Exploring variant definitions of pointer length in MDL

Aris Xanthos
Department of Linguistics
University of Chicago
Chicago IL 60637
axanthos@uchicago.edu

Yu Hu
Department of
Computer Science
University of Chicago
Chicago IL 60637
yuhu@uchicago.edu

John Goldsmith
Departments of Linguistics and
Computer Science
University of Chicago
Chicago IL 60637
goldsmith@uchicago.edu

Abstract

Within the information-theoretical framework described by (Rissanen, 1989; de Marcken, 1996; Goldsmith, 2001), pointers are used to avoid repetition of phonological material. Work with which we are familiar has assumed that there is only one way in which items could be pointed to. The purpose of this paper is to describe and compare several different methods, each of which satisfies MDL's basic requirements, but which have different consequences for the treatment of linguistic phenomena. In particular, we assess the conditions under which these different ways of pointing yield more compact descriptions of the data, both from a theoretical and an empirical perspective.

1 Introduction

The fundamental hypothesis underlying the *Minimum Description Length* (*MDL*) framework (Rissanen, 1989; de Marcken, 1996; Goldsmith, 2001) is that the selection of a model for explaining a set of data should aim at satisfying two constraints: on the one hand, it is desirable to select a model that can be described in a highly compact fashion; on the other hand, the selected model should make it possible to model the data well, which is interpreted as being able to describe the data in a maximally compact fashion. In order to turn this principle into an operational procedure, it is necessary to make explicit the notion of *compactness*. This is not a trivial problem, as the compactness (or conversely, the *length*) of a description depends not only on the complexity of the object being described (in this case, either a model or a set of data given a model), but also on the "language" that is used for the description.

Consider, for instance, the model of morphology described in Goldsmith (2001). In this work, the data consist in a (symbolically transcribed) corpus segmented into words, and the "language" used to describe the data contains essentially three objects: a list of *stems*, a list of *suffixes*, and a list of *signatures*, i.e. structures specifying which stems associate with which suffixes to form the words found in the corpus. The length of a particular model (or *morphology*) is defined as the sum of the lengths of the three lists that compose it; the length of each list is in turn defined as the sum of the lengths of elements in it, plus a small cost for the list structure itself[1]. The length of an individual morpheme (stem or suffix) is taken to be proportional to the number of symbols in it.

Calculating the length of a signature involves the notion of *pointer*, with which this paper is primarily concerned. The function of a signature is to relate a number of stems with a number of suffixes. Since each of these morphemes is spelled once in the corresponding list, there is no need to spell it again in a signature that contains it. Rather, each signature comprises a list of pointers to stems and a list of pointers to suffixes. A pointer is a symbol that *stands for* a particular morpheme, and the recourse to pointers relies on the assumption that

[1] More on this in section 2.1 below

Proceedings of the Eighth Meeting of the ACL Special Interest Group on Computational Phonology at HLT-NAACL 2006, pages 32–40,
New York City, USA, June 2006. ©2006 Association for Computational Linguistics

their length is lesser than that of the morphemes they replace. Following information-theoretic principles (Shannon, 1948), the length of a pointer to a morpheme (under some optimal encoding scheme) is equal to -1 times the binary logarithm of that morpheme's probability. The length of a signature is the sum of the lengths of the two lists it contains, and the length of each list is the sum of the lengths of the pointers it contains (plus a small cost for the list itself).

This work and related approaches to unsupervised language learning have assumed that there is only one way in which items could be pointed to, or identified. The purpose of this paper is to describe, compare and evaluate several different methods, each of which satisfies MDL's basic requirements, but which have different consequences for the treatment of linguistic phenomena. One the one hand, we contrast the expected description length of "standard" lists of pointers with *polarized* lists of pointers, which are specified as either (i) pointing to the relevant morphemes (those that belong to a signature, or undergo a morpho-phonological rule, for instance) or (ii) pointing to their complement (those that do not belong to a signature, or do not undergo a rule). On the other hand, we compare (polarized) lists of pointers with a method based on binary strings specifying each morpheme as relevant or not (for a given signature, rule, etc.). In particular, we discuss the conditions under which these different ways of pointing are expected to yield more compact descriptions of the data.

