Unsupervised Learning of the Morphology of a Natural Language

John Goldsmith*
University of Chicago

This study reports the results of using minimum description length (MDL) analysis to model unsupervised learning of the morphological segmentation of European languages, using corpora ranging in size from 5,000 words to 500,000 words. We develop a set of heuristics that 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.

1. Introduction

This is a report on the present results of a study on unsupervised acquisition of morphology.[1] The central task of morphological analysis is the segmentation of words into the components that form the word by the operation of concatenation. While that view is not free of controversy, it remains the traditional conception of morphology, and the one that we shall employ here.[2] Issues of interface with phonology, traditionally known as morphophonology, and with syntax are not directly addressed.[3] While some of the discussion is relevant to the unrestricted set of languages, some of the assumptions made in the implementation restrict the useful application of the algorithms to languages in which the average number of affixes per word is less than what is found in such languages as Finnish, Hungarian, and Swahili, and we restrict our testing in the present report to more widely studied European languages. Our general goal, however, is the treatment of unrestricted natural languages.

* Department of Linguistics, University of Chicago, 1010 E. 59th Street, Chicago, IL 60637. E-mail: ja-goldsmith@uchicago.edu.

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. A first version was written in September, 1998, and a much-revised version was completed in December, 1999. This work was also supported in part by a grant from the Argonne National Laboratory-University 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, Jorma Rissanen, 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 Sylvain Neuvel has recently produced an interesting computational implementation of a theory of morphology that does not have a place for morphemes, as described at http://www.neuvel.net. It is well established that nonconcatenative morphology is found in some scattered language families, notably Semitic and Penutian. African tone languages require simultaneous morphological analyses of the tonal and the segmental material.

3 But see the following note.

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 that 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 statements 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 that LT works properly across a range of languages (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, since we still need good morphologies of many European languages and to produce a morphology of a given language "by hand" can take weeks or months. With the advent of considerable historical text available on-line (such as the ARTFL database 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.

Table 1
Some signatures from *Tom Sawyer*.

Signature	Example	Stem Count (type)	Token Count
NULL.ed.ing	betray betrayed betraying	69	864
NULL.ed.ing.s	remain remained remaining remains	14	516
NULL.s.	cow cows	253	3,414
e.ed.es.ing	notice noticed notices noticing	4	62

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.[4]

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, in Section 5.)

The present morphology learning algorithm is contained in a C++ program called *Linguistica* that runs on a desktop PC and takes a text file as its input.[5] Analyzing a

4 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; there is a very brief discussion below.

5 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*.

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 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.

In this paper, I will discuss prior work in this area (Section 2), the nature of the MDL model we propose (Section 3), heuristics for the task of the initial splitting of words into stem and affix (Section 4), the resulting signatures (Section 5), use of MDL to search the space of morphologies (Section 6), results (Section 7), the identification of entirely spurious generalizations (section 8), the grouping of signatures into larger units (Section 9), and directions for further improvements (Section 10). Finally, I will offer some speculative observations about the larger perspective that this work suggests and work in progress (Section 11).

2. Previous Research in this Area

The task of automatic word analysis has intrigued workers in a range of disciplines, and the practical and theoretical goals that have driven them have 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. (As noted earlier, 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), which 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 $n+1$st letter given the first n letters (or the mirror-image measure). This was first proposed by Zellig Harris (1955, 1967), and further developed by others, notably by Hafer and Weiss (1974). The second approach seeks to identify bigrams (and trigrams) that 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 related words. The fourth approach, which includes that used in this paper, is top-down, and seeks an analysis that is globally most concise. In this section, we shall review some of the work that has pursued these approaches—briefly, necessarily.[6] 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 that 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

6 Another effort is that attributed to Andreev (1965) and discussed in Altmann and Lehfeldt (1980), especially p. 195 and following, though their description does not facilitate establishing a comparison with the present approach.

case, for example, in Pacak and Pratt (1976), Koch, Küstner, and Rüdiger (1989), and Wothke and Schmidt (1992). With the exception of Harris's algorithm, the complexity of the algorithms is such as to make implementation for purposes of comparison prohibitively time-consuming.

At the heart of the first approach, due to Harris, is the desire to place boundaries between letters (respectively, phonemes) in a word based on conditional entropy, in the following sense. We construct a device that 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 one-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's 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 which I describe below. It has the same problem that good heuristics frequently have: it has many inaccuracies, and it does not lend itself to a next step, a qualitatively more reliable approximation of the correct solution.[7]

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 15 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 of prefix conditional entropy, in others by seeking a value above a threshold, and in yet others, good results are obtained only when this 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 6,200 word types. (Precision here indicates proportion of predicted morpheme breaks that are correct, and recall denotes 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 (respectively, 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 that never occur in that sequence

7 But Harris's 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.

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), and Flenner (1994, 1995). Any brief description that 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 (respectively, phonemes) is assigned a score that 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 that were explicitly assigned on the basis of overtly marked morpheme boundaries. But it remains unclear how one should proceed further 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) and proponents of this approach have felt some freedom on making this choice in an ad hoc fashion. Janssen (1992, 81–82) observes that the French word *linguistique* displays three peaks, predicting the analysis *linguist-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 at all why these numbers should be added; they do not, for example, 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 proposes 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; there is no scholarly tradition according to which the presence of zero in a sum should lead to a total of 0.

