Unsupervised Learning of the

Morphology of a Natural Language

John Goldsmith

University of Chicago

Abstract

This study reports the results of using Minimum Description Length analysis to model unsupervised learning of the morphology of European languages, using corpora ranging in sizes from 5,000 word to 500,000 words. We develop a set of heuristics which rapidly develop a probabilistic morphological grammar, and use MDL as our primary tool to determine whether the modifications proposed by the heuristics will be adopted or not. The resulting grammar matches well the analysis that would be developed by a human morphologist.

In the final section, we discuss the relationship of this style of MDL grammatical analysis to the notion of *evaluation metric* in early generative grammar.

Version of April 25, 2000

A revised version of this paper will appear in *Computational Linguistics* (2001).

1. Introduction

This is a report on the present results of a study on unsupervised acquisition of morphology in European languages.[1] By the phrase “European languages,” I mean what we might call "stem + suffix" languages, though for present purposes I will restrict my discussion to Indo-European languages, returning in section 11 to the question of dealing with an unrestricted range of languages. The program in question takes a text file as its input (typically in the range of 5,000 to 1,000,000 words) and produces a partial morphological analysis of most of the words of the corpus; the goal is to produce an output that matches as closely as possible the analysis that would be given by a human morphologist. It performs unsupervised learning in the sense that the program's sole input is the corpus; we provide the program with the tools to analyze, but no dictionary and no morphological rules particular to any specific language. At present, the goal of the program is restricted to providing the correct analysis of words into component pieces (morphemes), though with only a rudimentary categorical labeling.

The underlying model that is utilized invokes the principles of the Minimum Description Length (MDL) framework (Rissanen 1989), which provides a helpful perspective for understanding the goals of traditional linguistic analysis. MDL focuses on the analysis of a corpus of data which is optimal by virtue of providing both the most compact representation of the data and the most compact means of extracting that compression from the original data. It thus requires both a quantitative account whose parameters match the original corpus reasonably well (in order to provide the basis for a satisfactory compression) and a spare, elegant account of the overall structure.

The novelty of the present account lies in the use of simple statement of morphological patterns (called *signatures* below) which aid both in quantifying the MDL

account and in constructively building a satisfactory morphological grammar (for MDL offers no guidance in the task of seeking the optimal analysis). In addition, the system whose development is described here sets reasonably high goals: the reformulation in algorithmic terms of the strategies of analysis used by traditional morphologists.

Developing an unsupervised learner using raw text data as its sole input offers several attractive aspects, both theoretical and practical. At its most theoretical, unsupervised learning constitutes a (partial) linguistic theory, producing a completely explicit relationship between data and analysis of that data. A tradition of considerable age in linguistic theory sees the ultimate justification of an analysis A of any single language L as residing in the possibility of demonstrating that analysis A derives from a particular linguistic theory LT, and that LT works properly across a range of languages (that is, not just for language L). There can be no better way to make the case that a particular analysis derives from a particular theory than to automate that process, so that all the linguist has to do is to develop the theory-as-computer-algorithm; the application of the theory to a particular language is carried out with no surreptitious help.

From a practical point of view, the development of a fully automated morphology-generator would be of considerable interest. It remains a project measured in man-weeks or months to produce a morphology of a given language, and from a purely practical point of view, we still need good morphologies of many European languages. With the advent of considerable historical text available online (such as the ARTFL data base of historical French), it is of great interest to develop morphologies of particular stages of a language, and the process of automatic morphology-writing can simplify this stage — where there are no native-speakers available — considerably.

A third motivation for this project is that it can serve as an excellent preparatory phase (in other words, a bootstrapping phase) for an unsupervised grammar acquisition system. As we will see, a significant proportion of the words in a large corpus can be assigned to categories, though the labels that are assigned by the morphological analysis are corpus-internal; nonetheless, the assignment of words into distinct morphologically motivated categories can be of great service to a syntax-acquisition device.

The problem, then, involves both the determination of the correct morphological split for individual words, and the establishment of accurate categories of stems based on the range of suffixes that they accept:

1. *Splitting words*: We wish to accurately analyze any word into successive morphemes in a fashion that corresponds to the traditional linguistic analysis. Minimally we wish to identify the stem, as opposed to any inflectional suffixes. Ideally, we would also like to identify all the inflectional suffixes on a word which contains a stem that is followed by two or more inflectional suffixes, and we would like to identify derivational prefixes and suffixes. We want to be told that in this corpus, the most important suffixes are *-s, -ing, -ed,* and so forth, while in the next corpus, the most important suffixes are *-e, -en, -heit, -ig,* and so on. Of course, the program is not a language identification program, so it will not name the first as "English" and the second as "German" (that is a far easier task), but it will perform the task of deciding for each word what is stem and what is affix.

2. *Range of suffixes*: The most salient characteristic of a stem in the languages that we will consider here is the range of suffixes with which it can appear. Adjectives in English, for example, will appear with some subset of the suffixes *-er/-est/-ity/-ness,* etc. We would like to determine automatically what the range of the most regular suffix

groups is for the language in question, and rank suffix groupings by order of frequency in the corpus.[2]

To give a sense of the results of the program, consider one aspect of its analysis of the novel *The Adventures of Tom Sawyer* -- and this result is consistent, by and large, regardless of the corpus one chooses. Consider the top-ranked signatures, illustrated in Table 1: a signature is an alphabetized list of affixes that appear with a particular stem in a corpus. A larger list of these patterns of suffixation in English are given in Table 2.

	Top signatures from *Tom Sawyer*		
Signature	**Example**	**Type count**	**Token count**
NULL.ed.ing	betray betrayed betraying	74	864
NULL.ed.ing.s	remain remained remaining remains	18	516
NULL.s.	cow cows	301	3414
e.ed.es.ing	notice noticed notices noticing	4	62

Table 1

The present morphology-learning algorithm is contained in a C++ program called *Linguistica* that runs on a desktop PC and which takes a text file as its input.[3] Analyzing a corpus of 500,000 words in English requires about five minutes on a Pentium II 333. Perfectly respectable results can be obtained from corpora as small as 5,000 words. The system has been tested the system on corpora in English, French, German, Spanish, Italian, Dutch, Latin, and Russian; some quantitative results are reported below. The corpora that serve as its input are largely materials that have been obtained over the Internet, and I have endeavored to make no editorial changes to the files that are the input. Occasionally this creates additional burdens for the program, when English control phrases creep into the files, an unintended consequence of this principle.[4]

In this paper, I will discuss prior work in this area (Section 2), the task of the initial splitting of words into stem and affix (Section 3), the notion of *signature* (Sections 4-7), including an elaboration of the model in the terms of Minimum Description Length (Section 5), results (section 8), the determination of spurious generalizations (section 9), the grouping of signatures into larger groupings of *paradigms* (Section 10), and directions for further improvements (section 11). Finally, I will offer some speculative observations about the larger perspective that this work suggests and work in progress (Section 12).

2 . Previous Research in this area, and MDL

The task of automatic word analysis has intrigued workers in a range of disciplines, and the practical and theoretical goals that have driven them has varied considerably. Some, like Zellig Harris (and the present writer), view the task as an essential one in defining the nature of the linguistic analysis. But workers in the area of data compression,

dictionary construction, and information retrieval have all contributed to the literature on automatic morphological analysis. (Note that our primary concern here is with morphology and not with regular allomorphy or morphophonology, which is the study of the changes in the realization of a given morpheme that are dependent on the grammatical context in which it appears, an area occasionally confused for morphology. Several researchers have explored the morphophonologies of natural language in the context of two level systems in the style of the model developed by Kimmo Koskenniemi (1983), Lauri Karttunen (1993), and others.) The only general review of work in this area that I am aware of is found in Langer (1991), and it is ten years old, and unpublished

Work in automatic morphological analysis can be usefully divided into four major approaches. The first approach proposes to identify morpheme *boundaries* first, and thus indirectly to identify morphemes, on the basis of the degree of predictability of the n+1st letter given the first n letters (or the mirror-image measure). This was first proposed by Zellig Harris (1955, 1967), and further developed others, notably by Hafer and Weiss (1974). The second approach seeks to identify bigrams (and trigrams) which have a high likelihood of being morpheme internal, a view pursued in work discussed below by Klenk, Langer, and others. The third approach focuses on the discovery of patterns (we might say, of *rules*) of phonological relationships between pairs of words. The fourth approach, which is presented in this paper, is top-down, and seeks an analysis which is globally most concise. In this section, we shall review some of the work that has pursued these approaches – briefly, necessarily.[5] While not all of the approaches discussed here use *no* prior language-particular knowledge (which is the goal of the present system), I exclude from discussions those systems which are based essentially on a prior human-designed analysis of the grammatical morphemes of a language, aiming at identifying the

stem(s) and the correct parsing; such is the case, for example, in Pacak and Pratt (1976), Koch et al. (1989) and Wothke and Schmidt (1992).

At the heart of the first approach, due to Harris, is the desire to place boundaries between letters (resp., phonemes) in a word based on conditional entropy, in the following sense. We construct a device which generates a finite list of words, our corpus, letter by letter and with uniform probability, in such a way that at any point in its generation (having generated the first n letters $l_1 l_2 l_3 \ldots l_n$) we can inquire of it what the entropy is of the set consisting of the next letter of all the continuations it might make. (In current parlance, we would most naturally think of this as a path from the root of a trie to one of its terminals, inquiring of each node its associated 1-letter entropy, based on the continuations from that node.) Let us refer to this as the *prefix conditional entropy*; clearly we may be equally interested in constructing a trie from the right-edge of words, which then provides us with a suffix conditional entropy, in mirror-image fashion.

Harris himself employed no probabilistic notions, and the inclusion of entropy in the formulation had to await Hafer and Weiss (1974); but allowing ourselves the anachronism, we may say that Harris proposed that local peaks of prefix (and suffix) conditional entropy should identify morpheme breaks. The method proposed in Harris (1955) appealed to what today we would call an oracle for information about the language under scrutiny, but in his (1967) article, Harris implemented a similar procedure on a computer and a fixed corpus, restricting his problem to that of finding morpheme-boundaries within words. Harris' method is quite good as a heuristic for finding a good set of candidate morphemes, comparable in quality to the mutual information-based heuristic that I have used, and describe below. It has the same problem that good

heuristics frequently have: it does not lend itself to a next step, a qualitatively more reliable approximation of the correct solution.[6]

Hafer and Weiss (1974) explore in detail various ways of clarifying and improving on Harris's algorithm while remaining faithful to the original intent. A brief summary does not do justice to their fascinating discussion, but for our purposes their results confirm the character of the Harrisian test as heuristic: with Harris's proposal, a quantitative measure is proposed (and Hafer and Weiss develop a range of different measures, all of them rooted in Harris's proposal), and best results for morphological analysis are obtained in some cases by seeking a local maximum, in others by seeking a value above a threshold, and in yet others, good results are obtained only when the measure is paired with a similar measure constructed in mirror-image fashion from the end of the word – and then some arbitrary thresholds are selected which yield the best results. While no single method emerges as *the* best, one of the best yields precision of 0.91 and recall of 0.61 on a corpus of approximately 6200 word types. (Precision here indicates proportion of predicted morpheme breaks that are correct, recall the proportion of correct breaks that are predicted.)

The second approach that can be found in the literature is based on the hypothesis that local information in the string of letters (resp., phonemes) is sufficient to identify morpheme boundaries. This hypothesis would be clearly correct if all morpheme boundaries were between pairs of letters l_1 - l_2 which never occur in that sequence morpheme-internally, and the hypothesis would be invalidated if conditional probabilities of a letter given the previous letter were independent of the presence of an intervening

boundary. The question is where real languages distribute themselves along the continuum that stretches between these two extremes.

A series of publications has explored this question, including Janssen (1992), Klenk (1992), Flenner (1994, 1995). Any brief description which overlooks the differences among these publications is certain to do less than full justice to all of them. The procedure described in Janssen (1992) and Flenner (1994, 1995) begins with a training corpus with morpheme boundaries inserted by a human, and hence the algorithm is not in the domain of unsupervised learning. Each bigram (and the algorithm has been extended in the natural way to treating trigrams as well) is associated with a triple (whose sum must be less than or equal to 1.0) indicating the frequency in the training corpus of a morpheme boundary occurring to the left of, between, or to the right of that bigram. In a test word, each space between letters (resp., phonemes) is assigned a score which is the sum of the relevant values derived from the training session: in the word *string*, for example, the score for the potential cut between *str* and *ing* is the sum of three values: the probability of a morpheme boundary after *tr* (given *tr*), the probability of a morpheme boundary between *r* and *i* (given *ri*), and the probability of a morpheme boundary before *in* (given *in*).

That these numbers should give *some* indication of the presence of a morpheme boundary is certain, for they are the sums of numbers which were explicitly assigned on the basis of overtly marked morpheme boundaries. But it remains unclear how further one should proceed with the sum. As Hafer and Weiss discover with Harris's measure, it is unclear whether local *peaks* of this measure should predict morpheme boundaries, or whether a threshold should be set, above which a morpheme boundary is predicted. (Flenner 1995: 64, 65). Janssen (1992, 81-82) observes that the French word *linguistique*

displays three peaks, predicting the analysis *lin-guist-ique*, employing a trigram model. The reason for the strong, but spurious, peak after *lin* is that *lin* occurs with high frequency word-finally, just as *gui* appears with high frequency word-initially. One could respond to this observation in several ways: word-final frequency should not contribute to word-internal morpheme-final status; or perhaps frequencies of this sort should not be added. Indeed, it is not clear to me why these numbers should be added (they do not represent probabilities that can be added). Janssen notes that the other two trigrams that enter into the picture (*ing* and *ngu*) had a zero frequency of morpheme break in the desired spot, and that the presence of any zeros in the sum forces the *sum* to be 0, raising again the question of what kind of quantity is being modeled.

I do not have room to discuss the range of greedy affix-parsing algorithms these authors explore, but that aspect of their work has less bearing on the comparison with the present paper, whose focus is on data-driven learning. But the major question to carry away from this approach is this: can the information which is expressed in the division of a set of words into morphemes be compressed into local information (bigrams, trigrams)? The answer, I believe, is in general negative. Morphology operates at a higher level, so to speak, and has only weak statistical links to local sequencing of phonemes or letters.[7]

The third approach focuses on the discovery of patterns explicating the overt shapes of related forms in a paradigm. Dzeroski and Erjavec (1997) report on work that they have done on Slovene, a South Slavic language with a complex morphology, in the context of a similar project. Their goal essentially was to see if an inductive logic program could infer the principles of Slovene morphology to the point where it could correctly predict the nominative singular form of a word if it were given an oblique (non-nominative) form. Their project apparently shares with the present one the requirement

that the automatic learning algorithm be responsible for the decision as to which letters constitute the stem and which are part of the suffix(es), though the details offered by Szeroski and Erjavec are sketchy as to how this is accomplished. In any event, they present their learning algorithm with a labeled pair of words — a base form, and an inflected form. It is not clear from their description whether the base form that they supply is a surface form from a particular point in the inflectional paradigm (the nominative singular), or a more articulated underlying representation in a generative linguistic sense; the former appears to be their policy.

Szeroski and Erjavec's goal is the development of rules couched in traditional linguistic terms; the categories of analysis are decided upon ahead of time by the programmer (or, more specifically, by the tagger of the corpus), and each individual word is identified with regard to what morphosyntactic features it bear. The form "bolecina" is marked, for example, as a feminine noun singular genitive. In sum, their project thus gives the system a good deal more information than the present project does.[8]

The fourth approach to morphology analysis is top-down, and seeks a globally optimal analysis of the corpus. Kazakov (1997)[9] presents an analysis along these lines, using a straightforward measurement of the success of a morphological analysis, which is to count the number of letters in the inventory of stems and suffixes that have been hypothesized; the improvement in this count over the number of letters in the original word-list is a measure of the fitness of the analysis. He used a list of 120 French words in one experiment, and 39 forms of the same verb in another experiment, and employed what he terms a genetic algorithm to find the best cut in each word. He associated each of the 120 words (resp., 39) with an integer (between 1 and the length of the word minus 1)

indicating where the morphological split was to be, and measured the fitness of that grammar in terms of its decrease in number of total letters. He does not describe the fitness function used, but seems to suggest that the single top-performing grammar of each generation is preserved, all others are eliminated, and the top-performing grammar is then subjected to mutation. That is, in a case-by-case fashion, the split between stems and suffixes is modified (in some cases by a shift of a single letter, in others by an unconstrained shift to another location within the word) to form a new grammar. In one experiment described by Kazakov, the population was set to 800, and 2,000 generations were modeled. On a Pentium 90 and a vocabulary of 120 items, the computation took over eight hours.