The remainder of this paper is organized as follows. In the next section, we give a formal review of the standard treatment of lists of pointers as described in (Goldsmith, 2001); then we successively introduce polarized lists of pointers and the method of binary strings, and make a first, theoretical comparison of them. Section three is devoted to an empirical comparison of these methods on a large natural language corpus. In conclusion, we discuss the implications of our results in the broader context of unsupervised language learning.

2 Variant definitions of pointers

In order to simplify the following theoretical discussion, we temporarily abstract away from the complexity of a full-blown model of morphology. Given a set of N stems and their distribution, we consider the general problem of pointing to a subset of M stems (with $0 < M \leq N$), first by means of "standard" lists of pointers, then by means of polarized ones, and finally by means of binary strings.

2.1 Expected length of lists of pointers

Let τ denote a set of N stems; we assume that the length of a pointer to a specific stem $t \in \tau$ is its inverse log probability $-\log pr(t)$.[2] Now, let $\{M\}$ denote the set of all subsets of τ that contain exactly $0 < M \leq N$ stems. The *description length* of a list of pointers to a particular subset $\mu \in \{M\}$ is defined as the sum of the lengths of the M pointers it contains, plus a small cost of for specifying the list structure itself, defined as $\lambda(M) := 0$ if $M = 0$ and $\log M$ bits otherwise[3]:

$$DL^{\text{ptr}}(\mu) := \lambda(M) - \sum_{t\in\mu} \log pr(t)$$

The *expected* length of a pointer is equal to the entropy over the distribution of stems:

$$h^{\text{stems}} := E_{t\in\tau}\left[-\log pr(t)\right] = -\sum_{t\in\tau} pr(t)\log pr(t)$$

Thus, the expected description length of a list of pointers to M stems (over all subsets $\mu \in \{M\}$) is:

$$\begin{aligned} E_{\mu\in\{M\}}\left[DL^{\text{ptr}}(\mu)\right] &= \tfrac{1}{|\{M\}|}\sum_{\mu\in\{M\}} DL^{\text{ptr}}(\mu) \\ &= \lambda(M) + Mh^{\text{stems}} \end{aligned} \tag{1}$$

This value increases as a function of both the number of stems which are pointed to and the entropy over the distribution of stems. Since $0 \leq h^{\text{stems}} \leq \log N$, the following bounds hold:

$$\begin{aligned} 0 \leq h^{\text{stems}} \leq E_{\mu\in\{M\}}\left[DL^{\text{ptr}}(\mu)\right] \\ \leq \log N + Nh^{\text{stems}} \leq (N+1)\log N \end{aligned}$$

[2] Here and throughout the paper, we use the notation $\log x$ to refer to the *binary* logarithm of x; thus entropy and other information-theoretic quantities are expressed in terms of *bits*.

[3] Cases where the argument of this function can have the value 0 will arise in the next section.

2.2 Polarization

Consider a set of $N = 3$ equiprobable stems, and suppose that we need to specify that a given morphophonological rule applies to one of them. In this context, a list with a single pointer to a stem requires $\log 1 - \log \frac{1}{3} = 1.58$ bits. Suppose now that the rule is more general and applies to two of the three stems. The length of the new list of pointers is thus $\log 2 - 2\log \frac{1}{3} = 4.17$ bits. It appears that for such a general rule, it is more compact to list the stems to which it does *not* apply, and mark the list with a flag that indicates the "negative" meaning of the pointers. Since the flag signals a binary choice (either the list points to stems that undergo the rule, or to those that do not), $\log 2 = 1$ bit suffices to encode it, so that the length of the new list is $1.58 + 1 = 2.58$ bits.

