday_]
[an][_investigation_of][_atlanta][_'s_][recent]
[_primary_][election][_produced][_``_no][_evidence]
[_''_][that_any][_irregularities][_took_place_][@]
[_the_jury][_further][_said_][in_term][-e][nd_]
[present][ments][_that_][the_city_][executive]
[_committee][_,_which_had][_over-all_][charge_of]
[_the_election][_,_``_][deserves][_the_][praise]
[_and_][thanks][_of_the_c][ity_of_][atlanta][_''_]
[for][_the_manner_in_which][_the_election][_was]
[_conducted_][@][_the][_september][-][october_]
[term][_jury][_had_been_][charge][d_by_][fulton_]
[superior_court][_judge][_dur][wood_][py][e_to]
[_investigat][e][_reports_of_][possible][_``_]
[irregularities][_''_][in_the][_hard-][fought_]
[primary][_which_was_][w][on_by_][mayor][-][nominat]
[e_][iv][an_allen_][jr][_..][_``_][only_a][_relative]
[_handful_of][_such_][reports][_was][_received]
[_''_,_][the_jury][_said_,_``_][considering_the]
[_widespread][_interest_in_][the_election][_,_]
[the_number_of_][vo][ters_and_][the_size_of]
[_this_c][ity_''_][@][_the_jury_said][_it_did]
[_find][_that_many_of][_georgia_'s][_registration]
[_and_][election][_laws][_``_][are_][out][mode][d_]
[or][_inadequate][_and_often_][ambiguous][_''_][@]
[_it][_recommended][_that_][fulton][_legislators_]
[act][_``_][to_have_the][s][e_laws][_studied_]
[and_][revi][sed_to][_the_end_of_][moderniz][ing]
[_and_improv][ing_them][_''_][@][_the][_grand_jury]
[_commented_][on][_a_number_of_][other_][top][ics_,_]
[among_them][_the_atlant][a][_and_][fulton_county]
[_purchasing][_department][s_which_][it][_said][_``_]
[are_well][_operated_][and_follow][_generally_]
[accepted_][practices][_which_in][ure_to][_the_best]
[_interest][_of_both][_government][s_''_][@]
```

where uppercase letters are converted to lowercase ones, the spaces are visualised by an underscore and the full-stops are all replaced by @'s.

Although a space is not distinguished from any other characters for the learner, we have to rely on the spaces to judge the correctness of a word boundary prediction: a predicted word boundary immediately before or after a space is judged as correct. But we also have observed that this criterion overlooks many meaningful predictions like "···charge][d_by···", "···are_outmode][d_···" and "···government][s···". If this is taken into account, the learning performance is evidently better than the precision and recall figures reported in Table 1 below.

Interestingly, it is observed that n-gram counts derived from a larger volume of data can significantly improve the precision but decrease the recall of the word boundary prediction. The correlation between the volume of data used for deriving n-gram counts and the change of precision and recall is shown in Table 1. The effectiveness of the unsupervised learning is evidenced by the fact that its precision and recall are, respectively, all three times as high as the precision and recall by random guessing. The best learning performance, in terms of both precision and recall, in the experiments is the one with 79.33% precision and 63.01% recall, obtained from the experiment on the entire Brown corpus.

Corpus size		0.54	0.88	1.30	3.65	6.13
Learn -ing	P(%)	67.75	70.57	71.97	77.10	79.33
	R(%)	70.39	69.16	67.95	64.98	. 63.01
Guess -ing	P(%)	23.42	24.07	24.54	25.33	26.40
	R(%)	24.34	23.58	23.17	21.35	20.97

Table 1: The correlation between corpus size (million char.) and precision/recall

It is straightforwardly understandable that the increase of data volume leads to a significant increase of precision in the learning, because prediction based on more data is more reliable. The reason for the drop of recall is that when the volume of data increases, more multi-word strings have a higher compression effect (than individual words) and, consequently, they are learned by the learner as lexical items, e.g., [fulton_county], [grand_jury] and [_took_place]. If the credit in such multi-word lexical items is counted, the recall must be much better than the one in Table 1. Of course, this also reflects a limitation of the learning algorithm: it only conducts an optimal segmentation instead of a hierarchical chunking on an utterance.

The precision and recall reported above is not a big surprise. To our knowledge, however, it is the first time that the performance of unsupervised learning of word boundaries is examined with the criteria of *both* precision and recall. Unfortunately, this performance can't be compared with any previous studies, for several reasons. One is that the learning results of previous studies are not presented in a comparable manner, for example, [Wolff 1975, Wolff 1977] and [Nevill-Manning 1996], as noted by [de Marken 1996] as well. Another is that the learning outcomes are different. For example, the output of lexical learning from an utterance (as a character sequence) in [Nevill-Manning 1996] and [de Marken 1995, de Marken 1996] is a hierarchical chunking of the utterance. The chance to hit the correct words in such chunking is obviously many times higher than that in a flat segmentation. The hierarchical chunking leads to a recall above 90% in de Marken's work. Interestingly, however, de Marken does not report the precision, which seems too low, therefore meaningless, to report, because the learner produces so many chunks.