Work by Michael Brent (1993) and Carl de Marcken (1995) has explored analyses of the fourth type as well. Researchers have been aware of the utility of the information theoretic notion of compression from the earliest days of information theory, and there have been efforts to discover useful, frequent chunks of letters in text, such as Radhakrishnan (1978), but to my knowledge, Brent's and de Marcken's works were the first to explicitly propose the guiding of linguistic hypotheses by such notions. Brent's work addresses the question of determining the correct morphological analysis of a corpus of English words, given their syntactic category, utilizing the notion of minimal encoding, while de Marcken's addresses the problem of determining the "breaking" of an unbroken stream of letters or phonemes into chunks that correspond as well as possible to our conception of words, implementing a well-articulated algorithm couched in a Minimum Description Length framework, and exploring its effects on several large corpora. Brent (1999) provides a thorough review of work in this area, with special attention to work in the psychological literature.

Brent (1993) aims at finding the appropriate set of suffixes from a corpus, rather than the more comprehensive goal of finding the correct analysis for each word, both stem and suffix, and I think it would not be unfair to describe it as a test of concept trial on a corpus ranging in size from 500 to 8,000 words; while this is not a small number of words, our studies below focus on corpora with on the order of 30,000 distinct words. Brent indicates that he places other limitations as well on the hypothesis space, such as permitting no suffix which ends in a sequence that is also a sequence (i.e., if *s* is a suffix, then *less* and *ness* are not suffixes, and if *y* is a suffix, *ity* is not). Brent's observation is very much in line with the spirit of the present analysis: "The input lexicons contained thousands of non-morphemic endings and mere dozens of morphemic suffixes, but the output contained primarily morphemic suffixes in all cases but one. Thus, the effects of non-morphemic regularities are minimal" (35). Brent's corpora were different in the extreme from those used in the experiments reported below; his were based on choosing the *n* most common words in a *Wall Street Journal* corpus, while the present study has used large and heterogeneous sources for corpora, which makes for a considerably more difficult task. In addition, Brent scored his algorithm solely on how well it succeeded in identifying suffixes (or combinations of suffixes), rather than on how well it simultaneously analyses stem and suffix for each word, the goal of the present study.[10] Brent makes clear the relevance and importance of information theoretic notions, but does not provide a synthetic and over-all measure of the length of the morphological grammar.

De Marcken (1995) addresses a similar but distinct task, that of determining the correct breaking of a continuous stream of segments into distinct words. This problem has been addressed in the context of Asian (Chinese-Japanese-Korean) languages, where

standard orthography does not include white space between words, and it has been discussed in the context of language acquisition as well. De Marcken describes an unsupervised learning algorithm for the development of a lexicon using a Minimum Description Length framework. He applies the algorithm to a written corpus of Chinese, as well as to written and spoken corpora of English (the English text has had the spaces between words removed), and his effort inspired the work reported here. De Marcken's algorithm begins by taking all individual characters to be the baseline lexicon, and it successively adds items to the lexicon if the items will be useful in creating a better compression of the corpus in question, or rather, when the improvement in compression yielded by the addition of a new item to the codebook is greater than the length (or "cost") associated with the new item in the codebook. In general, a lexical item of frequency F can be associated with a compressed length of -log F, and de Marcken's algorithm computes the compressed length of the Viterbi-best parse of the corpus, where the compressed length of the whole is the sum of the compressed lengths of the individual words (or hypothesized chunks, we might say) plus that of the lexicon. In general, the addition of chunks to the lexicon (beginning with such high frequency items as "th") will improve the compression of the corpus as a whole, and de Marcken shows that successive iterations add successively larger pieces to the lexicon. De Marcken's procedure builds in a bottom-up fashion, looking for larger and larger chunks that are worth (in an MDL sense) assigning the status of dictionary entries. Thus, if we look at unbroken orthographic texts in English, the 2 letter combination "th" will become the first candidate chosen for lexical status; later, "is" will achieve that status too, and soon "this" will as well. The entry "this" will not, in effect, point to its four letters directly, but will rather point to the chunks "th" and "is", which still retain their status in the lexicon

(for their robust integrity is supported by their appearance throughout the lexicon). The creation of larger constituents will occasionally lead to the elimination of smaller chunks, but only when the smaller chunk appears almost always in a single larger unit.

An example of an analysis provided by de Marcken's algorithm is given in (1), taken from de Marcken 1995, in which I have indicated the smallest level constituent by placing letters immediately next to one another, and then higher structure with various pair brackets (parentheses, etc.) for orthographic convenience; there is no theoretical significance to the difference between "<>" and "()", etc. de Marcken's analysis succeeds quite well at identifying words, but does not make any significant effort at identifying morphemes as such.

(1)

< [t he] { ([un it] ed) ([st at] es) } > < of {a me ([r ic]) a } >

Applying de Marcken's algorithm to a "broken" corpus of a non-CJK language (for example, English) provides interesting results, but none that provide anything approaching a linguistic analysis, such as identification of stems and affixes. The broken character of the corpus serves essentially as an upper bound for the chunks that are postulated, while the letters represent the lower bound.

De Marcken's MDL-based figure of merit for the analysis of a substring of the corpus is the sum of the inverse log frequencies of the components of the string in question; the best analysis is that which minimizes that number (which is, again, the optimal compressed length of that substring), plus the compressed length of each of the lexical items that have been hypothesized to form the lexicon of the corpus. It would certainly be natural to try using this figure of merit on words in English, along with the constraint that all words should be divided into exactly two pieces. Applied straightforwardly, however, this gives uninteresting results: words will always be divided into two pieces, where one of the pieces is the first or the last letter of the word, since individual letters are so much more common than morphemes.[11] (I will refer to this effect as “peripheral cutting” below.) In addition – and this is less obvious – the hierarchical character of de Marcken’s model of chunking leaves no place for a qualitative difference between high-frequency “chunks”, on the one hand, and true morphemes, on the other: *str* is a high-frequency chunk in English (as *schl* is in German), but it is not at all a morpheme. The possessive marker *‘s*, on the other hand, is of relatively low frequency in English, but is clearly a morpheme.

MDL is nonetheless the key to understanding this problem. In the next section, I will present a brief description of the algorithm used to bootstrap the problem, one which avoids the trap already mentioned briefly in note 11. This provides us with a set of candidate splittings, and the notion of the signature of the stem becomes the working tool for determining which of these splits is linguistically significant. MDL is a framework for evaluating proposed analyses, but it does not provide a set of heuristics that are nonetheless essential for obtaining candidate analyses, which will be the subject of the next two sections.

3. The nature of the initial split of a word

We have explored two rather different heuristics for obtaining an initial split of words into stem and suffix. For present purposes, the two work equally well and provide very similar results; one is much quicker and less demanding of computational resources. We will call the first the *split-all-words* strategy, and the second the *ngram-MI* ("mutual information") strategy.[12]

4.1 *Split-all-words*. Under this strategy, we begin by seeking (or rather, imposing) an initial parsing of all words of two or more letters into two pieces, which I shall refer to as *stem* and *suffix*, although at first the significance of those terms is simply that all words are divided into two pieces, and the stem precedes the suffix. From a purely terminological point of view, it might have been preferable to refer to the stem as the *prefix*, and to say that all words are composed of a prefix followed by a suffix. However, substantial content corresponding to the traditional notions of stem and suffix will be associated with these terms later in the program, as we shall see.

The first stage of the algorithm, then, seeks an optimal division of each word into stem and suffix. What does "optimal" mean in this case? *Audacity*, for example, should be parsed as *audac + ity*, and this either because we might say we simply know that these are the morphemes, or because we can refer to the related words *audacious* and the range of words ending in *ity* (*sanity, obesity, paucity*, etc.). In the case of a word such as *stomach*, on the other hand, no particular parsing is of any use from a morphological point of view: *stom + ach* (or any other parsing) provides us with no link to other words. In the case of words such as *the, this, that, those*, should they be divided after the initial *th* sequence? Is the *th* some kind of morpheme? The answer is not entirely obvious, as are a number of other cases in different languages. Our algorithm decides that *th-* is neither a stem nor a prefix, and upon reflection, that is almost surely the correct answer.

But the clearest of all, obviously, are words like *dogs* (= *dog+s*) and *fights* (= *fight+s*). Only slightly less clear are cases like *likes*, which might best be divided as *like+s*, but which might be parsed as *lik+es*, on the grounds that *liking* should be brought into alignment with *like* and *likes*, and the analysis *lik+e+s* is also quite reasonable. In the overwhelming majority of cases in the material that we will consider below, it is clear what parsing we (as linguists on the outside) would like the algorithm to produce (but see note 27). We must provide the program with enough judgment to model what it is that we look for when we split a word into a stem and its suffixes.

We would like to have a figure of merit for each of the possible parses of a word W into Stem_i and Suffix_j which reflects the fact that a hypothetical stem S_i is a good candidate as a stem if and only if S_i is a good candidate as a stem in many other words, and Suffix_j is as well, in parallel fashion. As we have noted (see note 11), using the sum

of the inverse log frequencies of the component parts leads to a peripheral cut, which is linguistically incorrect in most cases; we minimally need greater weight to be placed on morphemes which are longer. A natural measure to try, then, is the sum of the products of the log frequencies times the length in letters, a figure of merit which pushes the optimal cut towards the middle of the word. We will return below (section 5) to the question as to why this is a reasonable step to take. As we have noted, the strategies described in this section constitute heuristics, and are distinct from the MDL *model* presented below. We will use the familiar notation by which |*w*| is the length of word *w* in letters, and [*w*] is number of occurrences of word *w* in the corpus; *W* will represent the entire set of words in the corpus, and we will extend the notation so that [*W*] represents the number of words in the corpus.

We first establish two structures into which we place candidate stems and suffixes, and associate with each a somewhat hypothetical figure estimating how many times each occurs in the corpus. Employing Expectation-Maximization (Dempster et al 1977), on our first pass through the lexicon, we consider all possible parses of each word *w* as equally likely, and since in general there are |*w*|-1 ways to parse a word of |*w*| letters, this means that each stem is assigned an occurrence of [*w*] /(|*w*| - 1) by virtue of appearing in word *w*. We perform this counting operation on all the words in the lexicon. On successive passes, we evaluate the goodness of a parse by means of the measure in (2).

(2) (a) $H(stem/suffix) = -(|stem| * \log freq(stem) + |suffix| * \log freq(suffix)$

or more generally,

(b) $H(stem/suffix) = - \quad |morpheme_i| * \log freq(morpheme_i)$

Using the frequency values established on the previous iteration, for each word, we consider all possible parses, and assign a value to each parse by the H function in (2). Having computed the H function for all |*w*|-1 parses, we distribute *w*'s count among the parses using H in a Boltzmann distribution, so that the probability mass associated with a particular stem/suffix split S_i is proportional to $e^{H(S_i)}$. Thus for any given word *w*, we establish a normalization term Z, which is the sum over all parses P_i of the figure of merit, i.e., $Z = \sum_i e^{H(S_i)}$. Then the model assigns a probability, to any given split of a word, of $\frac{1}{Z} e^{H(S_i)}$.

For each word, we note what the best parse is, that is, which parse has the highest rating by virtue of the H-function. We iterate until no word changes its optimal parse, which empirically is typically less than 5 iterations on the entire lexicon.[13] We now have an initial split of all words into stem plus suffix. Even for words like *this* and *stomach* we have such an initial split.

4.2 *N-gram MI*. The second approach which we have employed provides a much more rapid convergence on the suffixes of a language. Since our goal presently is to identify word-final suffixes, we assume by convention that all words end with an end-of-word symbol (traditionally "#"), and we then tally the counts of all n-grams of length between 2 and 6 letters that appear word-finally. Thus, for example, the word *elephant#* contains 1 occurrence of the word-final bigram *t#*, 1 occurrence of the word-final 3-gram *nt#*, and so forth; we stop at 6-grams, on the grounds that no grammatical morphemes

require more than 5 letters in familiar languages. We also require that the n-gram in question be a proper substring of its word.

We employ as a rough indicator of likelihood that such an n-gram $n_1n_2\ldots n_k$ is a grammatical morpheme the measure: $\frac{[n_1n_2\ldots n_k]}{Total\ count\ of\ \ k-grams}\log\frac{[n_1n_2\ldots n_k]}{[n_1][n_2]\ldots[n_k]}$. We choose the top 100 n-grams on the basis of this measure as our set of candidate suffixes.

We should bear in mind that this ranking will be guaranteed to give wrong results as well as correct; for example, while *ing* is very highly ranked in an English corpus, *ting* and *ng* will also be highly ranked, the former because so many stems end in *t*, the latter because all *ing*s end in *ng*, but of the three, only *ing* is a morpheme in English.

We then parse all words into stem plus suffix if such a parse is possible using a suffix from this candidate set. A considerable number of words will have more than one such parse under those conditions, and we utilize the figure of merit described in the preceding section to choose among those potential parses.

4.3 Regardless of which of the two approaches we have taken, our task now is to decide which splits are worth keeping, which ones need to be dropped, and which ones need to be modified. In addition, if we follow the *split-all-words* approach, we have many splits which (from our external vantage point) are splits between prefix and stem: words beginning with *de* (*defense, demand, delete*, etc.) will at this point all be split after the initial *de*. So there is work to be done, and for this we introduce the central notion of a *signature.*

4. Signatures

Each word now has been assigned an optimal split into stem and suffix, and we consider henceforth only the best parse for that word, and we retain only those stems and suffixes that were optimal for at least one word. For each stem, we make a list of those suffixes that appear with it, and we call an alphabetized list of such suffixes (separated by an arbitrary symbol, such as period) the stem's *signature*; we may think of it as a mini-paradigm. For example, in one English corpus, the stems *despair, pity, appeal, insult* appear with the suffixes *ing* and *ingly*. However, they also appear as free-standing words, and so we indicate that with the word "NULL", to indicate a zero suffix. Thus their signature is *NULL.ing.ingly*. Similarly, the stems *assist* and *ignor* [sic] are assigned the signature *ance.ant.ed.ing* in a certain corpus. Because each stem is associated with exactly one signature, we will also use the term *signature* to refer to the set of affixes along with the associated set of stems when no ambiguity arises.

We establish a data structure of all signatures, keeping track for each signature of which stems are associated with that signature. As an initial heuristic, subject to correction below, we discard all signatures that are associated with only one stem (these latter form the overwhelming majority, well over 90%) and all signatures with only one suffix. The remaining signatures we shall call *regular signatures*, and we will call all of the suffixes that we find in them the *regular suffixes*. As we shall see, the regular suffixes are not quite the suffixes we would like to establish for the language, but they are a very good approximation, and constitute a good initial analysis. The non-regular signatures produced by the *split-all-words* approach are typically of no interest, as examples such as "*ch.e.erial.erials.rimony.rons.uring* " and "*el.ezed.nce.reupon.ther*" illustrate. The reader may identify the single English pseudo-stem that occurs with each of these signatures.