We propose to use the term *polarized* to refer to lists of pointers bearing a such flag. If it is useful to distinguish between specific settings of the flag, we may speak of *positive* versus *negative* lists of pointers (the latter being the case of our last example). The expected description length of a polarized list of M pointers is:

$$\begin{gathered} E_{\mu\in\{M\}}\left[DL^{\text{pol}}(\mu)\right] = 1 + \lambda(\hat{M}) + \hat{M}h^{\text{stems}} \\ \text{with } \hat{M} := \min(M, N-M) \end{gathered} \tag{2}$$

From (1) and (2), we find that in general, the expected *gain* in description length by polarizing a list of M pointers is:

$$\begin{aligned} & E_{\mu\in\{M\}}\left[DL^{\text{ptr}}(\mu) - DL^{\text{pol}}(\mu)\right] \\ &= \begin{cases} -1 \text{ iff } M \leq \frac{N}{2} \\ -1 + \lambda(M) - \lambda(N-M) + (2M-N)h^{\text{stems}} \\ \quad \text{otherwise} \end{cases} \end{aligned}$$

Thus, if the number of stems pointed to is lesser than or equal to half the total number of stems, using a polarized list rather than a non-polarized one means wasting exactly 1 bit for encoding the superfluous flag. If the number of stems pointed to is larger than that, we still pay 1 bit for the flag, but the reduced number of pointers results in an expected saving of $\lambda(M) - \lambda(N-M)$ bits for the list structure, plus $(2M-N)\cdot h^{\text{stems}}$ bits for the pointers themselves.

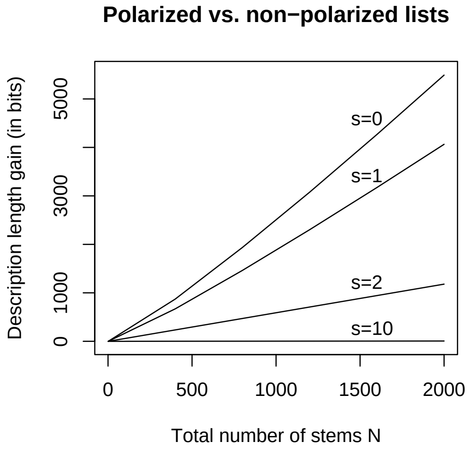

Figure 1: Expected gain in description length by using polarized rather than non-polarized lists of pointers.

Now, let us assume that we have no information regarding the number M of elements which are pointed to, i.e. that it has a uniform distribution between 1 and N ($M \sim U[1, N]$). Let us further assume that stems follow a Zipfian distribution of parameter s, so that the probability of the k-th most frequent stem is defined as:

$$f(k, N, s) := \frac{1/k^s}{H_{N,s}} \quad \text{with} \quad H_{N,s} := \sum_{n=1}^{N} 1/n^s$$

where $H_{N,s}$ stands for the *harmonic number* of order N of s. The entropy over this distribution is:

$$h^{\text{Zipf}}_{N,s} := \frac{s}{H_{N,s}} \sum_{k=1}^{N} \frac{\log k}{k^s} + \log H_{N,s}$$

Armed with these assumptions, we may now *compute* the expected description length gain of polarization (over all values of M) as a function of N and s:

$$\begin{gathered} E_M\big(E_{\mu\in\{M\}}\left[DL^{\text{ptr}}(\mu) - DL^{\text{pol}}(\mu)\right]\big) \\ = -1 + \tfrac{1}{N}\sum_{M=1}^{N} \lambda(M) - \lambda(\hat{M}) + (M - \hat{M})h^{\text{Zipf}}_{N,s} \end{gathered}$$

Figure 1 shows the gain calculated for N = 1, 400, 800, 1200, 1600 and 2000, and s = 0, 1, 2 and 10. In general, it increases with N, with a slope that depends on s: the greater the value of s, the lesser the entropy over the distribution of stems; since the entropy corresponds to the expected length

