Finding Structure via Compression

Jason L. Hutchens
Dept. of E&E Engineering
University of Western Australia
Nedlands W.A. 6907, Australia
hutch@ciips.ee.uwa.edu.au

Michael D. Alder
Dept. of Mathematics
University of Western Australia
Nedlands W.A. 6907, Australia
mike@maths.uwa.edu.au

Abstract

A statistical language model may be used to segment a data sequence by thresholding its instantaneous entropy. In this paper we describe how this process works, and we apply it to the problem of discovering separator symbols in a text. Our results show that language models which bootstrap themselves with structure found in this way undergo a reduction in perplexity. We conclude that these techniques may be useful in the design of generic grammatical inference systems.

1 Introduction

The modelling of a symbol sequence requires some assumptions about the nature of the process which generated it, and the modelling of English text would, for example, commonly make the assumption that the text consists of *words*, short strings which usually recur and which are separated by whitespace, and punctuation symbols. The whitespace symbol (which we shall represent explicitly by $\wedge$) and its distinctive function do not seem to occur in spoken English, and would not seem to be essential in written English. We are concerned with finding such structure using weak assumptions, rather than being given it as part of a model.

In this paper we show that a statistical model may be used to do just that, and our results indicate that a model which bootstraps itself using this structure undergoes a reduction in perplexity.

2 Entropic Chunking

A predictive model M is a model which, when presented with a sequence of symbols s, is able to make a prediction about the next symbol in the sequence in the form of a probability distribution over the alphabet Σ (for the purposes of this investigation, Σ is the set of ASCII characters). We assume that the estimated probability distribution is smoothed to avoid the zero-frequency problem. The specifics of the model are unimportant; the methods presented in this paper are intended to be generic, but it is clear that n-th order Markov models, for n less than the length of s, would qualify.

The *information* of a symbol w with respect to a statistical model M and a context s is defined in Equation 1. Intuitively we may think of the information as the surprise the model experiences upon receipt of the symbol w; it is low if the model's expectations are vindicated, high if they are erroneous (Shannon and Weaver, 1949).

$$I(w|M,s) = -\log_2 P(w|M,s) \tag{1}$$

The *entropy* of a language model, defined in Equation 2, is the expected value of the information. The entropy is a measure of the model's uncertainty about the future; it will be low if the model expects one particular symbol to occur with a high probability, and it increases as the estimated probability distribution approaches the uniform.

$$H(M,s) = \sum_{w \in \Sigma} P(w|M,s) I(w|M,s) \tag{2}$$

If one monitors the instantaneous entropy of a language model as it scans across an English text, one generally finds that regions of high entropy correspond with word boundaries (Alder, 1988). This is convincingly demonstrated by Figure 1, which plots the entropy of a second-order Markov model across the first sentence of "A Scandal in Bohemia", by Sir Arthur Conan Doyle. The training corpus used in this example was 3.5 megabytes of Sherlock Holmes stories, minus the testing sentence.[1]

Segmentation is a matter of *chunking* the data whenever the instantaneous entropy exceeds some threshold value (Wolff, 1977). A *chunk* is merely a

[1] Email the authors for further information about this corpus.