The regular signatures are thus those that specify exactly the entire set of suffixes used by at least two stems in the corpus. The presence of a signature rests upon the existence of a structure as in (3), where there are at least two members present in each column, and all combinations indicated in this structure are present in the corpus, and, in addition, each stem is found with no other suffix. (This last condition does not hold for the suffixes; a suffix may well appear in other signatures, and this is the difference between stems and affixes.) We will say that in a pattern as in (3) the suffixes may be *commuted* with each other, and likewise the stems; structuralist linguists would refer to two elements as being commutable if they appeared in the same left- and right-hand environment.[14]

(3) $\left\{ \begin{matrix} stem_1 \\ stem_2 \\ stem_n \end{matrix} \right\} \left\{ \begin{matrix} suffix_1 \\ suffix_2 \end{matrix} \right\}$

If we have a morphological pattern of 5 suffixes, let us say, and there is a large set of stems that appear with all 5 suffixes, then that set will give rise to a regular signature with 5 suffixal members. This simple pattern might be perturbed by the (for our purpose) extraneous fact that a stem appearing with these suffixes might also appear with some other suffix; and if all stems that associate with these 5 suffixes appear with idiosyncratic suffixes (i.e., each different from the others), then the signature of those 5 suffixes would never emerge. In general, however, in a given corpus, a good proportion of stems appears with a complete set of what a grammarian would take to be the paradigmatic set of suffix for its class: this will be neither the stems with the highest nor the stems with the lowest frequency, but those in-between. In addition, there will be a large range of words with no acceptable morphological analysis, which is just as it should be: *John, stomach, the*, and so forth.

5. An MDL model

What we have considered so far constitutes a partial set of heuristics for the discovery of stems and suffixes in a language such as English. But our goal should be to characterize a morphology in an abstract and quantitative way so as to make it possible to view identification of the best morphology as an optimization problem. This goal is not so characterized in order to satisfy computational needs; it is also a statement of our desire to explicitly characterize what a good morphological analysis is from a linguist's point of view. We wish to formulate a general conception of morphology that rewards a particular analysis for being concise and representing generalizations where possible.

A natural model to use in such a case is the Minimum Description Length framework, whose fundamental tenet is that the optimal means of modeling a set of data is that which jointly provides the best compressed length of the data while requiring the least amount of information to do so. For a model to provide a good compressed length of the data in the case we are looking at — the morphology of words — means essentially that the model is able to provide good statements regarding the frequencies of the various words, and it is able to do so on the basis of the frequencies it infers for the various morphemes. This is relatively straightforward, but what does it mean for a model to do so while requiring little information? The condition is essentially one of conciseness: we must find a method that quantifies, in bits, all of the information that a morphology infers about the word structure of a given corpus, and we will seek the most compact such description.

There is a much deeper connection between MDL and the problem of morphology discovery, which I will sketch in the text, and describe in greater detail in a note.

Consider the assumptions made by a word-list of length N, such as we use as our data input. Implicit in this characterization is the notion that words can be described at two levels: (a) at the word-level, in which a word can be conceived of as a single thing, and as a sequence of letters; (b) as a sequence of letters, the word list is simply a selection from among the possible strings of letters chosen from the alphabet. If I should ask for a word, the request could be taken in one of two ways: either (a) as a request for the selection of a word from the word list (which in effect amounts to choosing a number from 1 to N), or (b) as a request to compose a sequence of letters from the alphabet, which amounts, in the case of a word of *n* letters, to choosing from a candidate set of size 26^n (in an alphabet with 26 letters). Clearly, in the second case, the size of the degree of freedom is much, much larger, but most of those options would not be English, nor any other language. The crucial point is that a morphology is a system with degrees of freedom in-between these two. If we restrict ourselves to the case where the morphological split is made between stem and suffix, we can show that the decrease in the universe of word-possibilities corresponds roughly to the minimum description length of the morphology.[15]

With this understanding, we can return to the question as to why the figure of merit *H* is based on the sum of the logarithmic frequencies *multiplied by length, in letters*. As such, *H* appears to have going for it little more than the fact that it provides a very good, though certainly not flawless, first cut of words of two morphemes. But why should this be so? Why should a morpheme's contribution to the well-formed splitting of a word be increased by virtue of being longer?

The reason is simply this: *the frequency of a letter*[16] *is a very poor approximation of how likely that letter is to be a morpheme*. The likelihood that a letter of frequency f_1 is

a morpheme is *not* greater, all other things being equal, than that of a letter (phoneme) of frequency $f_2 < f_1$. And this is even more important if we wish to understand what we are doing: the process of finding the morphemes of a corpus of a language *can* be viewed as a process of finding an optimal compression scheme, but only if the compression scheme is not permitted to mix morphological representation and letter- (or phoneme-) based representation. It is this which accounts for why de Marcken's account cannot be extended to a solution of the morpheme-discovery procedure, but this is by no means obvious. Let us expand on this observation, and make explicit the MDL model that we have applied. I will assume that the reader is familiar enough with information theory to be comfortable with the idea that the length of a representation of an object of raw frequency F selected from a well-defined set of total frequency T is equal to log T/F. [17]

The model of morphology that we use is illustrated in Figure 1. It consists of three major parts: a list of unanalyzable stems (what a linguist might call *roots*), a list of affixes, and a list of organizers of this information, which is to say, a list of signatures. Each signature consists of a list of pointers to its stems, and a list of pointers to its affixes.

As we have noted briefly, the hallmark of an MDL analysis is the computation of both the compressed length of the corpus and the length of the analysis which has achieved that compression. Optimizing the compression of the corpus in this sense is equivalent to coming up with the best model of the frequencies of the words of the corpus. The very best model of the frequencies of each word is always obtained by simply counting the number of occurrences, but that involves no morphology: each word is treated as an atomic unit. To posit a morphology is to posit the hypothesis that the frequencies of any given polymorphemic word is computed, not counted; the prediction

is based on three computable subchoices: the probability of any given signature (which is to say, pattern of morphological analysis), the probability of a given stem, and the probability of a given suffix in that signature. That prediction will never be as accurate as the value derived simply from counting, but that is not the issue here; what we care about is *comparing* different predictions from different morphologies. Each morphology has a limited amount of probability mass; it can only distribute up to 1.00, so the morphology that wastes too much of its probability mass on words not observed in the corpus will have less probability mass to distribute to the words actually seen, and hence the compressed length of the corpus under its regime will be longer.[18]

In what follows, it will be important to distinguish two distinct subparts of the lexicon and of the corpus. We shall refer to the set of all words as *W*, and decompose *W* into two parts: W_{SIMPLE}, consisting of words with a stem plus suffix, or simply a stem with no suffix, and secondly $W_{COMPLEX}$, consisting of words in which the stem itself is morphologically complex, consisting of a stem plus suffix. In the analysis that follows, the variable *f* stands for a *suffix*, *t* for a *stem*, and σ for a *signature*. As noted above, we will use square brackets "[W]" to indicate token-counts, either of a specific word, morpheme, or a category, depending on context, while angled brackets "<Stems>", will indicate type-counts, i.e., the number of orthographically distinct stems.

How, then, do we compute a *length* of a morphology? We offer an answer in (4) below, which we will discuss here term by term. The length of the morphology is the sum of the lengths of the suffixes, stems, and signatures, plus some organizational overhead, and we will review each one in turn. The organizational overhead is the information that normally would be built into the graphical structure of a model in an illustration such as

Figure 1, but which remains to be treated in a quantitative fashion. The morphology consists of a list of pointers organized into three successive groups, one to each suffix, stem, and signatures, prefixed by a number indicating how many such pointers are in each group. These three numbers take log (number of suffixes) + log (number of stems) + log (number of signatures) bits of information, using *log(x)* as a measure of the length of a number (see (4(i)).

Let us consider the length of the suffix list.[19] It consists of a list of pointers to the suffixes of the language, and the list itself of those suffixes. The pointer to a suffix *f* will be of length $\log\frac{[W]}{[f]}$, where the denominator is the count of the suffix, and the length of the entry for each suffix will be directly proportional to the number of letters in the suffix, with a factor of proportionality λ, which we take to be log (A), where *A* is the number of letters in the alphabet employed in the corpus.[20] Thus the combined length of the suffix component of the morphology is as given in (4ii).

In an entirely parallel fashion, the length of the stem component consists of the lengths of the list of pointers to each of the stems, plus the lengths of each of the stems in letters; see (4iii).

The signature component is more complex. The information in the signature component breaks down into the following subcomponents, and we will use the organization that follows in the explicit quantification of description length in later equations:

a) List of pointers to each signature;

b) For each signature:

(i) The number of stems in the signature;

(ii) The number of suffixes in the signature;

(iii) A pointer to each stem, which consists of:

i. Simple/complex flag;

ii. Stem-pointer

(iv) A pointer to each suffix

Let us review each part in turn. We have, as with the preceding suffix and stem components, a list of pointers to each signature. This is followed by the representation of each signature. For each signature, the representation consists of, first, information describing how many stems and how many suffixes are present, plus a list of pointers specifying the stems followed by a list of pointers specifying the suffixes.

A pointer to a stem can take one of two forms, depending on whether the word is in W_{SIMPLE} or $W_{COMPLEX}$. In the latter case, the word contains nested morphological structure which we take into account. A signature entry with a simple stem contains a flag indicating that it is a simple stem along with a pointer to the stem in question, which is found in the stem component. A signature entry with a complex stem contains a flag indicating that it is a complex stem, followed by the information needed to specify the complex stem (we return to that momentarily).

Thus, every stem-entry in a signature begins with a flag indicating what kind of stem it is, and this flag will be of length $\log \frac{[W]}{[W_{SIMPLE}]}$ for simple stems, and of length $\log \frac{[W]}{[W_{COMPLEX}]}$ for complex stems.

For simple stems, then, this flag will be followed by a pointer to a stem *t* in the stem list, which will be of length $\log\frac{[W]}{[t]}$, while for complex stems, the flag will be followed by three pointers: a pointer to a signature σ, a pointer to a stem *t* within the signature, and a pointer to a suffix *f* in the signature; these three pointers jointly specify the word which serves as the morphological base of the present word. These three pointers are, respectively, of length: $\log\frac{[W]}{[\sigma]} + \log\frac{[\sigma]}{[t]} + \log\frac{[\sigma]}{[f\ in\ \sigma]}$ (on the last term, see immediately below).

Finally, the signatures also include a set of suffix pointers; a pointer inside signature σ to a suffix *f* is of length $\log\frac{[\sigma]}{[words(f) \cap words(\sigma)]}$, where the denominator is the count of the words ending in the suffix that belong to the signature.[21] We will indicate this simply as $\log\frac{[\sigma]}{[f\ in\sigma]}$.

Note that the signature list will contain one signature for all of those stems which have no suffixes at all. We have chosen *not* to attribute to these words the presence of the same phonologically null suffix found in paradigms such as *dog/dogs*. This decision has precise linguistic consequences, but we have not been able to test them yet, pending the incorporation into the model of predictions of counts of signatures from paradigms (see Section 11).

This completes the specification of the list of the morphology as such.

Now in compressing the corpus, each word is represented by three pointers (and the length of its representation is the sum of the lengths of these three pointers): a pointer to

its signature, a pointer to its stem within the signature, and a pointer to its suffix within the signature. The lengths of the first two appear to be straightforward and unambiguous: the length of the pointer to the signature is $\log\frac{[W]}{[\sigma]}$, the length of the pointer to the stem within the signature is $\log\frac{[\sigma]}{[t]}$, and the length of the pointer to the suffix is the log of the ratio of the number of words in the signature to the number of words in that subset containing the suffix.

We must distinguish between the size of *W* used to compute the size of the lexicon, and the closely related, but distinct, *W* used in computing the compressed length of the corpus. Let us refer to the corpus-based set as W_{raw}. The *W* which we need to compute the relevant numbers in the morphology is different from W_{raw}, in two ways. First of all, as we will see below, there are occasionally "virtual" words that the analysis decides must exist; these words, however, have no counts in the corpus. Second, and more importantly, the distribution over words is somewhat different in the morphology and the corpus because of the nested morphological structure that we have discussed several times. Thus, the word *savings* contains two words: *savings* and *saving*, and both contribute to the count of *W*. We will need to distinguish between counts of a word *w* where *w* is a free-standing word, and counts where it is part of a larger word; we shall refer to the latter class as *secondary counts*. In order to simplify computation and exposition, we have adopted the convention that the total number of words remains fixed, even when nested structure is posited by the morphology, thus forcing the convention that counts are distributed in a non-integral fashion over the two or more nested word-structures found in complex words. We consider the more complex case in the Appendix.

The basic equations are given in (4) and (5).

(4) Compressed length of morphology

(*i*) $\log < \textit{suffixes} > + \log < \textit{stems} > + \log < \textit{signatures} >$

(*ii*) *Suffix list* $\sum_{f \in Suffixes} \left(\lambda^* | f | + \log \frac{[W_A]}{[f]} \right)$

(*ii*) *Suffix list* $\sum_{f \in Suffixes} \left(\lambda^* | f | + \log \frac{[W_A]}{[f]} \right)$

(*iii*) *Stem list* : $\sum_{w \in W_{UN} \cup W_{SIMPLE}} \left(\lambda^* | stem(w_i) | + \log(\frac{[W]}{[stem(w_i)]}) \right)$

(iv) *Signature component*

Stated once for the whole component:

(a) Signature list: $\sum_{\sigma \in Signatures} \log \frac{[W]}{[\sigma]}$

For *each* signature:

(b) Size of the count of the number of stems plus size of the count of the number of suffixes: log <stems(σ)> + log <suffixes(σ)>

(c) A pointer to each stem, consisting of a simple/complex flag, and a pointer to either a simple or complex stem:

(i) Case of simple stem: flag of length

$$\log\frac{[W]}{[W_{SIMPLE}]}$$ plus a pointer to a stem of

length $\log\frac{[W]}{[t]}$; or

(ii) Case of complex stem: flag of length

$\log\frac{[W]}{[W_{COMPLEX}]}$, followed by a sequence of two pointers of

total length $\log\frac{[W]}{[stem(t)]} + \log\frac{[\sigma]}{[suffix(t)\ in\ \sigma]}$.

(d) a pointer to each suffix, of total length $\sum_{f \in Suffixes(\sigma)} \log\frac{[\sigma]}{[f\ \ in\ \sigma]}$.

(5) Compressed length of corpus

$$\sum_{w \in W} [w]_{raw} \left\lceil \log\frac{[W]}{[\sigma(w)]} + \log\frac{[\sigma(w)]}{[stem(w)]} + \log\frac{[\sigma(w)]}{[suffix(w) \cap \sigma(w)]} \right\rceil$$

MDL thus provides a figure of merit which we wish to minimize, and we will seek heuristics which modify the morphological analysis in such a fashion as to decrease the this figure of merit in a large proportion of cases. In any given case, we will accept a modification to our analysis just in case the description length decreases, and we will suggest that this strategy coincides with traditional linguistic judgment in all clear cases.

It remains to establish the length of a letter in the informational units used to represent the length of a pointer. It is reasonable that this value would be proportional to the inverse log frequency of the letter, as already noted, or the log of the number of letters, but more generally, it seems to me that it remains an open question what the weighting should be, for this value determines the relative cost of generalization in a morphology. That is, suppose we take the length of a pointer to a letter *a* to be $-\lambda \log(\frac{1}{freq(a)})$. If we choose λ to be large (λ >> 1), then savings in compression of the corpus are relatively more important, compared to savings in the parsimony of the morphological generalizations, while if λ << 1, then the simplicity of the morphology is relatively more important. Very general considerations suggest taking λ to be 1, but in the long run, we will want to consider the value of this parameter very carefully.[22]

The reader will note that the form of the primary heuristic that we have described so far, given graphically as the signature in (3) above, through its combination of information regarding both frequency and length, allows it to serve as an extremely rough approximation of the considerations which MDL computes in a precise fashion. The relatively high frequency of a single letter is of relatively little interest from a

morphological point of view; the savings to be gained in a thorough-going MDL description do not "kick in," so to speak, until the chunks succeed both in providing good compression and spare description. It should be clear that organization of data into large signatures is the ideal way to organize phonological material from the MDL point of view. It serves to minimize the length of the description of the morphology, in the sense that a signature with *m* stems and *n* suffixes (of length of the order of *m* + *n*) summarizes *m***n* words. It provides a compression method of the corpus which will be poorer than a method based on actual counts only to the extent that the frequency of a word diverges from the product of the frequency of its stem and the frequency of its suffix.