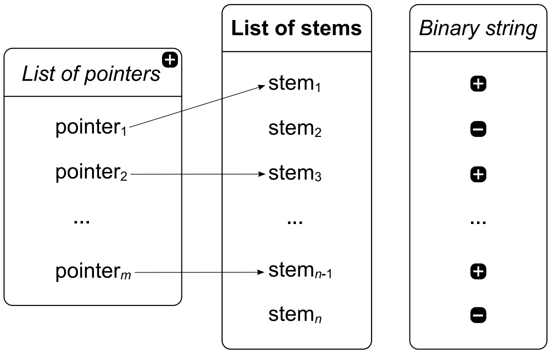

Figure 2: Two ways of pointings to stems: by means of a polarized list of pointers, or a binary string.

of a pointer, its decrease entails a decrease in the number of bits that can be saved by using polarized lists (which generally use less pointers). However, even for an aberrantly skewed distribution of stems[4], the expected gain of polarization remains positive. Since the value of s is usually taken to be slightly greater than 1 for natural languages (Mandelbrot, 1953), it seems that polarized lists generally entail a considerable gain in description length.

2.3 Binary strings

Consider again the problem of pointing to one out of three equiprobable stems. Suppose that the list of stems is ordered, and that we want to point to the first one, for instance. An alternative to the recourse to a list of pointers consists in using a binary string (in this case `100`) where the i-th symbol is set to `1` (or +) if the i-th stem is being pointed to, and to `0` (or −) otherwise. Figure 2 gives a schematic view of these two ways of pointing to items.

There are two main differences between this method and the previous one. On the one hand, the number of symbols in the string is constant and equal to the *total* number N of stems, regardless of the number M of stems that are pointed to. On the other hand, the compressed length of the string depends on the distribution of symbols in it, and *not* on the distribution of stems. Thus, by comparison with the description length of a list of pointers, there is a loss due to the larger number of encoded symbols, and a gain due to the use of an encoding specifically tailored for the relevant distribution of pointed versus "unpointed" elements.

The entropy associated with a binary string is entirely determined by the number of `1`'s it contains, i.e. the number M of stems which are pointed to, and the length N of the string:

$$h^{\text{bin}}_{N,M} := -\frac{M}{N}\log\frac{M}{N} - \frac{N-M}{N}\log\frac{N-M}{N}$$

The compressed length of a binary string pointing to M stems is thus:

$$DL^{\text{bin}}(M) := Nh^{\text{bin}}_{N,M} \qquad (3)$$

It is maximal and equal to N bits when $M = \frac{N}{2}$, and minimal and equal to 0 when $M = N$, i.e. when *all* stems have a pointer on them. Notice that binary strings are intrinsically polarized, so that interverting `0`'s and `1`'s results in the same description length regardless of their distribution.[5]

The question naturally arises, under which conditions would binary strings be more or less compact than polarized lists of pointers. If we assume again that the distribution of the number of elements pointed to is uniform and the distribution of stems is Zipfian of parameter s, (2) and (3) justify the following expression for the expected description length gain by using binary strings rather than polarized lists (as a function of N and s):

$$E_M\big[E_{\mu\in\{M\}}[DL^{\text{pol}}(\mu)] - DL^{\text{bin}}(M)\big]$$
$$= 1 + \tfrac{1}{N}\textstyle\sum_{M=1}^{N}\lambda(\hat{M}) + \hat{M}h^{\text{Zipf}}_{N,s} - Nh^{\text{bin}}_{N,M}$$

Figure 3 shows the gain calculated for N = 1, 400, 800, 1200, 1600 and 2000, and s = 0, 1, 2 and 3. For s small, i.e. when the entropy over the distribution of stems is greater or not much lesser than that of natural languages, the description length of binary strings is considerably lesser than that of polarized lists. The difference decreases as s increases,

[4] In the case $s = 10$, the probability of the most frequent stem is .999 for $N = 2000$.

[5] As one the reviewers has indicated to us, the binary strings approach is actually very similar to the method of *combinatorial codes* described by (Rissanen, 1989). This method consists in pointing to one among $\binom{N}{M}$ possible combinations of M stems out of N. Under the assumption that these combinations have a uniform probability, the cost for pointing to M stems is $\log\binom{N}{M}$ bits, which is in general slightly lesser than the description length of the corresponding binary string (the difference being maximal for $M = N/2$, i.e. when the binary string encoding cannot take advantage of any compression).