5. Conclusions and Future Work

We have presented an unsupervised learning algorithm for lexical acquisition based on the goodness measure description length gain formulated following information theory. The learning algorithm follows the essence of the MDL principle to search for the optimal segmentation of an utterance that has the maximal description length gain (and therefore approaches the minimum description length of the utterance). Experiments on word boundary prediction with large-scale corpora have shown the effectiveness of the learning algorithm.

For the time being, however, we are unable to compare the learning performance with other researchers' previous work, simply because they do not present the performance of their learning algorithms in terms of the criteria of both precision and recall. Also, our algorithm is significantly simpler, in that it rests on n-gram counts only, instead of any more complicated statistical data or a more sophisticated training algorithm.

Our future work will focus on the investigation into two aspects of the lexical learning with the DLG measure. First, we will incorporate the expectation-maximization (EM) algorithm [Dempster *et al.* 1977] into our lexical learning to see how much performance can be improved. Usually, a more sophisticated learning algorithm leads to a better learning result. Second, we will explore the hierarchical chunking with the DLG measure. We are particularly interested to know how much more compression effect can be further squeezed out by hierarchical chunking from a text corpus (e.g., the Brown corpus) and how much improvement in the recall can be achieved.

Acknowledgements

The first author gratefully acknowledges a University of Sheffield Research Scholarship awarded to him that enables him to undertake this work. We wish to thank two anonymous reviewers for their invaluable comments, and thank Hamish Cunningham, Ted Dunning, Rob Gaizauskas, Randy LaPolla, Steve Renals, Jonathon Webster and many other colleagues for various kinds of help and useful discussions.

References

[Cover and Thomas 1991] Cover, T. M., and J. A. Thomas, 1991. *Elements of Information Theory.* John Wiley and Sons, Inc., New York.

[de Marken 1995] de Marken, C. 1995. The Unsupervised Acquisition of a Lexicon from Continuous Speech. Technical Report A.I. Memo No. 1558, AI Lab., MIT. Cambridge, Massachusetts.

[de Marken 1996] de Marken, C. 1996. *Unsupervised Language Acquisition.* Ph.D. dissertation, MIT, Cambridge, Massachusetts.

[Dempster *et al.* 1977] Dempster, A. P., N. M. Laird, and D. B. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, **39**(B):1-38.

[Francis and Kucera 1982] Francis, W. N., and H. Kucera. 1982. *Frequency Analysis of English Usage: Lexical and Grammar.* Houghton-Mifflin, Boston.

[Kit 1998] Kit, C. 1998. A goodness measure for phrase learning via compression with the MDL principle. In *The ESSLLI-98 Student Session*, Chapter 13, pp.175-187. Aug. 17-28, Saarbrüken.

[Kit and Wilks 1998] Kit, C., and Y. Wilks. 1998. The Virtual Corpus approach to deriving n-gram statistics from large scale corpora. In C. N. Huang (ed.), *Proceedings of 1998 International Conference on Chinese Information Processing*, pp.223-229. Nov. 18-20, Beijing.

[Li and Vitányi 1993] Li, M., and P. M. B. Vitányi. 1993. *Introduction to Kolmogorov Complexity and its Applications.* Springer-Verlag, New York. Second edition, 1997.

[Marcus *et al.* 1993] Marcus, M., B. Santorini and M. Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, **19**(2):313-330.

[Nevill-Manning 1996] Nevill-Manning, C. G. *Inferring Sequential Structure.* Ph.D. dissertation, University of Waikato, New Zealand.

[Powers 1997] Powers, D. M. W. Unsupervised learning of linguistic structure: an empirical evaluation. *International Journal of Corpus Linguistics*, **2**(1):91-131.

[Rissanen 1978] Rissanen, J. 1978. Modelling by shortest data description. *Automatica*, **14**:465-471.

[Rissanen 1982] Rissanen, J. 1982. A universal prior for integers and estimation by minimum description length. *Ann. Statist.*. **11**:416-431.

[Rissanen 1989] Rissanen, J. 1989. *Stochastic Com-plexity in Statistical Inquiry.* World Scientific, N.J.

[Shannon 1948] Shannon, C. 1948. A mathematical theory of communication. *Bell System Technical Journal*, **27**:379-423, 623-656.

[Solomonoff 1964] Solomonoff, R. J. 1964. A formal theory of inductive inference, part 1 and 2. *Information Control*, **7**:1-22, 224-256.

[Stolcke 1994] Stolcke, A. 1994. *Bayesian Learning of Probabilistic Language Models.* Ph.D. dissertation, UC Berkeley, CA.

[Vitányi and Li 1996] Vitányi, P. M. B., and M. Li. 1996. Minimum Description Length Induction, Bayesianism, and Kolmogorov Complexity. Manuscript, CWI, Amsterdam.

[Wolff 1975] Wolff, J. G. An algorithm for the segmentation of an artificial language analogue. *British Journal of Psychology*, **66**:79-90.

[Wolff 1977] Wolff, J. G. The discovery of segments in natural language. *British Journal of Psychology*, **68**:97-106.

[Wolff 1982] Wolff, J. G. Language acquisition, data compression and generalization. *Language and Communication*, **2**:57-89.