Let us return, then, to the discussion of the signatures actually encountered in a natural language corpus.

6. Initial inspection of regular signatures in English

As we have noted, the MDL framework functions to choose between analyses of data, but it does not provide us with the candidate analyses. In sections 3 and 4 we developed an initial set of heuristics which provide a baseline morphological analysis of a language such as English or French. Let us examine the sorts of results its provides, and see what it does well and where it errs.

To get a sense of what are identified as regular signatures in familiar language such as English, let us look at the results of a preliminary analysis of the the 86,976 words of *(The Adventures of) Tom Sawyer*, by Mark Twain. The signatures in Table 2 are ordered by the "breadth" of a signature, defined as follows. A signature S has both a *stem-count* (the number of stems associated with it) and an *affix count* (the number of affixes it contains), and we use log (stem count) * log (affix count) as a rough guide to the

centrality of a signature in the corpus. The suffixes identified are given in **Error! Reference source not found.**

In this corpus of some 87,000 words, there are 202 regular signatures identified through the procedure we have outlined so far (that is, preceding the refining operations described in the next section), and 803 signatures composed entirely of regular suffixes (the 601 additional signatures either have only one suffix, or pertain to only a single stem).

The top five signatures are: *NULL.ed.ing, e.ed.ing, NULL.s, NULL.ed.s,* and *NULL.ed.ing.s*; the third is primarily composed of noun stems (though it includes a few words from other categories -- *hundred, bleed, new),* while the others are verb stems. Number 7, *NULL.ly,* identifies 105 words, of which all are adjectives (*apprehensive, sumptuous, gay,*…) except for *Sal, name, love, shape,* and perhaps *earth*. The results in English are typical of the results in the other European languages which I have studied.

These results, then, are those derived by the application of the heuristics described above. The overall sketch of the morphology of the language is remarkably correct in its outlines. Nevertheless, the results, when studied up close, show that there remain a good number of errors which must be uncovered using additional heuristics and evaluated using the MDL measure. These errors may be organized in the following ways:

1. The collapsing of two suffixes into one: for example, we find the suffix *ings* here; in most corpora, the equally spurious suffix *ments* is found.
2. The systematic inclusion of stem-final material into a set of (spurious) suffixes. In English, for example, the high frequency of stem-final *t*s leads the system to analyze a set of suffixes as in the spurious signature *ted.ting.ts,* or *ted.tion*

3. The inclusion of spurious signatures, largely derived from short stems and short suffixes, and the related question of the extent of the inclusion of signatures based on real suffixes but overapplied. *s*, for example, is a real suffix of English, but not every word ending in *s* should be analyzed as containing that suffix. On the other hand, every word ending in *ness* should be analyzed as containing that suffix (in this corpus, this reveals the stems: *selfish, uneasi, wretched, loveli, unkind, cheeri, wakeful, drowsi, cleanli, outrageous,* and *loneli*). In the initial analysis of *Tom Sawyer*, the stem *ca* is posited with the signature *n.n't.p.red.st.t.*

4. The failure to break all words actually containing the same stem in a consistent fashion: for example, the stem *abbreviate* with the signature *NULL.d.s* is not related to *abbreviat* with the signature *ing*.

5. Stems may be related in a language without being identical. The stem *win* may be identified as appearing with the signature *NULL.s* and the stem *winn* may be identified with the signature *er.ing*, but these stems should be related in the morphology.

In the next section, we discuss some of the approaches we have taken to resolving these problems.

7. Optimizing description length using heuristics and MDL

We can use the description length of the grammar formulated in (4) and (5) to evaluate any proposed revision, as we have already observed: note the description length of the grammar and the compressed corpus, perform a modification of the grammar, recompute the two lengths, and see if the modification improved the resulting description length.[23]

Following the morphological analysis of words described in the previous section, suffixes are checked to determine if they are spurious amalgams of independently motivated suffixes: *ments* is typically, but wrongly, analyzed as a suffix. Upon identification of such suffixes as spurious, the vocabulary containing these words is reanalyzed. For example, in *Tom Sawyer*, the suffix *ings* is split into *ing* and *s,* and thus the word *beings* is split into *being* plus *s*; the word *being* is, of course, already in the lexicon. The word *breathings* is similarly reanalyzed as *breathing* plus *s*, but the word *breathing* is *not* found in the lexicon; it is entered, with the morphological analysis *breath+ing*. Words that already existed include *chafing, dripping, evening, feeling,* and *flogging,* while new "virtual" words include *belonging, bustling, chafing,* and *fastening*. The only new word that arises that is worthy of notice is *jing*, derived from the word *jings* found in Twain's expression *by jings!*. In a larger corpus of 500,000 words, 64 suffixes are tested for splitting, and 31 are split, including *tions, les, ists, ians, ened, lines, ents,* and *ively*.[24]

Following this stage, the signatures are studied to determine if there is a consistent pattern in which all suffixes from the signature begin with the same letter or sequence of letters, as in *te.ting.ts*.[25] Such signatures are evaluated to determine if the description length improves when such a signature is modified to become *e.ing.s*. It is necessary to precede this analysis by one in which all signatures are removed which consist of a single suffix composed of a single letter. This set of signatures includes, for example, the singleton signature *e*, which is a perfectly valid suffix in English; however, if we permit all words ending in *e*, but having no other related forms, to be analyzed as containing the suffix *e*, then the *e* will be inappropriately highly valued in the analysis. (We return to

this question in section 11, where we address the question of how many occurrences of a stem with a single suffix we would expect to find in a corpus.)

In the next stage of analysis, *triage*, signatures containing a small number of stems or a single suffix are explored in greater detail. The challenge of *triage* is to determine when the data is rich and strong enough to support the existence of a linguistically real signature. A special case of this is the question of how many stems must exist to motivate the existence of a signature (and hence, a morphological analysis for the words in question) when the stems only appear with a single suffix. For example, if a set of words appear in English ending with *hood*, should the morphological analysis split the words in that fashion, even if the stems thereby created appear with no other suffixes? And, at the other extreme, what about a corpus which contains the words *look, book, loot*, and *boot?* Does that data motivate the signature *l.k*, for the stems *boo* and *loo*? The matter is rendered more complex by a number of factors. The length of the stems and suffixes in question clearly plays a role: suffixes of one letter are, all other things being equal, suspicious; the pair of stems *loo* and *boo*, appearing with the signature *k.t*, does not provide an example of a convincing linguistic pattern. On the other hand, if the suffix is long enough, even one stem may be enough to motivate a signature, especially if the suffix in question is otherwise quite frequent in the language. A single stem occurring with a single pair of suffixes may be a very convincing signature for other reasons as well. In Italian, for example, even in a relatively small corpus we are likely to find a signature such as *a.ando.ano.are.ata.ate.ati.ato.azione.ò* with several stems in it; once we are sure that the ten-suffix signature is correct, then the discovery of a sub-signature along with a stem is perfectly natural, and we would not expect to find multiple stems associated with each of the occurring combinations. (A similar example in English from

Tom Sawyer is *NULL.ed.ful.ing.ive.less* for the single stem *rest.*) And a signature may be "contaminated," so to speak, by a spurious intruder. A corpus containing *rag, rage, raged, raging,* and *rag* gave rise to a signature: *NULL.e.ed.ing.s* for the stem *rag*. It seems clear that we need to use information that we have obtained regarding the larger, robust patterns of suffix combinations in the language to influence our decisions regarding smaller combinations. We return to the matter of triage below.

We are currently experimenting with methods to improve the identification of related stems. Current efforts yield interesting but inconclusive results. We compare all pairs of stems to determine whether they can be related by a simple substitution process (1 letter for none, 1 letter for 1 letter, 1 letter for 2 letters), ignoring those pairs that are related by virtue of one being the stem of the other already within the analysis. We collect all such rules, and compare by frequency. In a 500,000 word English corpus, the top two such pairs of 1:1 relationships are (1) 46 stems related by a final *d/s* alternation, including *intrud/intrus*, *apprendend/apprenhens*, *provid/provis, suspend/suspens*, and *elud/elus*, and (2) 43 stems related by a final *i/y* alternation, including *reli/rely, ordinari/ordinary, decri/decry, suppli/supply,* and *accompani/accompany*. This approach can quickly locate patterns of allomorphy that are well-known in the European languages (e.g., alternation between *a* and *ä* in German, between *o* and *ue* in Spanish, between *c* and *ç* in French). However, we do not currently have a satisfactory means of segregating these meaningful cases, such as these, from the (typically less frequent and) spurious cases of stems whose forms are parallel but ultimately not related.

8. Results

On the whole, the inclusion of the strategies described in the preceding sections leads to very good, but by no means perfect, results. In this section we shall review some of these results qualitatively, some quantitatively, and discuss briefly the origin of the incorrect parses.

We obtain the most striking result by looking at the top list of signatures in a language, if we have some familiarity with the language: it is almost as if the textbook patterns have been ripped out and placed in a chart. As these examples suggest, the large morphological patterns identified tend to be quite accurately depicted. To illustrate the results on European languages, we include signatures found from (Table 4) a 500,000 word corpus of English, (Table 5) a 350,000 word corpus of French, (Table 6) *Don Quijote*, which contains 124,716 words of Spanish, (Table 7) a 125,000 word corpus of Latin, (Tables 8 and 9) 100,000 words and 1,000,000 words of Italian. The 500,000 word (token-count) corpus of English (the first part of the Brown Corpus) contains slightly more than 30,000 distinct words.

To illustrate the difference of scale that is observed depending on the size of the corpus, observe the difference in the signatures obtained in Italian between a corpus of 100,000 words (Table 8) and a corpus of 1,000,000 words (Table 9). When one sees the rich inflectional pattern emerging, as with the example of the 10 suffixes on 1st conjugation stems (*a.ando.ano.are.ata.ate.ati.ato.azione.ò),* one cannot but be struck by the grammatical detail that is emerging from the study of a larger corpus.[26]

Turning to French, we may briefly inspect the top 10 signatures that we find in a 350,000 word corpus in Table 5. It is instructive to consider the signature *a.aient.ait.ant.e.ent.er.es.èrent.é.ée.és,* which is ranked 9th among signatures. It contains a

large part of the suffixal pattern from the most common regular conjugation, the 1st conjugation.

Within the scope of the effort covered by this project, the large-scale generalizations extracted about these languages appear to be quite accurate (leaving for further discussion below the questions of how to link related signatures and related stems). It is equally important to take a finer-grain look at the results and quantify them. To do this, we have selected from the English and the French analyses a set of 1000 consecutive words in the alphabetical list of words from the corpus and divided them into distinct sets regarding the analysis provided by the present algorithm. See Table 10 Results (English) and Table 11 Results (French).

The first category of analyses, labeled "Good," is self-explanatory in the case of most words (e.g., *proceed, proceeded, proceeding, proceeds),* and many of the errors are equally easy to identify by eye (*abide* with no analysis, next to *abid-e* and *abid-ing*, or *Abn-er*). Quite honestly, I was surprised how many words there were in which it was difficult to say what the correct analysis was. For example, consider the pair *aboli-tion* and *abol-ish*. The words are clearly related, and *abolition* clearly has a suffix; but does it have the suffix *–ion*, *-tion*, or *–ition*, and does *abolish* have the suffix *–ish*, or *–sh*? It is hard to say. In a case of this sort, my policy for assigning success or failure has been influenced by two criteria. The first is that analyses are better insofar as they explicitly relate words that are appropriately parallel in semantics, as in the *abolish/abolition* case; thus I would give credit to either the analysis *aboli-tion/aboli-sh* or the analysis *abol-ition/abol-ish*. The second criterion is a bit more subtle. Consider the pair of words *alumnus* and *alumni*. Should these be morphologically analyzed in a corpus of English, or rather, should failure to analyze them be penalized for this morphology algorithm?

(Compare in like manner *alibi or allegretti*; do these English words contain suffixes?). My principle has been that if I would have given the system additional credit by virtue of discovering that relationship, I have penalized it if it did not discover it; that is a relatively harsh criterion to apply, to be sure. Should proper names be morphologically analyzed? The answer is often unclear. In the 500,000 word English corpus, we encounter *Alex* and *Alexis*, and the latter is analyzed as *alex-is*. I have scored this as correct, much as I have scored as correct the analyses of *Alexand-er* and *Alexand-re*. On the other hand, the failure to analyze *Alexeyeva* despite the presence of *Alex* and *Alexei* does not seem to me to be an error, while the analysis *Anab-el* has been scored as a error, but *John-son* (and a bit less obviously *Wat-son)* have not been treated as errors.[27] Difficult to classify, too, is the treatment of words such as *abet*/*abetted/abetting.* The present algorithm selects the uniform stem *abet* in that case, assigning the signature *NULL.ted.ting*. Ultimately what we would like to have is a means of indicating that the doubled *t* is predictable, and that the correct signature is *NULL.ed.ing*. At present this is not implemented, and I have chosen to mark this as correct, on the grounds that it is more important to identify words with the same stem than to identify the (in some sense) correct signature. Still, unclear cases remain: for example, consider the words *accompani-ed / accompani-ment / accompani - st.* The word *accompany* does not appear as such, but the stem *accompany* is identified in the word *accompany-ing*. The analysis *accompani-st* fails to identify the suffix *–ist*, but it will successfully identify the stem as being the same as the one found in *accompanied* and *accompaniment*, which it would not have done if it had associated the *-i-* with the suffix. I have, in any event, marked this analysis as wrong, but without much conviction behind the decision. Similarly, the analysis of French putative stem *embelli* with suffixes *e/rent/t* passes the low test of treating related words with the same stem, but

I have counted it as in error, on the grounds that the analysis is unquestionably one letter off from the correct, traditional analysis of 2nd conjugation verbs. This points to a more general issue regarding French morphology, which is more complex than that of English. The infinitive *écrire* 'to write' would ideally be analyzed as a stem *écr* plus a derivational suffix *i* followed by an infinitival suffix *re*. Since the derivational suffix *i* occurs in all its inflected forms, it is not unreasonable to find an analysis in which the *i* is integrated into the stem itself. This is what the algorithm does, employing the stem *écri* for the words *écri-re* and *écri-t*. *Écrit-* in turn is the stem for *écrite, écrite, écrites, écrits*, and *écriture*. An alternate stem form *écriv-* is used for past tense forms (and the nominalization *écrivain*) with the suffixes *aient, ait, ant, irent, it*. The algorithm does not make explicit the connection between these two stems, as it ideally would.

Thus in the tables below, "Good" indicate the categories of words where the analysis was clearly right, while the incorrect analyses have been broken into several categories. "Wrong analysis" is for bimorphemic words which are analyzed, but incorrectly analyzed, by the algorithm. "Failed to analyze" are the cases of words that are bimorphemic but for which no analysis was provided by the algorithm, and "spurious analysis" are the cases of words which are not morphologically complex but were analyzed as containing a suffix.

For both English and French, correct performance is found in 83% of the words; details are presented in Table 10 and Table 11. For English, these figures correspond to precision of 829/(829+52+83) = 85.9%, and recall of 829/(829+52+36) = 90.4%.

9. Triage

As noted above, the goal of *triage* is to determine how many stems must occur in order for the data to be strong enough to support the existence of a linguistically real

signature. MDL provides a simple but not altogether satisfactory method of achieving this end.

Using MDL for this task amounts to determining whether the total description length decreases when a signature is eliminated by taking all of its words and eliminating their morphological structure, and reanalyzing the words as morphologically simple (i.e., as having no morphological structure). This is how we have implemented it, in any event; one could well imagine a variant under which some or all subparts of the signature which comprised other signatures were made part of those other signatures. For example, the signature *NULL.ine.ly* is motivated just for the stem *just*. Under the former triage criterion, *justine* and *justly* would be treated as unanalyzed words, whereas under the latter, *just* and *justly* would be made members of the (large) *NULL.ly* signature, and *just* and *justine* might additionally be treated as comprising parts of the signature *NULL.ine* along with *bernard, gerald, eng, capitol, elephant, def,* and *sup* (although that would involve permitting a single stem to participate in two distinct signatures).