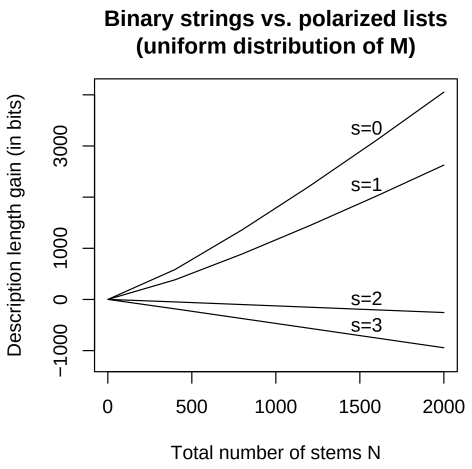

Figure 3: Expected gain in description length by using binary strings rather than polarized lists under the assumption that $M \sim U[1,N]$.

until at some point (around $s = 2$), the situation reverses and polarized lists become more compact. In both cases, the trend increases with the number N of stems (within the range of values observed).

By contrast, it is instructive to consider a case where the distribution of the number of elements pointed to departs from uniformity. For instance, we can make the assumption that M follows a binomial distribution ($M \sim B[N,p]$).[6] Under this assumption (and, as always, that of a Zipfian distribution of stems), the expected description length gain by using binary strings rather than polarized lists is:

$$E_M\big[E_{\mu\in\{M\}}[DL^{\text{ptr}}(\mu)] - DL^{\text{bin}}(M)\big]$$
$$= \sum_{M=1}^{N} pr(M)\Big(1+\lambda(\hat{M})+\hat{M}h^{\text{Zipf}}_{N,s} - Nh^{\text{bin}}_{N,M}\Big)$$
$$\text{with } pr(M) = \binom{N}{M}p^M(1-p)^{N-M}$$

Letting N and s vary as in the previous computation, we set the probability for a stem to have a pointer on it to $p = 0.01$, so that the distribution of pointed versus "unpointed" elements is considerably skewed.[7]

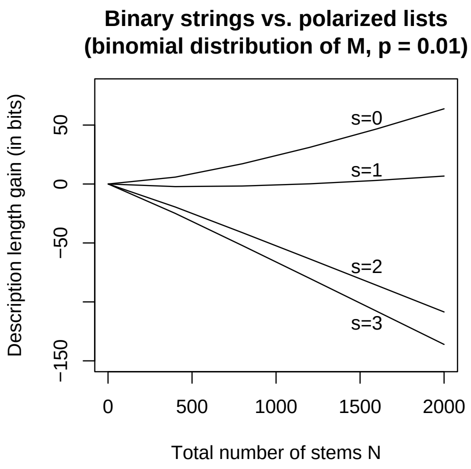

Figure 4: Expected gain in description length by using binary strings rather than polarized lists under the assumption that $M \sim B[N, 0.01]$.

As shown on figure 4, under these conditions, the absolute value of the gain of using binary strings gets much smaller in general, and the value of s for which the gain becomes negative for N large gets close to 1 (for this particular value, it becomes positive at some point between $N = 1200$ and $N = 1600$).

Altogether, under the assumptions that we have used, these theoretical considerations suggest that binary strings generally yield shorter description lengths than polarized lists of pointers. Of course, data for which these assumptions do not hold could arise. In the perspective of unsupervised learning, it would be particularily interesting to observe that such data drive the learner to induce a different model depending on the representation of pointers being adopted.

It should be noted that nothing prevents binary strings and lists of pointers from coexisting in a single system, which would select the most compact one for each particular case. On the other hand, it is a logical necessity that all lists of pointers be of the same kind, either polarized or not.