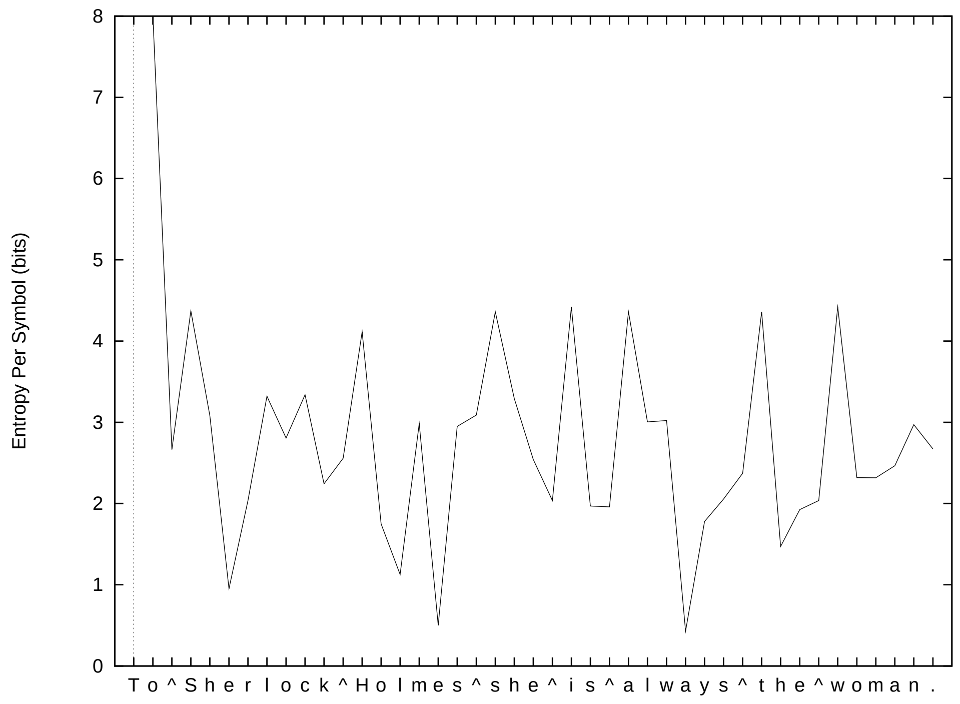

Figure 1: Entropy across a text segment.

string of symbols which constitute a higher-level lexeme. Throughout this paper a chunking threshold of $\frac{1}{2}\log_2 ||\Sigma||$ bits is used, although this is almost certainly not an optimal value. The problem of finding a good threshold automatically warrants investigation.

The Sherlock Holmes corpus was segmented in this way. Table 1 lists, in decreasing order of frequency, the most common chunks found in the text. The fact that they agree rather well with the most frequent words in the English language is encouraging.

Chunk	Number of Occurrences
the∧	21790
a∧	11018
of∧	10263
to∧	9838
and∧	9444
that∧	6035

Table 1: The most common chunks discovered in the Sherlock Holmes corpus.

A total of 70171 distinct chunks were found in the corpus. Of these, a massive 66821 chunks occurred ten times or less—these chunks were discarded due to their infrequency (all anomalous chunks, such as "halfc" and "ichth", occurred in this group). The majority of the remaining 3350 chunks were found to be valid English words. Those that weren't were strings of two or more English words, such as "in∧the∧", "it∧was∧" and "do∧you∧think∧that∧".

The previous experiment was repeated using a version of the Sherlock Holmes corpus which had many clues to word boundaries removed; all characters were replaced with their uppercase equivalents, and whitespace and punctuation symbols were deleted. Many good chunks were discovered, such as "THE", "TO", "WAS", "OFTHE", "HAVEBEEN" and "POLICE". However, anomalous chunks were prevalent, with "REWAS" and "STO" occurring as frequently as the chunks a human being would identify as English words.

Even so, entropic chunking provides a technique for discovering structure which makes very few assumptions about the information that the data contains.

2.1 Finding Separator Symbols

In natural language text, words are typically separated by whitespace. Entropic chunking may be used to discover this automatically, by recording which symbols occur immediately prior to a large jump in entropy.

Table 2 lists, in decreasing order of frequency, separator symbols discovered in the Sherlock Holmes corpus. The ∧ symbol precedes the majority of sudden jumps in entropy, which agrees with our expectations. The - symbol occurs within hyphenated words, which were usually broken up into their constituents, while the " symbol occurs as a chunk separator whenever two pieces of dialogue appear back-to-back. The remaining probability mass was distributed over 43 symbols, which were discarded as anomalies.

Separator Symbol	Frequency
$\wedge$	99.55%
-	0.23%
"	0.11%

Table 2: Separator symbols discovered in the Sherlock Holmes corpus.

Once one or more separator symbols have been found, traditional parsing techniques may be used to segment the text.

Many data sequences simply will not have separator symbols. For example, a database may store fields in a file based on their bit length only. In such situations entropic chunking must be used if no prior assumptions about the structure of the data are to be made.

3 Data Compression

In order to test the value of adding chunks to a language model's alphabet, we conducted a simple experiment. The Sherlock Holmes corpus was divided into three non-overlapping parts, each of roughly a megabyte in size. These three corpora were used for training, chunking and testing respectively.

A standard PPMC model was inferred from the training corpus and used to segment the chunking corpus (Moffat, 1990). The most common chunk was then added to the alphabet of the PPMC model in a process we refer to as the *upwrite* (Hutchens, 1997).

Evaluation was performed by measuring the *perplexity* of the PPMC model with respect to the testing corpus (Jelinek and Lafferty, 1991). The perplexity, defined in Equation 3 for a corpus of N symbols, is a monotone function of the average information of the model, and is therefore a measure of compression.

$$PP = P(w_1, w_2, \ldots, w_N)^{-\frac{1}{N}} \tag{3}$$

It should be mentioned that PPMC is usually used in adaptive data compression systems. In our experiment we used it in a non-adaptive way; the model was inferred from one corpus and tested on another. Although true compression systems avoid this two-pass approach due to the expense of transmitting the model, evaluation is performed this way in the speech recognition literature.