Our MDL-based measure tests the goodness of a signature by testing each signature σ to see if the analysis is better when that signature is deleted. This deletion entails treating the signature's words as members of the signature of unanalyzed words (which is the largest signature, and hence such signature-pointers are relatively short). Each word-member of the signature, however, now becomes a separate stem, with all of the increase in pointer length that that entails, as well as increase in letter-content for the stem component.

One may draw the following conclusions, I believe, from the straight-forward application of such a measure. On the whole, the effects are quite good, but by no means as close as one would like to a human's decisions in a certain number of cases. In

addition, the effects are significantly influenced by two decisions which we have already discussed: (i) the information associated with each letter, and (ii) the decision as to whether to model suffix frequency based solely on signature-internal frequences, or based on frequency across the entire morphology. (i) The greater the information associated with each letter, the more worthwhile morphology is (because maintaining multiple copies of nearly similar stems becomes increasingly costly and burdensome). (ii) When suffix frequencies (which are used to compute the compressed length of any analyzed word) are based on the frequency of the suffixes in the entire lexicon, rather than conditionally within the signature in question, the loss of a signature entails a hit on the compression of all other words in the lexicon that employed that suffix; hence *triage* is less dramatic under that modeling assumption.

Consider the effect of this computation on the signatures produced from a 500,000 word corpus of English. After the modifications discussed to this point, but before *triage*, there were 603 signatures with two or more stems *and* two or more sufffixes, and there were 1,490 signatures altogether. Application of triage leads to the loss of only 240 signatures. The single-suffix signatures which were eliminated were: *ide; it; rs; he; ton; o; ie*, all of which are spurious. However, a number of signatures which should not have been lost were eliminated, most strikingly: *NULL.ness*, with 51 good analyses, *NULL.ful*, with 18 good analyses, and *NULL.ish* with only 8 analyses. Most of the cases eliminated, however, were indeed spurious. Counting only those signatures that involves suffixes (rather than compounds) and which were in fact correct, the percentage of the words whose analysis was incorrectly eliminated by triage was 21.9% (236 out of 1077 changes). Interestingly, in light of the discussion on results above, one of the signatures that was lost was *i.us* for the Latin plural (based in this particular case on *genii/genius*).

Also eliminated (and this is most regrettable) was *NULL.n't* (*could/had/does/were/would /did*).

Because maximizing correct results is as important as testing the MDL model proposed here, I have also utilized a triage algorithm which departs from the MDL-based optimization in certain cases which I shall identify in a moment. I believe that when the improvements are made that are identified in section 11 below, the purely MDL-based algorithm will be more accurate; that prediction remains to be tested, to be sure. On this account, we discard any signature for which the total number of stem letters is less than 5, and any signature consisting of a single, one-letter suffix; we keep, then, only signatures for which the savings in letter counts is greater than 15 (where *savings in letter counts* is simply the difference between the sum of the length of words spelled out as a monomorphemic word and the sum of the lengths of the stems and the suffixes); 15 is

chosen empirically.

10 Paradigms

As we noted briefly above, the existence of a regular pattern of suffixation with *n* distinct suffixes will generally give rise to a large set of stems displaying all *n* suffixes, but it will also give rise in general to stems displaying most possible combinations of subsets of these suffixes. Thus, if there is a regular paradigm in English consisting of the suffixes *NULL, -s, -ing,* and *-ed,* we expect to find stems appearing with most possible combinations of these suffixes as well. As this case clearly shows, not *all* such predicted subpatterns are merely partially-filled paradigms. Of stems appearing with the signature

NULL.s, some are verbs (such as *occur/occurs*), but the overwhelming majority, of course, are nouns.

In the present version of the algorithm, no effort is made to directly relate signatures to one another, and this has a significant and negative impact on performance, because analyses in which stems are affiliated with high-frequency signatures are more highly valued than those in which they are affiliated with low-frequency signatures; it is thus of capital importance not to underestimate the total frequency of a signature.[28] When two signatures as we have defined them here are collapsed, there are two major effects on the description length: on the one hand, pointers to the merged signature are shorter -- leading to a shorter total description length -- but in general predicted frequencies of the composite words are worse than they were, leading to a poorer description (via increased cross-entropy, we might say). In practice, the collapsing of signatures is rejected by the MDL measure that we have implemented here.

In work in progress, we treat groups of signatures (as defined here) as parts of larger groups, called *paradigms*. A paradigm consisting of the suffixes *NULL.ed.ing.s*, for example, includes all 15 possible combinations of these suffixes. We can in general estimate the number of stems we would expect to appear with zero counts for one or more of the suffixes, given a frequency distribution, such as a multinomial distribution, for the suffixes.[29] In this way, we can establish some reasonable frequencies for the case of stems appearing in a corpus with only a single suffix. It appears at this time that the unavailability of this information is the single most significant cause of inaccuracies in the present algorithm. It is thus of considerable importance to get a handle on such estimates.[30]

11 Remaining issues

A number of practical questions remain at this point. The most important are the following:

1. Identifying related stems (allomorphs). Languages typically have principles at work relating pairs of stems, as in English many stems (like *win*) are related to another stem with a doubled consonant (*winn*, as in *winn-ing*). We have been reasonably successful in identifying such semi-regular morphology, and will report this in a future publication. There is a soft line between the discovery of related stems, on the one hand, and the parsing of a word into several suffixes. For example, in the case mentioned briefly above for French, it is not unreasonable to propose two stems for 'to write' *ecri-* and *écriv-*, each used in distinct forms. It would also be reasonable, in this case, to analyze the latter stem *écriv* as composed of *ecri-* plus a suffix *–v-*, although in this case, there are no additional benefits to be gained from the more fine-grained analysis.

2. Identifying paradigms from signatures. We would like to automatically identify *NULL.ed.ing.* as a subcase of the more general *NULL.ed.ing.s.* This is a difficult task to accomplish well, as English illustrates, for we would like to be able to determine that *NULL.s* is primarily a subcase of *'s.NULL.s.,* and not of (e.g.) *NULL.ed.s.*[31]

3. Determining the relationship between prefixation and suffixation. The system currently assumes that prefixes are to be stripped off the stem that has already been identified by suffix-stripping. In future work, we would

like to see alternative hypotheses regarding the relationship of prefixation and suffixation tested by the MDL criterion.

4. Compounds. In work reported in Goldsmith and Reutter (1998), we have explored the usefulness of the present system for determining the linking elements used in German compounds, but more work remains to be done to identify compounds in general. Here we run straight into the problem of assigning very short strings a lower likelihood of being words than longer strings. That is, it is difficult to avoid positing a certain number of very short stems, as in English *m-* and *an-*, the first because of pairs such as *me* and *my*, the second because of pairs such as *an* and *any*, but these facts should not be taken as strong evidence that *man* is a compound.

5. As noted at the outset, the present algorithm is limited in its ability to discover the morphology of a language in which there are not a sufficient number of words with only one suffix in the corpus. In work in progress we are developing a related algorithm that deals with the more general case. In the more general case, it is even more important to develop a model that deals with the layered relationship among suffixes in a language. The present system does not explicitly deal with these relationships: for example, while it does break up *ments* into *ment* and *s*, it does not explicitly determine which suffixes *s* may attach to, etc. This must be done in a more adequate version.

6. In work in progress, we have added to the capability of the algorithm the ability to posit suffixes which are in part subtractive morphemes. That is, in English, we would like to establish a single signature that combines

NULL.ed.ing.s and e.ed.es.ing (for *jump* and *love*, respectively). We posit an operator "<x>" which deletes a preceding character x, and with the mechanism, we can establish a single signature *NULL.<e>ed.<e>ing.s*, composed of familiar suffixes NULL and s, plus two suffixes *<e>ed* and *<e>ing*, which delete a preceding (stem-final) *e* if one is present.

12 Conclusion

Linguists face at the present time the question as to whether, and to what extent, information theoretic notions will play a major role in our understanding of linguistic theory over the years to come, and the present system perhaps casts a small ray of light in this area. As we have already noted, MDL analysis makes clear what the two areas are in which an analysis can be judged: it can be judged in its ability to deal with the data, as measured by its ability to compress the data, and it can be judged on its complexity as a theory. While the former view is undoubtedly controversial when viewed from the light of main-stream linguistics, it is the prospect of being able to say something about the complexity of a theory that is potentially the most exciting. Even more importantly, to the extent that we can make these notions explicit, we stand a chance of being able to develop an explicit model of language acquisition employing these ideas.

A natural question to ask is whether the algorithm presented here is intended to be understood as a hypothesis regarding the way in which human beings acquire morphology. I have not employed, in the design of this algorithm, a great deal of innate knowledge regarding morphology, but that is for the simple reason that knowledge of how words divide into subpieces is an area of knowledge which no-one would take to be innate in any direct fashion: if *sanity* is parsed as *san + ity* in one language, it may perfectly well be parsed as *sa+ nity* in another language.

That is, while passion may flame disagreements between partisans of Universal Grammar and partisans of statistically-grounded empiricism regarding the task of syntax acquisition, the task which we have studied here is a considerably more humble one which must in some fashion or other be figured out by grunt work by the language learner. It thus allows us a much sharper image of how powerful the tools are likely to be that the language acquirer brings to the task. And *does* the human child perform computations at all like the ones proposed here?

From most practical points of view, nothing hinges on our answer to this question, but it is a question that ultimately we cannot avoid facing. Reformulated a bit, one might pose the question, does the young language learner — who has access not only to the spoken language, but perhaps also to the rudiments of the syntax and to the intended meaning of the words and sentences — does the young learner have access to additional information which simplifies the task of morpheme identification? It is the belief that the answer to this question is *yes* which drives the intuition (if one has this intuition) that an MDL-based analysis of the present sort is an unlikely model of human language acquisition.

But I think that such a belief is very likely mistaken. Knowledge of semantics and even grammar is unlikely to make the problem of morphology discovery significantly easier. In surveying the various approaches to the problem that I have explored (only the best of which have been described here), I do not know of *any* problem (of those which the present algorithm deals with successfully) which would have been solved by having direct access to either syntax or semantics. To the contrary: I have tried to find the simplest algorithm capable of dealing with the facts as we know them. The problem of

determining whether two distinct signatures derive from a single larger paradigm would be simplified with such knowledge, but that is the exception and not the rule.

So in the end, I think that the hypothesis that the child uses an MDL-like analysis has a good deal going for it. In any event, it is far from clear to me how one could use information, either grammatical or contextual, to elucidate the problem of the discovery of morphemes without recourse to notions along the lines of those used in the present algorithm.

Of course, in all likelihood, the task of the present algorithm is not the same as the language learner's task; it seems unlikely that the child *first* determines what the words are in the language (at least, the words as they are defined in traditional orthographic terms) and then infers the morphemes. The more general problem of language acquisition is one that includes the problems of identifying morphemes, of identifying words both morphologically analyzed and non-, of identifying syntactic categories of the words in question, and of inferring the rules guiding the distribution of such syntactic categories. It seems to me that the only manageable kind of approach to dealing with such a complex task is to view it is an optimization problem, of which MDL is one particular style. Chomsky's early conception of generative grammar (Chomsky 1975 [1955], hereinafter *LSLT*) was developed along these lines as well; his notion of an evaluation metric for grammars was equivalent in its essential purpose to the description length of the morphology utilized in the present paper. The primary difference between the *LSLT* approach and the MDL approach is that the *LSLT* conjectured that the grammar of a language could be factored into two parts, one universal and one language-particular; hence, the problem of optimizing the grammatical description of a given corpus (the

child's input) could be achieved by a grammar written in a simpler notation than one which no computational assumptions, i.e., a Turing machine of some sort. The difference between these hypotheses vanishes asymptotically (as Janos Simon has pointed out to me) as the size of the language increases, or to put it another way, strong Chomskian rationalism is indistinguishable from pure empiricism as the information content of the (empiricist) MDL-induced grammar increases in size relative to the information content of UG. Rephrasing that slightly, the significance of Chomskian-style rationalism is greater, the simpler language-particular grammars are, and they are less significant as language-particular grammars as larger, and in the limit, as the size of grammars grows asymptotically, traditional generative grammar is indistinguishable from MDL-style rationalism. We return to this point shortly.

There is a striking point which has remained tacit up till now regarding the treatment of this problem in contemporary linguistic theory. That point is this: the problem addressed in this paper is not mentioned, not defined, and not addressed. The problem of dividing up words into morphemes is generally taken as one that is so trivial and devoid of interest that morphologists, or linguists more generally, simply do not feel obliged to think about the problem.[32] In a very uninteresting sense, the challenge presented by the present paper to current morphological theory is no challenge at all, because morphological theory makes no claims to knowing how to discover morphological analysis; it claims only to know what to do once the morphemes have been identified.

The early generative grammar view, as explored in *LSLT,* posits a grammar of possible grammars, that is, a format in which the rules of the morphology and syntax

must be written, and it establishes the semantics of these rules, which is to say, how they function. This grammar of grammars is called variously Universal Grammar, or Linguistic Theory, and it is generally assumed to be accessible to humans on the basis of an innate endowment, though one need not buy into that assumption to accept the rest of the theory. In *Syntactic Structures* (Chomsky 1957, pp. 51ff.), Chomsky famously argued that the goal of a linguistic theory that produces a grammar automatically, given a corpus as input, is far too demanding a goal. His own theory cannot do that, and he suggests that no-one else has any idea how to accomplish the task. He suggests furthermore that the next weaker position — that of developing a linguistic theory that could determine, given the data and the account (grammar), whether this was the best grammar — was still significantly past our theoretical reach, and he suggests finally that the next weaker position is a not unreasonable one to expect of linguistic theory: that it be able to pass judgment on which of two theories is superior with respect to a given corpus.

That position is, of course, exactly the position taken by the MDL framework, which offers no help in coming up with analyses, but which is excellent at judging the relative merits of two analyses of a single corpus of data. In this paper, we have seen this point throughout, for we have carefully distinguished between heuristics, which propose possible analyses and modifications of analyses, on the one hand, from the MDL measurement which makes the final quantitative judgment call, deciding whether to accept a modification proposed by the heuristics.

On so much, early generative grammar of LSLT and MDL agree. But they disagree with regard to two points, and on these points, MDL makes clearer, more explicit claims, and both claims appear to be strongly supported by the present study. The two points are

these: the generative view is that there is inevitably an idiosyncratic character to Universal Grammar that amounts to a substantive innate capacity, on the grounds (in part) that the task of discovering the correct grammar of a human language, given only the corpus available to the child, is not sufficient to home in on the correct grammar. The research strategy associated with this position is to hypothesize certain compression techniques (generally called *rule formalisms* in generative grammar) which lead to significant reduction in the size of the grammars of a number of natural languages, compared to what would have been possible without them. Sequential rule ordering is one such suggestion discussed at length in *LSLT*.

To reformulate this in a fashion that allows us to make a clearer comparison with MDL, we may formulate early generative grammar in the following way: To select the correct Universal Grammar out of a set of proposed Universal Grammars $\{UG_i\}$, given corpora for a range of human languages, select that UG for which the *sum of the sizes of the grammars* for all of the corpora is the smallest. It does not follow -- it need not be the case -- that the grammar of English (or German, etc.) selected by the winning UG is the shortest one of all the candidate English grammars, but the winning UG is all-round the supplier of the shortest grammars around the world.[33]

MDL could be formulated in those terms, undoubtedly, but it also can be formulated in a language-particular fashion, which is how it has been used in this paper. Generative grammar is inherently universalist; it has no language-particular format, other than to say that the best grammar for a given language is the shortest grammar.

But we know that such a position is untenable, and it is precisely out of that knowledge that MDL was born. The position is untenable because we can always make

an arbitrarily small compression of a given set of data, if we are allowed to make the grammar arbitrarily complex, to match and, potentially, to overfit the data, and it is untenable because generative grammar offers no explicit notion of how well a grammar must match the training data. MDL's insight is that it is possible to make explicit the trade-off between complexity of the analysis and snugness of fit to the data-corpus in question.