3 Experiments

In the previous section, by assuming frequencies of stems and possible distributions of M (the number of stems per signature), we have explored theoretically the differences between several encoding

[6] This model predicts that most of the time, the number M of elements pointed to is equal to $N \cdot p$ (where p denotes the probability for a stem to have a pointer on it), and that the probability $pr(M)$ of other values of M decreases as they diverge from $N \cdot p$.

[7] By symmetry, the same results would be found with $p = 0.99$.

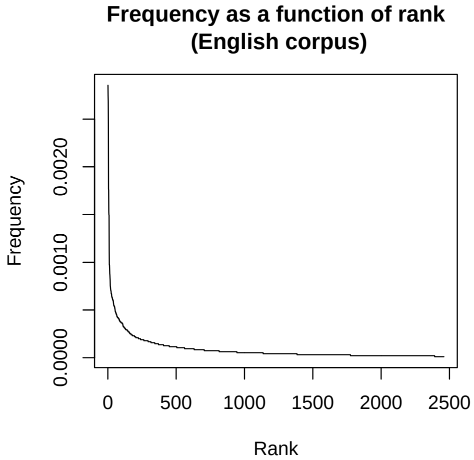

Figure 5: Frequency versus rank (stems) in English corpus.

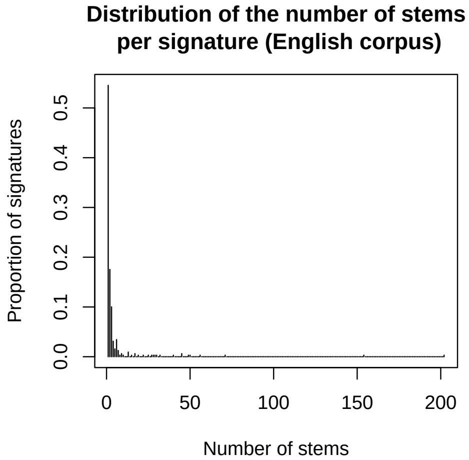

Figure 6: Distribution of number of stems per signature (English corpus)

methods in the MDL framework. In this section, we apply these methods to the problem of suffix discovery in natural language corpora, in order to verify the theoretical predictions we made previously. Thus, the purpose of these experiments is not to state that one encoding is preferable to the others; rather, we want to answer the three following questions:

1. Are our assumptions on the frequency of stems and size of signatures appropriate for natural language corpora?

2. Given these assumptions, do our theoretical analyses correctly predict the difference in description length of two encodings?

3. What is the relationship between the gain in description length and the size of the corpus?

3.1 Experimental methodology

In this experiment, for the purpose of calculating distinct description lengths while using different encoding methods, we modified *Linguistica*[8] by implementing *list of pointers* and *binary strings* as alternative means to encode the pointers from signatures to their associated stems[9]. As a result, given a set of signatures, we are able to compute a description length for each encoding methods.

Within *Linguistica*, the morphology learning process can be divided into a sequence of heuristics, each of which searches for possible incremental modifications to the current morphology. For example, in the suffix-discovery procedure, ten heuristics are carried out successively; thus, we have a distinct set of signatures after applying each of the ten heuristics. Then, for each of these sets, we encode the *pointers* from each signature to its corresponding stems in three rival ways: as a *list of pointers* (polarized or not), as traditionally understood, and as a *binary string*. This way, we can compute the total description length of the signature-stem-linkage for each of the ten sets of signatures and for each of three two ways of encoding the pointers. We also collect statistics on word frequencies and on the distribution of the size of signatures M, i.e. the number M of stems which are are pointed to, both of which are important parametric components in our theoretical analysis.

Experiments are carried out on two orthographic corpora (English and French), each of which has 100,000 word tokens.

3.2 Frequency of stems and size of signatures

The frequency of stems as a function of their rank and the distribution of the size of signatures are plot-

[8] The source and binary files can be freely downloaded at `http://linguistica.uchicago.edu`.

[9] Pointers to *suffixes* are not considered here.