An iteration of this process was used to produce the plot shown in Figure 2. Perplexity is given in units of characters rather than symbols—this is necessary because the alphabet size increases with every chunk added.

A minimum perplexity of 4.48 characters was attained after 154 chunks had been added to the model's alphabet. This represents a 9.5% reduction of the model's initial perplexity of 4.95 characters, equivalent to a 6.2% improvement in compression performance. Although this result is by no means ground-breaking, we believe that it illustrates the advantage of chunking.

The initial reduction in perplexity is rapid, as the first chunks discovered correspond to the most frequent English words. The continued addition of chunks reduces the perplexity further, discounting minor local variations. We expect that the performance of the model will degrade once too many chunks are added to its alphabet, but the experiment didn't proceed long enough to make this apparent.

4 Conclusion and Future Work

We have shown that a statistical language model may discover high-level structure in a data sequence by thresholding its instantaneous entropy. When this structure is used to augment the model, its compression performance improves. Although the example presented in this paper used a natural language corpus, we stress that these techniques are suited to the analysis of all kinds of data.

We plan to investigate how much structure can be learned by the most trivial of language models. The upwrite process provides scaffolding which allows high-level structure to be found: we believe that a low-order language model which uses the binary alphabet may be able to find characters, then words, and eventually larger scale structures in natural language corpora.

Methods to select appropriate entropic thresholds need to be investigated, and the application of entropic chunking to adaptive data compression systems is being explored and looks promising.

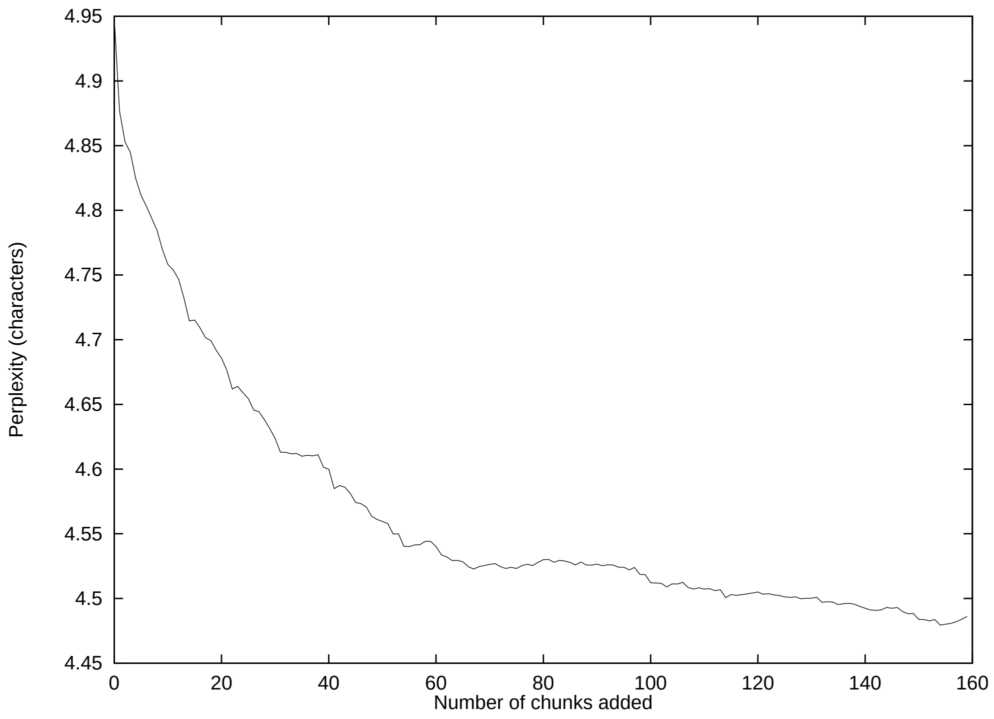

Figure 2: Effect of chunking on model perplexity.

References

Alder, Mike. 1988. Stochastic grammatical inference. Master's thesis, University of Western Australia.

Hutchens, Jason L. 1997. Language acquisition and data compression. In Sarah Boyd, editor, *1997 Australasian Natural Language Processing Summer Workshop*, pages 39–49, February.

Jelinek, Frederick and John D. Lafferty. 1991. Computation of the probability of initial substring generation by stochastic context free grammars. *Computational Linguistics*, 17(3):315–323.

Moffat, Alistair. 1990. Implementing the PPM data compression scheme. *IEEE Transactions on Communications*, 38(11):1917–1921, November.

Shannon, Claude E. and Warren Weaver. 1949. *The Mathematical Theory of Communication.* University of Illinois Press.

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