The first tool in that computational trade-off is the use of a probabilistic model to compress the data, using stock tools of classical information theory. These notions were rejected as irrelevant by early workers in early generative grammar (Goldsmith in press). Notions of probabilistic grammar due to Solomonoff (Solomonoff 1995) were not integrated into that framework, and the possibility of using them to quantify the goodness of fit of a grammar to a corpus was not exploited.

It seems to me that it is in this context that we can best understand the way in which traditional generative grammar and contemporary probabilistic grammar formalism can be understood as complementing each other. I, at least, take it in that way, and this paper is offered in that spirit.

Appendix on change in description length

Since what we are really interested in computing is not the minimum description length as such, but rather the *difference* between the description length of one model and that of a variant, it is convenient to consider the general form of the difference between two MDL computations. In general, let us say we will compare two analyses S_1 and S_2 for the same corpus, where S_2 typically contains some item(s) that S_1 does not (or they may differ by where they break a string into factors). Let us write out the difference in length between these two analyses, as in (6), calculating the length of S_1 minus the length of S_2. The general formula derived in (6) is not generally of direct computational interest; it serves rather as a template which can be filled in to compute the change in description length occasioned by a particular structural change in the morphology proposed by a particular heuristic. It is rather complex in its most general form, but it simplifies considerably in any specific application. The heuristic determines which of the terms in these formulas take on non-zero values, and what their values are; the overall formula determines whether the change in question improves the description length. In addition, we may regard the formula in (6) as offering us an exact and explicit statement of how a morphology can be improved.

The notation can be considerably simplified if we take some care in advance. Note first that in (6) and below, several items are subscripted to indicate whether they should be counted as in S_1 or S_2. Much of the simplification comes from observing, first, that $\log\frac{M_1}{x} - \log\frac{M_2}{y} = \log\frac{M_1}{M_2} - \log\frac{y}{x}$; second, that this difference is generally computed inside a summation over a set of morphemes, and hence the first term simplifies to a constant times the type-count of the morphemes in the set in question. Indeed, so

prevalent in these calculations is the formula $\log \frac{x_{state\,1}}{x_{state\,2}}$ that the introduction of a new abbreviation considerably simplifies the notation. We use $\Delta(x)$ to denote $\log \frac{[x]_1}{[x]_2}$, where the numerator is a count in S_1, and the denominator a count of the same variable in S_2.[34]

Let us review the terms listed in (6). ΔW is a measure of the change in the number of total words due to the proposed modification (the difference between the S_1 and S_2 analyses) ; an increase in the total number of words results in a very slightly negative value. In the text above, I indicated that we could, by judicious choice of word-count distribution, keep $W_1 = W_2$; I have included the more general case in (6) where the two may be different. ΔW_S and ΔW_C are similar measures in the change of words that are morphologically simple, and morphologically complex, stems respectively. They measure the global effects of the typically small changes brought about by a hypothetical change in morphological model. In the derivation of each formula, we consider the first the case of those morphemes which are found in both S_1 and S_2 (indicated (S_1,S_2)), followed by those found only in S_1 ($S_1,\sim S_2$), and then those only found in S_2 ($\sim S_1,S_2$). Recall that angle brackets are used to indicate the *type* count of a set, the number of typographically distinct members of a set.

In (5ii), we derive a formula for the change in length of the suffix component of the morphology. Observe the final formulation, in which the first two terms involve suffixes present in both S_1 and S_2, while term 3 involves suffixes present only in S_1 and term 4 involves suffixes present only in S_2. This format will appear in all of the components of this computation. Recall that λ specifies the number of bits per letter.

In (5iii), we derive the corresponding formula for the stem component.

The general form of the computation of the change to the signature component (5iv) is more complicated, and this complexity motivates a little bit more notation to simplify it. First, we can compute the change in the pointers to the signatures, and the information that each signature contains regarding the count of its stems and suffixes as in (5.iv.a). But the heart of the matter is the treatment of the stems and suffixes within the signatures, given in (5.iv.b-d).

Bear in mind, first of all, that each signature consists of a list of pointers to stems, and a list of pointers to suffixes. The treatment of suffixes is given in (5.iv.d), and is relatively straightforward, but the treatment of stems (5.iv.c) is a bit more complex. Recall that all items on the stem list will be pointed to by exactly one stem pointer, located in some particular signature. All stem pointers in a signature which point to stems on the suffix list are directly described a "simple" word, a notion we have already encountered: a word whose stem is not further analyzable. But other words may be *complex*, that is, may contain a stem whose pointer is to an analyzable word, and hence the stem's representation consists of a pointer-triple: a pointer to a signature, a stem within the signature, and a suffix within the signature. And each stem-pointer is preceded by a flag indicating which type of stem it is.

We thus have three things whose difference in the two states, S_1 and S_2, we wish to compute. The difference of the lengths of the flag is given in (5.iv.c.i). In (5.iv.c.ii), we need change in the total length of the pointers to the stems, and this has actually already been computed already, during the computation of (5ii).[35] Finally (5.iv.c.iii), the set of pointers from certain stem-positions to words consists of pointers to all of the words that we have already labeled as being in W_C, and we can compute the length of these pointers by adding counts to these words; the length of the pointers to these words needs to be

computed anyway in the determining the compressed length of the corpus. This completes the computations needed to compare two states of the morphology.

In addition, we must compute the difference in the compressed length of the corpus in the two states, and this is given in (6v).

(6)

(i) Differences in description length due to organizational information:

Δ<Suffixes> + Δ<Stems> + Δ <Signatures>

(ii) Difference in description length for suffix component of the morphology:

$$\Delta W * < Suffixes >_{(1,2)} \quad - \quad \underset{f \in Suffixes_{(1,2)}}{\Delta f} \quad + \underset{f \in Suffixes_{(1,\sim 2)}}{} \left[\log \frac{[W]_1}{[f]} + \lambda * |f| \right]$$

$$- \underset{f \in Suffixes_{(\sim 1,2)}}{} \left[\log \frac{[W]_2}{[f]} + \lambda * |f| \right]$$

(iii) Difference in description length for stem component of the morphology:

$$\Delta W * \langle Stems \rangle_{(1,2)} \quad - \quad \underset{t \in Stems_{(1,2)}}{\Delta t} + \underset{t \in Stems_{(1,\sim 2)}}{} \left[\log \frac{[W]_1}{[t]} + \lambda * |t| \right]$$

$$- \underset{t \in Stems_{(\sim 1,2)}}{} \left[\log \frac{[W]_2}{[t]} + \lambda * |t| \right]$$

(iv) Difference in description length for the signature component of the morphology:

(a) Change in size of list of pointers to the signatures,

$$\Delta W * \left\langle Signatures_{(1,2)} \right\rangle \quad - \sum_{\sigma \in Signatures_{(1,2)}} \Delta\sigma$$
$$+ \sum_{\sigma \in Signatures_{(1,\sim 2)}} \log \frac{[W]_1}{[\sigma]} \quad - \sum_{\sigma \in Signatures_{(\sim 1,2)}} \log \frac{[W]_2}{[\sigma]}$$

(b) Change in counts of stems and suffixes within each signature, summed over all signatures:

$$\sum_{\sigma \in Signatures_{(1,2)}} \left[\Delta\left\langle stems(\sigma)\right\rangle + \Delta\left\langle suffixes(\sigma)\right\rangle\right]$$
$$- \sum_{\sigma \in Signatures_{(1,\sim 2)}} \left[\log\left\langle stems(\sigma)\right\rangle + \log\left\langle suffixes(\sigma)\right\rangle\right]$$
$$+ \sum_{\sigma \in Signatures_{(\sim 1,2)}} \left[\log\left\langle stems(\sigma)\right\rangle + \log\left\langle suffixes(\sigma)\right\rangle\right]$$

(c) Change in the lengths of the stem pointers within the signatures = (c.i) + (c.ii) + (c.iii), as follows:

(c.i) Change in total length of flags for each stem indicating whether simple or complex:

$$\left\langle W_{SIMPLE} \right\rangle_{1,2} * (\Delta W - \Delta W_{SIMPLE}) + \left\langle W_{COMPLEX} \right\rangle_{1,2} * (\Delta W - \Delta W_{COMPLEX})$$
$$+ \left\langle W_{SIMPLE} \right\rangle_{1,\sim 2} * \log \frac{[W]_1}{[W_{SIMPLE}]_1}$$
$$- \left\langle W_{SIMPLE} \right\rangle_{\sim 1,2} * \log \frac{[W]_2}{[W_{SIMPLE}]_2}$$
$$+ \left\langle W_{COMPLEX} \right\rangle_{1,\sim 2} * \log \frac{[W]_1}{[W_{COMPLEX}]_1}$$
$$- \left\langle W_{COMPLEX} \right\rangle_{\sim 1,2} * \log \frac{[W]_2}{[W_{COMPLEX}]_2}$$

(c.ii) Set of simple stems, change of pointers to stems:

(c.iii) Change in length of pointers to complex stems from within signatures:

$$\Delta W * \left\langle W_{Complex} \right\rangle_{(1,2)} + \sum_{w \in W_{Complex}(1,2)} \Delta stem(w)$$
$$+ \sum_{w \in W_{Complex}(1,\sim 2)} \log \frac{[W]_1}{[stem(w)]_1} - \sum_{w \in W_{Complex}(\sim 1,2)} \log \frac{[W]_2}{[stem(w)]_2}$$
$$+ \sum_{w \in W_{COMPLEX}(1,2)} \Delta\sigma(w) - \Delta[suff(w)\ in\ \sigma(w)]$$
$$+ \sum_{w \in W_{COMPLEX}(1,\sim 2)} \log \frac{[\sigma(w)]}{[suff(w)\ in\ \sigma(w)]}$$
$$- \sum_{w \in W_{COMPLEX}(\sim 1,2)} \log \frac{[\sigma(w)]}{[suff(w)\ in\ \sigma(w)]}$$
$$\Delta W * \left\langle W_{COMPLEX} \right\rangle_{(1,2)} + \sum_{w \in W_{COMPLEX}(1,2)} \Delta stem(w)$$
$$+ \sum_{w \in W_{COMPLEX}(1,\sim 2)} \log \frac{[W]_1}{[stem(w)]_1} - \sum_{w \in W_{COMPLEX}(\sim 1,2)} \log \frac{[W]_2}{[stem(w)]_2}$$
$$+ \sum_{w \in W_{COMPLEX}} \Delta\sigma(w) - \Delta[suff(w)\ in\ \sigma(w)]$$
$$+ \sum_{w \in W_{COMPLEX}(1,\sim 2)} \log \frac{[\sigma(w)]}{[suff(w)\ in\ \sigma(w)]}$$
$$- \sum_{w \in W_{COMPLEX}(\sim 1,2)} \log \frac{[\sigma(w)]}{[suff(w)\ in\ \sigma(w)]}$$

(d) Change in size of suffix information in signatures:

$$\sum_{\sigma \in Signatures_{(1,2)}} \left[\Delta\sigma * <\sigma> - \sum_{f \in \sigma} \Delta f \right]$$
$$+ \sum_{\sigma \in Signatures_{(1,\sim 2)}} \sum_{f \in \sigma} \log \frac{[\sigma]}{[f\ in\ \sigma]}$$
$$- \sum_{\sigma \in Signatures_{(\sim 1,2)}} \sum_{f \in \sigma} \log \frac{[\sigma]}{[f\ in\ \sigma]}$$

(v) Change in compressed length of corpus

$$
\begin{aligned}
&[W]_{raw}\Delta W - \\
&\sum_{w\in W_{A(1,2)}} [w]_{raw}[\,\Delta stem(w) + \Delta[suffix(w)\cap\sigma(w)] - \Delta\sigma(w)\,] \\
&+ \sum_{w\in W_{UN(1,2)}} [w]_{raw}\Delta w \\
&+ \sum_{w\in W_{A(1,\sim 2)}} [w]_{raw}\log\frac{[stem(w)]_1 * [suffix(w)_1\cap\sigma(w)_1]}{[\sigma(w)]_1[w]_2} \\
&- \sum_{w\in W_{A(\sim 1,2)}} [w]_{raw}\log\frac{[stem(w)]_2 * [suffix(w)_2\cap\sigma(w)_2]}{[\sigma(w)]_2[w]_1}
\end{aligned}
$$

Bibliography

Altmann, Gabriel and Werner Lehfeldt. 1980. *Einführung in die Quantitative Phonologie*. Quantitative Linguistics vol. 7. Bochum: Studienverlag Dr. N. Brockmeyer.

Andreev, N. D. (editor) 1965. Statistiko-kombinatornoe modelirovanie iazykov. Moscow and Leningrad: Nauka.

Baroni, Marco (2000) Presentation at annual meeting of the Linguistics Society of America, Chicago, IL. January 2000.

Bell, Timothy C, John G. Cleary, and Ian H. Witten. 1990. *Text Compression*. Englewood Cliffs: Prentice Hall.

Brent, Michael R. 1993. Minimal generative models: A middle ground between neurons and triggers. In *Proceedings of the 15th Annual Conference of the Cognitive Science Society*, pages 28-36. Hillsdale, NJ: Lawrence Erlbaum Associates.

Brent, Michael R. 1999. Speech segmentation and word discovery: a computational perspective. *Trends in Cognitive Science* 3(8): 294-301.

Charniak, Eugene. 1993. *Statistical Language Learning*. Cambridge: MIT Press.

Chomsky, Noam. 1957. *Syntactic Structures.* The Hague: Mouton.

Chomskey, Noam. 1975 [1955]. *The Logical Structure of Linguistic Theory.* New York.

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, B*, 39(1):1-38.

Dobrin, Lise. 1999. *Phonological Form, Morphological Class, and Syntactic Gender: The Noun Class Systems of Papua New Guinea Arapeshan*. PhD dissertation, Department of Linguistics, University of Chicago.

Dzeroski, S. and T. Erjavec. 1997. Induction of Slovene nominal paradigms, in Nada Lavrac, Saso Dzeroski (eds.): *Inductive Logic Programming, 7th International Workshop*, ILP-97, Prague, Czech Republic, September 17-20, 1997, Proceedings. *Lecture Notes in Computer Science*, Vol. 1297, Springer, 1997. Pp. 17-20.

de Marcken, Carl. 1995. *Unsupervised Language Acquisition*. PhD dissertation, MIT.

Flenner, Gudrun. 1994. Ein quantitatives Morphsegmentierungssystem für spanische Wortformen. In Klenk, Ursula (ed.) *Computatio Linguae II*, p. 31-62. Stuttgart: Steiner Verlag.

Flenner, Gudrun. 1995. Quantitative Morphsegmentierung im Spanischen auf phonologisher Basis. *Sprache und Datenverarbeitung* 19(2) 63-79.

Goldsmith, John. 1990. *Autosegmental and Metrical Phonology*. Oxford: Basil Blackwell.

Goldsmith, John. 1995. Introduction to *The Handbook of Phonological Theory*. Oxford: Basil Blackwell.

Goldsmith, John. in press. Entropy and information theory in phonology in the 20th century. To appear in *Folia Linguistica*.

Goldsmith, John and Tom Reutter. 1998. Automatic collection and analysis of German compounds. In *The Computational Treatment of Nominals: Proceedings of the Workshop* COLING-ACL '98. Montreal. Edited by Frederica Busa, Inderjeet Mani and Patrick Saint-Dizier. Pp. 61-69.

Greenberg, Joseph. 1957. *Essays in Linguistics.* Chicago: University of Chicago Press.

Hafer, M. A., & Weiss, S. F. (1974). Word segmentation by letter successor varieties. *Information Storage and Retrieval, 10,* 371-385.

Harris, Zellig. 1955. From phoneme to morpheme. *Language* 31: 190-222, reprinted in Harris 1970.

Harris, Zellig. 1967. Morpheme boundaries within words: report on a computer test. *Transformationas and Discourse Analysis Papers*, 73. Reprinted in Harris 1970.

Harris, Zellig. 1970. *Papers in Structural and Transformational Linguistics*. Dordrecht: D. Reidel.

Janssen, Axel. 1992. Segmentierung französischer Wortformen in Morphe ohne Verwendung eines Lexikons. In Klenk, Ursula (ed.), *Computatio Linguae*, p. 74-95.Stuttgart: Steiner Verlag.