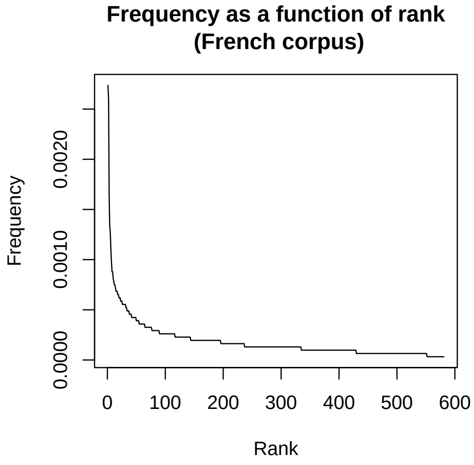

Figure 7: Frequency versus rank (stems) in French corpus.

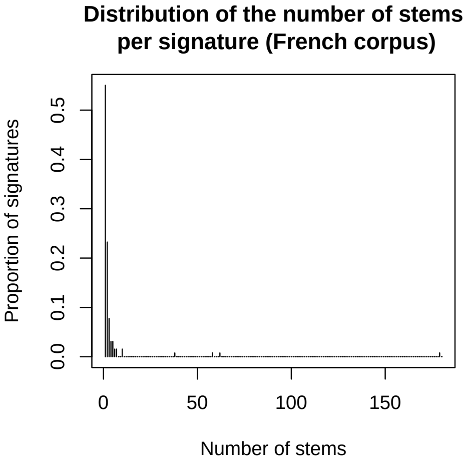

Figure 8: Distribution of number of stems per signature (French corpus)

ted in figures 5 and 6 for the English corpus, and in figures 7 and 8 for the French corpus. These graphs show that in both the English and the French corpora, stems appear to have a distribution similar to a Zipfian one. In addition, in both corpora, M follows a distribution whose character we are not sure of, but which appears more similar to a binomial distribution. To some extent, these observations are consistent with the assumptions we made in the previous theoretical analysis.

3.3 Description length of each encoding

The description length obtained with each encoding method is displayed in figures 9 (English corpus) and 10 (French corpus), in which the x-axis refers to the set of signatures resulting from the application of each successive heuristics, and the y-axis corresponds to the description length in bits. Note that we only plot description lengths of *non-polarized* lists of pointers, because the number of stems per signature is always less than half the total number of stems in these data (and we expect that this would be true for other languages as well).[10]

These two plots show that in both corpora, there is always a gain in description length by using binary strings rather than lists of pointers for encoding the pointers from signatures to stems. This observation is consistent with our conclusion in section 2.3, but it is important to emphasize again that for other data (or other applications), lists of pointers might turn out to be more compact.

3.4 Description length gain as a function of corpus size

In order to evaluate the effect of corpus size on the gain in description length by using binary string rather than lists of variable-length pointers, we applied *Linguistica* to a number of English corpora of different sizes ranging between 5,000 to 200,000 tokens. For the final set of signatures obtained with each corpus, we then compute the gain of binary strings encoding over lists of pointers as we did in the previous experiments. The results are plotted in figure 11.

This graph shows a strong positive correlation between description length gain and corpus size. This is reminiscent of the results of our theoretical simulations displayed in figures 3 and 4. As before, we interpret the match between the experimental results and the theoretical expectations as evidence supporting the validity of our theoretical predictions.

3.5 Discussion of experiments

These experiments are actually a number of case studies, in which we verify the applicability of our theoretical analysis on variant definitions of pointer lengths in the MDL framework. For the particu-

[10] See figures 6 and 8 as well as section 2.2 above.

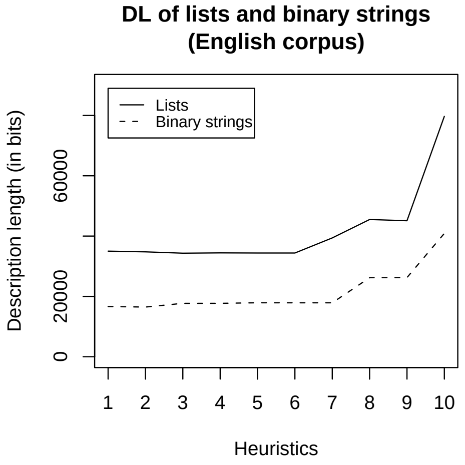

Figure 9: Comparison of DL of 10 successive morphologies using pointers versus binary strings (English corpus).

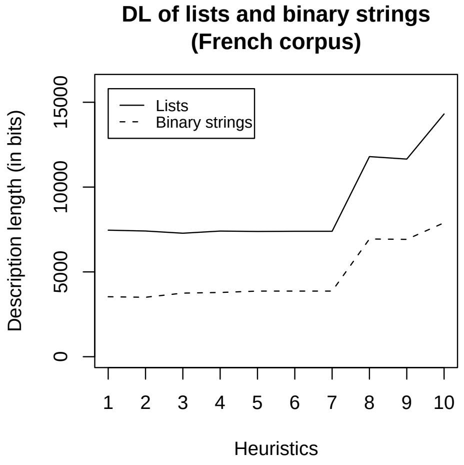

Figure 10: Comparison of DL of 10 successive morphologies using pointers versus binary strings (French corpus)

lar application we considered, learning morphology with *Linguistica*, binary strings encoding proves to be more compact than lists of variable-length pointers. However, the purpose of this paper is not to predict that one variant is always better, but rather to explore the mathematics behind different encodings. Armed with the mathematical analysis of different encodings, we hope to be better capable of making the right choice under specific conditions. In particular, in the suffix-discovery application (and for the languages we examined), our results are consistent with the assumptions we made and the predictions we derived from them.

4 Conclusion

The overall purpose of this paper has been to illustrate what was for us an unexpected aspect of using Minimum Description Length theory: not only does MDL not specify the form of a grammar (or morphology), but it does not even specify the precise form in which the description of the abstract linkages between concepts (such as stems and signatures) should be encoded and quantitatively evaluated. We have seen that in a range of cases, using binary strings instead of the more traditional frequency-based pointers leads to a smaller overall grammar length, and there is no guarantee that we will not find an even shorter way to accomplish the same thing tomorrow[11]. Simply put, MDL is emphatically an evaluation procedure, and not a discovery procedure.

We hope to have shown, as well, that a systematic exploration of the nature of the difference between standard frequency-based pointer lengths and binary string based representations is possible, and we can develop reasonably accurate predictions or expectations as to which type of description will be less costly in any given case.

Acknowledgements

This research was supported by a grant of the Swiss National Science Foundation to the first author.

References

C. de Marcken. 1996. *Unsupervised Language Acquisition*. Ph.D. thesis, MIT, Cambridge, MA.

J. Goldsmith. 2001. The unsupervised learning of natural language morphology. *Computational Linguistics*, 27(2):153–198.

B. Mandelbrot. 1953. An informational theory of the statistical structure of language. In Willis Jackson, editor, *Communication Theory, the Second London Symposium*, pages 486–502. Butterworth: London.

[11] See note 5.

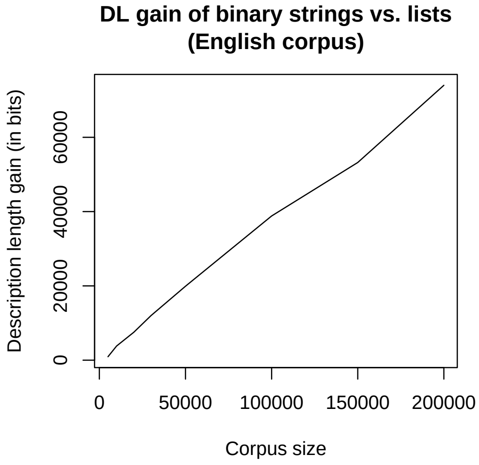

Figure 11: DL gain from using binary string versus size of corpus (English corpus)

J. Rissanen. 1989. *Stochastic Complexity in Statistical Inquiry*. World Scientific Publishing Co, Singapore.

C.E. Shannon. 1948. A mathematical theory of communication. *Bell Systems Technical Journal*, 27:379–423.