Karttunen, Lauri. 1993. Finite State Constraints. In John Goldsmith (ed.), *The Last Phonological Rule*, pp. 173-194. Chicago: University of Chicago Press.

Kazakov, Dimitar. 1997. Unsupervised Learning of Naïve Morphology with Genetic Algorithms. In W. Daelemans, A. van den Bosch, and A. Weijtera, eds., *Workshop Notes of the ECML/Mlnet Workshop on Empirical Learning of Natural Language Processing Tasks*, April 26, 1997, Prague.

Klenk, Ursula. 1992. Verfahren morphologischer Segmentierung und die Wortstruktur im Spanischen. In Klenk, Ursula (ed.), *Computatio Linguae*, p. Stuttgart: Steiner Verlag.

Koch, Sabine, Andreas Küstner, and Barbara Rüdiger. 1989. Deutsche Wortformensegmentierung ohne Lexicon. *Sprache und Datenverarbeitung* 13/1 35-44.

Koskenniemi, Kimmo. 1983. Two-level Morphology: A General Computational Model for Word-form Recognition and Production. Publication no. 11, Department of General Linguistics. Helsinki: University of Helsinki.

Langer, Hagen. 1991. *Ein automatisches Morphsegmentierungsverfahren für deutsche Wortformen.* Unpublished doctoral dissertation, Georg-August-Universität (Göttingen).

Nida, Eugene. 1948. The identification of morphemes. In Martin Joo, ed., *Readings in Linguistics I.* Chicago: The University of Chicago Press.

Nida, Eugene. 1949. *Morphology: Te Dscriptive Aalysis of Words.* Ann Arbor: The University of Michigan.

Pacak, M. G. and A. W. Pratt. 1976. Automated morphosyntactic analysis of medical language. *Information Processing and Management, 12*, 71-76.

Radhakrishnan, T. (1978). Selection of prefix and postfix word fragments for data compression. *Information Processing and Management, 14*(2), 97-106.

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

Solomonoff, R. (1995). The discovery of algorithmic probability: A guide for the programming of true creativity. In P. Vitányi (ed.), *Computational Learning Theory*. Berlin: Spring Verlag.

Wothke, K., & Schmidt, R. (1992). A Morphological Segmentation Procedure for German. *Sprache und Datenverarbeitung, 16*(1), 15-28.

Table 2 Top 81 signatures from *Tom Sawyer*

Rank	Signature	# stems
1	NULL.ed.ing	69
2	e.ed.ing	35
3	NULL.s	253
4	NULL.ed.s	30
5	NULL.ed.ing.s	14
6	's.NULL.s	23
7	NULL.ly	105
8	NULL.ing.s	18
9	NULL.ed	89
10	NULL.ing	77
11	ed.ing	74
12	's.NULL	65
13	e.ed	44
14	e.es	42
15	NULL.er.est.ly	5
16	e.es.ing	7
17	NULL.ly.ness	7
18	NULL.ness	20
19	e.ing	18
20	NULL.ly.s	6
21	NULL.y	17
22	NULL.er	16
23	e.ed.es.ing	4
24	NULL.ed.er.ing	4
25	NULL.es	16
26	NULL.ful	13
27	NULL.e	13
28	ed.s	13
29	e.ed.es	5
30	ed.es.ing	5
31	NULL.ed.ly	5
32	NULL.n't	10
33	NULL.t	10
34	'll.'s.NULL	4
35	ed.ing.ings	4
36	NULL.s.y	4
37	NULL.ed.er	4
38	NULL.ed.ment	4
39	NULL.ful.s	4
40	NULL.ed.ing.ings	3
41	ted.tion	9
42	's.NULL.ly.s	3
43	NULL.ed.s.y	3
44	t.tion	8
45	NULL.less	8
46	e.er	8
47	NULL.ment	8
48	le.ly	8
49	NULL.ted	7
50	NULL.tion	7
51	l.t	7
52	ence.ent	6
53	NULL.ity	6
54	NULL.est.ly	3
55	ed.er.ing	3
56	NULL.ed.ive	3
57	NULL.led.s	3
58	NULL.er.ly	3
59	NULL.ily.y	3
60	NULL.n.s	3
61	NULL.ed.ings	3
62	NULL.ed.es	3
63	e.en.ing	3
64	NULL.ly.st	3
65	NULL.s.ter	3
66	NULL.ed.ing.ings.s	2
67	NULL.i.ii.v.x	2
68	NULL.ed.ful.ing.s	2
69	ious.y	5
70	NULL.en	5
71	ation.ed	5
72	NULL.able	5
73	ed.er	5
74	nce.nt	5
75	NULL.an	4
76	NULL.ed.ing.y	2
77	NULL.en.ing.s	2
78	NULL.ed.ful.ing	2
79	NULL.st	4
80	e.ion	4
81	NULL.al.ed.s	2

Table 3 Suffixes from Tom Sawyer

Suffix	Number of stems	Occurrences	Remarks
s	1178	3290	
ed	1760	2447	
ing	1322	1685	
er	448	1531	
e	116	1174	
ly	566	857	
's	298	809	
d	286	738	
y	174	625	
n	52	472	
on	96	346	Spurious (bent-on, rivers-on): triage issue.
es	324	329	
t	52	291	
st	110	270	Signature NULL.ly.st, for stems such as safe-
en	116	229	behold, deaf, weak, sunk, etc.
le	142	176	Error: analyzed le.ly for e.y (stems such as feeb-, audib-, simp-).
al	150	167	
n't	32	164	
nce	78	151	Signature nce.nt, for stems fragr-, dista-, indiffere-
ent	102	148	Spurious: triage problem (pot-ent)
tion	134	135	
r	36	135	
ter	52	132	triage problem
k	22	129	triage problem
ful	82	125	
ion	126	124	
'll	34	117	
an	44	117	triage problem
ness	110	116	
nt	74	111	see above
ted	148	84	chat-ted, fit-ted, submit-ted, etc.
est	80	75	
ity	74	71	
ous	88	68	
ard	42	65	drunk-ard
able	70	64	
ious	50	57	
less	64	51	
ment	46	48	
id	38	48	id.or for stems horr-, splend-, liqu-
ure	44	47	
ive	62	44	

ty	40	39	novel, uncertain, six, proper
ence	30	38	
ily	30	31	
ward	14	21	
ation	26	21	
led	16	18	triage problem
'd	12	18	
ry	10	17	error: stems such as glo- with signature rious.ry
rious	10	15	error: stems such as glo- with signature rious.ry
rs	6	12	error: r should be in stem
ned	18	11	awake-ned, white-ned, thin-ned
ning	6	11	begin-ning, run-ning
age	14	9	
h	10	7	triage problem
te	8	6	should be -ate (e.g., punctua-te)
ant	8	4	triumph-ant, expect-ant
r's	6	4	error
ance	8	4	

Table 4 Top 10 signatures, English 500,000 word corpus

1. NULL.ed.ing.s
accent
add
administer
afford
alert
amount
appeal
assault
attempt

2. 's.NULL.s
adolescent
afternoon
airline
ambassador
amendment
announcer
architect
assessor
association

3. NULL.ed.er.ing.s
attack
back
bath
boil
borrow
charm
condition
demand
down
flow

4. NULL.s
abberation
abolitionist
abortion
absence
abstractionist
abutment
accolade
accommodation
accomodation

5. e.ed.es.ing
achiev
assum
brac
chang
charg
compris
conced
conclud
decid
describ

6. e.ed.er.es.ing
advertis
announc
bak
challeng
consum
enforc
gaz
glaz
invad
liv
pac

7. NULL.ed.ing
applaud
arrest
astound
blast
bless
bloom
boast
bolster
broaden
cater

8. NULL.er.ing.s
blow
bomb
broadcast
deal
draw
drink
dwell
farm
feed
feel

9. NULL.d.s
abbreviate
accommodate
aggravate
apprentice
arcade
balance
barbecue
bruise
catalogue
costume

10. NULL.ed.s
acclaim
beckon
benefit
blend
blister
bogey
bother
breakfast
buffet
burden

Table 5 Top 10 signatures, French 350,000 word corpus

1. NULL.e.es.s
abondant
abstrait
adjacent
approprié
atteint
bantou
bleu
brillant
byzantin

2. NULL.s
abandonnée
abbaye
abdication
abdominale
abélienne
aberration
abolitionniste
abordée
abrasif
abréviation

3. NULL.ment.s
administrative
agressive
anatomique
ancienne
annuelle
automatique
biologique
chimique
classique

4. NULL.e.es
acquis
aéropostal
afghan
albanais
allongé
anglais
appelé
arrondi
bavarois
carthaginois

5. NULL.e.s
adhérent
adolescent
affilié
aîné
assigné
assistant
bovin
cinglant
colorant

6. NULL.ne.s
abélien
acheuléen
alsacien
amérindien
ancien
anglo-saxon
araméen
aristotélicien
athénien

7. NULL.e
accueillant
acharné
admis
adsorbant
albigeois
alicant
aliénant
alléchant
amarant
ambiant

8. NULL.es.s
antioxydant
bassin
civil
craint
cristallin
cutané
descendant
doté
émulsifiant
ennemi

9. a.aient.ait.ant.e.ent.er.es.èrent.é.ée.és
contrôl
jou
laiss
rest

10. NULL.es
adopté
âgé
allié
annulé
apparenté
apprécié
armé
assiégé
associé
attaché

Table 6 Top 10 signatures, Spanish 130,000 word corpus

1. a.as.o.os	**4. NULL.n**	**7. NULL.a.as.o.os**
abiert	abría	algun
aficionad	abriría	buen
ajen	acabase	es
amig	acabe	mí
antigu	acaece	primer
compuest	acertaba	un
cortesan	acometía	**8. NULL.es**
cubiert	acompañaba	ángel
cuy	acordaba	animal
delicad	aguardaba	árbol
2. NULL.s	**5. NULL.n.s**	azul
aborrecido	caballero	bachiller
abrasado	cante	belianis
abundante	debía	bien
acaecimiento	dice	buey
accidente	dijere	calidad
achaque	duerme	cardenal
acompañado	entiende	**9. da.do.r**
acontecimiento	fuerza	amanceba
acosado	hubiera	ata
acostumbrado	miente	averigua
3. a.o.os	**6. a.as.o**	colga
afligid	agradezc	emplea
ánim	anch	feri
asalt	atónit	fingi
caballeriz	confus	heri
desagradecid	conozc	pedi
descubiert	decill	persegui
despiert	dificultos	**10. NULL.le**
dorad	estrech	abrazó
enemig	extrañ	acomodar
flac	fresc	aconsejó
		afligióse
		agradeció
		aguardar
		alegró
		arrojó
		atraer
		besó

Table 7 Top 10 signatures, Latin 125,000 word corpus

1. NULL.que	4. NULL.m	7. NULL.e.m
abierunt	abdia	angustia
acceperunt	abia	baptista
accepit	abira	barachia
accinctus	abra	bethania
accipient	adonira	blasphemia
addidit	adsistente	causa
adiuvit	adulescente	conscientia
adoravit	adulescentia	corona
adplicabis	adustione	ignorantia
adprehendens	aetate	lorica
2. NULL.m.s	5. i.is.o.orum.os.um.us	8. a.ae.am.as.i.is.o.orum.os.um.us
acie	angel	ann
aquaeductu	cubit	magn
byssina	discipul	mult
civitate	iust	univers
coetu	ocul	9. NULL.e.m.s
die	popul	azaria
ezechia	6. e.em.es.i.ibus.is.um	banaia
facultate	fratr	esaia
fide	greg	iosia
fimbria	homin	iuda
3. a.ae.am.as.is	reg	lucusta
ancill	vic	massa
aqu	voc	matthathia
lucern		pluvia
parabol		sagitta
plag		10. i.o.um
puell		brachi
stell		carmel
synagog		cenacul
tabul		damn
tunic		evangeli
		hysop
		lectul
		liban
		offici
		ole

Table 8 Top 10 signatures, Italian 100,000 word corpus

Rank	Signature	Number of stems participating in this signature
1	a.e.i.o	55
2	ica.iche.ici.ico	17
3	a.i.o	33
4	e.i	221
5	i.o	164
6	e.i.o	24
7	a.e.o	23
8	a.e.i	23
9	a.e	131
10	NULL.o	71
11	e.i.ità	14

Table 9 Top 10 signatures, Italian 1,000,000 word corpus

Rank	Signature	Number of stems participating in this signature
1	.a.e.i.o.	136
2	.ica.iche.ici.ico.	43
3	.a.i.o.	114
4	.ia.ica.iche.ici.ico.ie.	13
5	.a.ando.ano.are.ata.ate .ati.ato.azione.ò.	7
6	.e.i.	583
7	.a.e.i.	47
8	.i.o.	383
9	.a.e.o.	32
10	.a.e.	236

Table 10 Results (English)

Category	Count	Percent
Good	829	82.9%
Wrong analysis	52	5.2%
Failed to analyze	36	3.6%
Spurious analysis	83	8.3%

Table 11 Results (French)

Category	Count	Percent
Good	833	83.3%
Wrong analysis	61	6.1%
Failed to analyze	42	4.2%
Spurious analysis	64	6.4%

Figure 1

A simple sample morphology:

Signatures:

$$(sig1)\left\{\begin{array}{c} SimpleStem:1 \\ SimpleStem:2 \\ SimpleStem:3 \\ ComplexStem:sig2:1st\ stem+3rd\ suffix \end{array}\right\}\left\{\begin{array}{c} ptr(NULL) \\ ptr(s) \end{array}\right\}$$

$$(sig2)\{SimpleStem:7\}\left\{\begin{array}{l} ptr(e) \\ ptr(s) \\ ptr(ing) \end{array}\right\}$$

$$(sig3)\left\{\begin{array}{l} SimpleStem:5 \\ SimpleStem:6 \\ SimpleStem:8 \end{array}\right\}\left\{\begin{array}{c} ptr(NULL) \\ ptr(ed) \\ ptr(ing) \\ ptr(s) \end{array}\right\}$$

$$(sig4)\left\{\begin{array}{l} SimpleStem:4 \\ Simplestem:11 \end{array}\right\}$$

Affixes:

$$\left\{\begin{array}{c} 1:NULL \\ 2:ed \\ 3:ing \\ 4:s \end{array}\right\}$$

Stems:

{1:cat,

2:dog,

3:hat,

4:John,

5:jump,

6:laugh,

7:sav,

8:the,

9:walk}

This covers the words: *cat, cats, dog, dogs, hat, hats, save, saves, saving, savings, jump, jumped, jumping, jumps, laugh, laughed, laughing, laughs, walk, walked, walking, walks, the, John.* Note that the morphology includes the stem *saving*, but the stem list does not.

[1] Some of the work reported here was done while I was a visitor at Microsoft Research in the winter of 1998, and I am grateful for the support I received there. This work was also supported in part by a grant from the Argonne National Laboratory-Univeristy of Chicago consortium, which I thank for its support. I am also grateful for helpful discussion of this material with a number of people, including Carl de Marcken, Jason Eisner, Zhiyi Chi, Derrick Higgins, Janos Simon, Svetlana Soglasnova, Hisami Suzuki, and Jessie Pinkham. As noted below, I owe a great deal to the remarkable work reported in de Marcken's dissertation, without which I would not have undertaken the work described here. I am grateful as well to several anonymous reviewers for their considerable improvements to the content of this paper.

[2] In addition, one would like a statement of general rules of allomorphy as well; for example, a statement that the stems *hit* and *hitt* (as in *hits* and *hitting*, respectively) are forms of the same linguistic stem. In an earlier version of this paper, we discussed a practical method for achieving this. The work is currently under considerable revision, and we will leave the reporting on this aspect of the problem to a later paper.

[3] The executable is available at http://humanities.uchicago.edu/faculty/goldsmith/Linguistica2000, along with instructions for use. The functions described in this paper can be incrementally applied to a corpus by the user of Linguistica.

[4]For much the same reason I have let the C and C++ conventions dictate where word-breaks occur, and what counts as punctuation. These results are generally reasonable, but occasionally clearly wrong; the case of the apostrophe is the classic

example. In French, the apostrophe is virtually always a marker of a syntactic word-break (*c'est*, *qu'avez*), as are most hyphens (*as-tu* [*mangé*], etc.). In English, of course, an apostrophe may mark a syntactic word break (*he's gone*), but it often does not (*doesn't, don't*). I have generally let the C/C++ decisions as to how to treat text govern my results. At this point, the reason for this is to insist (in the extreme) that there be no language-particular information or structure built into the program. Some modification is at times not inappropriate; the determination that certain characters are punctuation or not can be country-particular in certain C++ environments, in the case of characters in the upper 128 range.

[5] Another effort is that attributed to Andreev (1965) and discussed in Altmann and Lehfeldt (1980), esp. pp. 195ff, though their description does not facilitate establishing a comparison with the present approach.

[6] But Harris' method does lend itself to a generalization to more difficult cases of morphological analysis going beyond the scope of the present paper. In work in progress, we have used minimization of mutual information between successive candidate morphemes as part of a heuristic for preferring a morphological analysis in languages with a large number of suffixes per word.

[7] On this score, language will surely vary to some degree. English, for example, tends to employ rules of morphophonology to modify the surface form of morphologically complex words so as to better match the phonological pattern of unanalyzed words. This is discussed at length in Goldsmith (1990, chapter 5).

[8] Baroni (2000) reported success using an MDL-based model in the task of discovering English prefixes. I have not had access to further details of the operation of the system.

[9] I am grateful to Scott Meredith for drawing my attention to this paper.

[10] Brent's description of his algorithm is not detailed enough to satisfy the curiousity of someone like the present writer, who has encountered problems that Brent's approach would seem certain to encounter equally. As we shall see below, the central practical problem to grapple with is the fact that when considering suffixes (or candidate suffixes) consisting of only a single letter (let us say *s*, for example), it is extremely difficult to get a good estimate of how many of the potential occurrences (of word-final *s*'s) are suffixal *s* and how many are not. As we shall suggest towards the end of this paper, the only accurate way to make an estimate is on the basis of a multinomial estimate once larger suffix signatures have been established. Without this, it is difficult *not* to overestimate the frequency of single-letter suffixes, a result that may often, in my experience, deflect the learning algorithm from discovering a correct 2-letter suffix (e.g., the suffix -*al* in French).

[11] It is instructive to think about why this should be so. Consider a word such as *diplomacy*. If we cut the word into the pieces *diplomac* + *y*, its compressed length is –1 * (log freq (diplomac) + log freq (y)), and constrast that value with the corresponding values of two other analyses: -1 * (log freq (diploma) + log freq (cy)), from *diploma* – *cy*, and –1 * (log freq (diplom) + log freq (acy)) from *diplom* - *acy*. Now, the difference in the log frequency of words that begin with *diploma* and those that begin with *diplomac*

is less than 3, while the difference between the difference of the log frequency of words that end in *y* and those that end in *cy* is much greater. In graphical terms, we might note that tries (the data structure) based on forward-spelling have by far the greatest branching structure early in the word, while tries based on backward spelling have the greatest branching structure close to the root node, which is to say at the end of the word. We will discuss this more below, when we will note that one must distinguish the frequency of a letter *qua* letter from its frequency *qua* morpheme.

[12] The n-gram based approach is faster and requires less memory. The results reported below are based in general on the split-all-words strategy.

[13] Experimenting with other functions suggests empirically that the details of our choices for a figure of merit, and the distribution reported in the text, are relatively unimportant. As long as the measurement is capable of ensuring that the cuts are not strongly pushed towards the periphery, the results we get are robust.

[14] Langer 1991 discusses some of the historical origins of this criterion, known in the literature as a Greenburg square (Greenberg 1957). As Langer points out, important antecedents in the literature include Bloomfield's brief discussion (1933, p. 161) as well as Nida (1948, 1949).

[15] It is tempting to pursue the notion that the entropy of a system is proportional to the log of the number of its accessible states. Let us define U(w,n,b) as the number of sets of words possible (the number of vocabularies) in a language with *b* letters, where words are all *w* letters long, and the sets of words all contain *n* words (in this first approximation, we assume all words are distinct. Clearly, log U(w,n,b) =

$\log\binom{b^w}{n} = \log\frac{b^w!}{(b^w-n)!n!} \cong nw\log b + n\log(1/n)$. *Log b* is the information content of a single letter, and hence the first term can be conceived of as the length of the list of words spelled out in letters, while the second term can be thought of as the sum of the length of the pointers to the words.

What is the difference, we may ask, between the following two linguistic systems: in both systems, the alphabet consists of *b* letters, and all words are *w* letters long. The number of words in the language is *N* altogether. In System 1, there is no structure to the words, but in System 2, all words consist of a stem that is *t* letters long, plus a suffix that is f letters long, and t + f = w. There are T stems, and F suffixes, and TF = N. Suppose there is a set of words that can be analyzed under either system. How do the systems differ? Clearly, the second system is more interesting and more informative, and that is because its total universe is more limited than that of System 1.

We wish, that is, to compute $\log\frac{U(w,N,b)}{U(t,T,b)U(f,F,b)}$, where *t* is the length of the stems (which for simplicity's sake we assume is fixed), and *f* the length of the suffixes. That tells us how much smaller the universe gets by having morphology, where w = t+f, and N = T*F.

Using the approximation given above, we find that

$$(i)\log\frac{U(w,N,b)}{U(t,T,b)U(f,F,b)} \approx \log b(wN - tT - fF) + N\log\frac{1}{N} - T\log\frac{1}{T} - F\log\frac{1}{F}.$$

One can directly see that this is the difference of two description length analyses, in which the first term describes the difference of the morphology lengths, and the second

the difference in the compressed corpus lengths. *Log b* is the compressed length of a single letter, while *wN* is the number of letters in the corpus, *tT* the number of letters in the stem list, and *fF* the number of letters in the suffix list. Thus we can see that this equation tells us the following: the model in which words are just sequences of letters is much larger than the model in which there are stems and suffixes, and that differences can be quantitatively measured by the amount in (i). That is, the universe of possibilities is shrunk by the amount in (i) by the assumption (or the knowledge) that there is morphological structure, and that knowledge is the amount given by a description length model: it is the sum of an amount due to model complexity plus an amount due to data compression.

A more complete treatment of this question would explore the consequences of incorporating observed word frequencies. That is, the number of possible vocabularies of size N when words may have integral frequencies (again of length *w* from an alphabet of length *b*) is $\frac{b^{LN}}{\prod_{i=1}^{N}(i!)^{Z(i)}}$,where *Z(i)* is the number of words of frequency *i*. The complexity of such a set is thus approximately $LN\log b - \quad Z(i)i\log i$; in the case of a set of words in which a Zipf distribution were observed (so that Z(i) = $\frac{NZ_0}{i}$), this would be $LN\log b - NZ_0 \quad \log i \cong LN\log b + N^2 Z_0(\log\frac{1}{N})$. The assumption of a Zipf distribution does not appear to hold, however, of the set of stems that are analyzed by the algorithm described in this text.

[16] Wherever I say "letter," the reader may also read "phoneme".

[17] I have simplified notation throughout by writing $\log\frac{a}{b}$ where it is common to write $-\log\frac{1}{\frac{a}{b}}$. Note, also, that encoding of fractional bits is neither a theoretical nor a practical problem; on the notion of arithmetic encoding (as opposed to the more familiar Huffman encoding, for example), see Bell et al (1990).

[18] The reader unfamiliar with this important notion may consult a text such as Charniak 1992 for further discussion.

[19] We refer to *suffixes* here rather than the more general *affix*. Most of the interesting morphology occurs with suffixes, but the same algorithm can be used to identify prefixes in the mirror image case.

[20] We will need to convert *number of letters* as a quantity to something expressed in units of information, so as to be comparable, and such a conversion might reasonably be made by a translation based on the frequency of each letter (so that a letter of frequency *f* was represented by length log (1/f)). A little reflection suggests that this makes sense from a linguistic point of view as well as an information theoretic one. Nonetheless, we leave this aside for present purposes.

[21] One might also model a suffix *f* as having length $\log\frac{[W]}{[f]}$. These two models differ with regard to whether a signature is additionally favored if it uses suffixes that are frequently used in other signatures; the expression used in the body of the text offers no such favoring, and I have experimented with both, without a definitive answer as to

which is preferable. Some evidence actually suggests that the alternative measure suggested in this note is superior. On this account, a signature's existence improves the description not only due to its own direct compression effects, but also indirectly by virtue of improving the compression of every signature with which it shares one or more suffixes, since every signature would contribute towards giving its own suffixes higher counts, and hence making them more compact. This effect becomes noticeable in the computation of *triage*; see below for further discussion.

[22] I discuss a closely related question in the context of underspecification in lexical phonology in Goldsmith (1995), though without explicitly drawing the connection to compression. It is not obvious, for example, that the value of λ should be the same for stems and for affixes. See the discussion of *triage* below.

[23] This computation is rather lengthy, and in actual practice it may be preferable to replace it with far faster approaches to testing a change. One way to speed up the task is to compute the differential of the MDL function, so that we can directly compute the change in description length given some prior changes in the variables that define the morphology that are modified in the hypothetical change being evaluated. This approach is illustrated in Appendix A. The second way to speed up the task is to, again, use heuristics to identify clear cases for which full description length computation is not necessary, and to identify a smaller number of cases where fine description length is appropriate. For example, in the case mentioned in the text, that of determining whether a suffix such as *ments* should always be split into two independently motivated suffixes *ment* and *s*, we can compute the fraction of words ending in *ments* that correspond to

free-standing words ending in *ment*. Empirical observation suggests that ratios over 0.5 should always be split into two suffixes, ratios under 0.3 should not be split, and those in-between must be studied with more care.

[24] This is accomplished by the command *am4* in *Linguistica*.

[25] This is accomplished by the command am5 in *Linguistica*.

[26] Signature 1 is formed from adjectival stems in the fem.sg., fem. pl., masc. pl, and masc. sg. forms; signature 2 is entirely parallel, based on stems ending with the morpheme *-ic/-ich*, where *ich* is used before *i* and *e.* Signature 4 is an extension of Signature 2, including nominalized (sg. and pl.) forms. Signature 5 is the large regular verb inflection pattern (7 such verb stems are identified). Signature 3 is a subset of signature 1, composed of stems accidently not found in the femine plural form (similarly for signatures. Signatures 6 and 8 are primarily masculine nouns, sg., and pl., Signature 10 is feminine nouns, sg., and pl., and the remaining Signatures 7 and 9 are again subsets of the regular adjective pattern of Signature 1.

[27] My inability to determine the correct morphological analysis in a wide range of words that I know perfectly well seems to me to be essentially the same response as has often been observed in the case of speakers of Japanese, Chinese, and Korean when forced to place word-boundaries in e-mail romanizations of their language. Ultimately the quality of a morphological analysis must be measured by how well the algorithm handles the clear cases, how well it displays the relationships between words perceived to be related, and how well it serves as the language model for a stochastic morphology of the language in question.

[28] As long as we keep the total number of words fixed, the global task of minimizing description length can generally be obtained by the local strategy of finding the largest cohort for a group of forms to associate with: if the same data can be analyzed in two ways, with the data forming groups of sizes $\{a^1{}_i\}$ in one case, and $\{a^2{}_i\}$in the other, maximal compression is obtained by choosing the case (k=1, 2) for which $\log(a^k{}_i)$ is the greatest.
$_i$

[29] In particular, consider a paradigm with a set $\{f_i\}$ of suffixes. We may represent a subsignature of that signature as a string of 0s and 1s (a *boolean string* **b**, of the form {0,1}*, abbreviated b_k) indicating whether (or not) the i^{th} suffix is contained in the subsignature. If a stem *t* occurs [t] times, then the probability that it occurs *without* a particular suffix f_i is $(1-prob(f_i))^{[t]}$; the probability that it occurs without all of the suffixes missing from the particular subsignature **b** = {b_k } is $\prod_k (1-b_k)(1-prob(f_i))^{[t]}$; and the probability that the particular subsignature **b** will arise at all is the sum of those values over all of the stems in the signature: $\prod_{t_n \in stems(\sigma)} \prod_k (1-b_k)(1-prob(f_i))^{[t_n]}$ Thus all that is necessary is to estimate the hidden parameters of the frequencies of the individual suffixes in the entire paradigm. See note 27.

[30] There may appear to be a contradiction between this observation about paradigms, and the statement in the preceding paragraph that MDL rejects signature mergers — but there is no contradiction. The rejection of signature mergers is performed (so to speak) by the model which posits that frequencies of suffixes inside a signature are

based only on suffix-frequencies of the stems that appear with exactly the same set of suffixes in the corpus. It is that modeling assumption that needs to be dropped, and replaced by a multinomial-based frequency prediction based on counts over the 2^n-1 signatures belonging to each paradigm of length n.

[31] We noted in the preceding section that we can estimate the likelihood of a subsignature assuming a multinomial distribution. We can in fact do better than was indicated there, in the sense that for a given observed signature σ*, whose suffixes constitute a subset of a larger signature σ, we can compute the likelihood that σ is responsible for the generation of σ*, where $\{\phi_i\}$ are the frequencies (summing to 1.0) associating with each of the suffixes in σ, and $\{c_i\}$ are the *counts* of the corresponding suffixes in the observed signature σ*:

$\binom{[t]}{[c_1],[c_2],...,[c_n]} \prod_{i=1}^{n} \varphi(i)^{c_i} = \frac{[t]!}{[c_1]![c_2]!...[c_n]!} \prod_{i=1}^{n} \varphi_i^{c_i}$ The log likelihood is then

$\log[t]! + \sum_{i=1}^{n} c_i \log \varphi_i - \log[c_i]!$, or approximately $t \log t - \sum c_i \log\left(\frac{c_i}{\varphi_i}\right)$ from Stirling's approximation. If we normalize the c_is to form a distribution (by dividing by [t]) and denote these by d_i , then this can be simply expressed in terms of the Kullback-Leibler distance D(σ*|| σ) :

$$[t]\log[t] - \sum c_i \log\left(\frac{c_i}{\varphi_i}\right) = [t]\log[t] - [t] \sum d_i \log\left(\frac{[t]d_i}{\varphi_i}\right) =$$

$$[t]\log[t] - [t]D(\sigma^* \| \sigma)) - [t] \sum d_i \log([t]) = [t]\log[t] - [t]D(\sigma^* \| \sigma)) - [t]\log[t] = -[t]D(\sigma^* \| \sigma).$$

[32] Though see Dobrin (1999) for a sophisticated look at this problem.

[33] As the discussion in the text may suggest, I am skeptical of the generative position, and I would like to identify what empirical result would confirm the generative position and dissolve my skepticism. The result would be the discovery of two grammars of English, G1 and G2, with the following properties: G1 is inherently simpler than G2, using some appropriate notion of Turing machine program complexity, and yet G2 is the *correct* grammar of English, based on some of the complexity of G2 being the responsibility of linguistic theory, hence “free” in the complexity competition between G1 and G2. That is, the proponent of the generative view must be willing to acknowledge that overall complexity of the grammar of a language may be greater than logically necessary due to evolution’s investment in one particular style of programming language.

[34] We beg the reader's indulgence in recognizing that we prepend the operator Δ immediately to the left of the name of a set to indicate the change in the size of the counts of the set, which is to say, "ΔW″ is shorthand for "Δ([W])", and "Δ<W>" for "Δ(<W>)".

[35] The equivalence between the number computed in (7) and the number needed here is not exactly fortuitous, but it is not an error either. The figure computed in (7) describes an aspect of the complexity of the morphology as a whole, where as the computation described here in the text is what it is because we have made the assumption that each stem occurs in exactly one signature. That assumption is not, strictly speaking, correct in natural language; we could well imagine an analysis which permitted the same stem to appear in several distinction signatures, and in that case, the computation here

would not reduce to (7). But the assumption made in the text is entirely reasonable, and simplifies the construction for us.