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. The major question to carry away from this approach is this: can the information that 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.[8]

8 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, Chap. 5).

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 (nonnominative) 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 Dzeroski 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.

Dzeroski 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 bears. 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.[9]

Two recent papers, Jacquemin (1997) and Gaussier (1999), deserve consideration here.[10] Gaussier (1999) approaches a very similar task to that which we consider, and takes some similar steps. His goal is to acquire derivational rules from an inflectional lexicon, thus insuring that his algorithm has access to the lexical category of the words it deals with (unlike the present study, which is allowed no such access). Using the terminology of the present paper, Gaussier considers candidate suffixes if they appear with at least two stems of length 5. His first task is (in our terms) to infer paradigms from signatures (see Section 9), which is to say, to find appropriate clusters of signatures. One example cited is *depart, departure, departer*. He used a hierarchical agglomerative clustering method, which begins with all signatures forming distinct clusters, and successively collapses the two most similar clusters, where similarity between stems is defined as the number of suffixes that two stems share, and similarity between clusters is defined as the similarity between the two least similar stems in the respective clusters. He reports a success rate of 77%, but it is not clear how to evaluate this figure.[11]

The task that Gaussier addresses is defined from the start to be that of derivational morphology, and because of that, his analysis does not need to address the problem of inflectional morphology, but it is there (front and center, so to speak) that the difficult clustering problem arises, which is how to ensure that the signatures *NULL.s.'s* (for nouns in English) and the signature *NULL.ed.s* (or *NULL.ed.ing.s*) are not assigned to single clusters.[12] That is, in English both nouns and verbs freely occur with the suffixes

9 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.

10 I am grateful to a referee for drawing my attention to these papers.

11 The analysis of a word w in cluster C counts as a success if most of the words that in fact are related to w also appear in the cluster C, and if the cluster "comprised in majority words of the derivational family of w." I am not certain how to interpret this latter condition; it means perhaps that more than half of the words in C contain suffixes shared by forms related to w.

12 In traditional terms, inflectional morphology is responsible for marking different forms of the same lexical item (lemma), while derivational morphology is responsible for the changes in form between distinct but morphologically related lexical items (lemmas).

NULL and *-s*, and while *-ed* and *-'s* disambiguate the two cases, it is very difficult to find a statistical and morphological basis for this knowledge.[13]

Jacquemin (1997) explores an additional source of evidence regarding clustering of hypothesized segmentation of words into stems and suffixes; he notes that the hypothesis that there is a common stem *gen* in *gene* and *genetic*, and a common stem *express* in *expression* and *expressed*, is supported by the existence of small windows in corpora containing the word pair *genetic. . . expression* and the word pair *gene. . . expressed* (as indicated, the words need not be adjacent in order to provide evidence for the relationship). As this example suggests, Jacquemin's work is situated within the context of a desire for superior information retrieval.

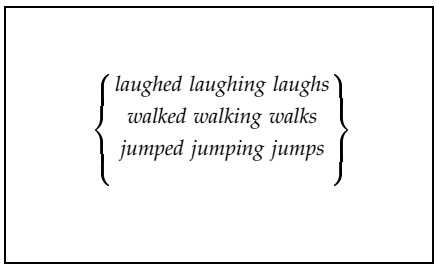
In terms of the present study, Jacquemin's algorithm consists of (1) finding signatures with the longest possible stems and (2) establishing pairs of stems that occur together in two or more windows of length 5 or less. He tests his results on 100 random pairs discovered in this fashion, placing upper bounds on the length of the suffix permitted between one and five letters, and independently varying the length of the window in question. He does not vary the minimum size of the stem, a consideration that turns out to be quite important in Germanic languages, though less so in Romance languages. He finds that precision varies from 97% when suffixes are limited to a length of one letter, to 64% when suffixes may be five letters long, with both figures assuming an adjacency window of two words; precision falls to 15% when a window of four words is permitted.

Jacquemin also employs the term *signature* in a sense not entirely dissimilar to that employed in the present paper, referring to the structured set of four suffixes that appear in the two windows (in the case above, the suffixes are *-ion*, *-ed*; *NULL*, *-tic*). He notes that incorrect signatures arise in a large number of cases (e.g., good: *optical control* ∼ *optimal control; adoptive transfer* ∼ *adoptively tranfer*, paralleled by bad: *ear disease* ∼ *early disease*), and suggests a quality function along the following lines: Stems are linked in pairs (*adopt-transfer, ear-disease*); compute then the average length of the shorter stem in each pair (that is, create a set of the shorter member of each pair, and find the average length of that set). The quality function is defined as that average divided by the length of the largest suffix in the signature; reject any signature class for which that ratio is less than 1.0. This formula, and the threshold, is purely empirical, in the sense that there is no larger perspective that bears on determining the appropriateness of the formula, or the values of the parameters.

The strength of this approach, clearly, is its use of information that co-occurrence in a small window provides regarding semantic relatedness. This allows a more aggressive stance toward suffix identification (e.g., *alpha interferon* ∼ *alpha2 interferon*). There can be little question that the type of corpus studied (a large technical medical corpus, and a list of terms—partially multiword terms) lends itself particularly to this style of inference, and that similar patterns would be far rarer in unrestricted text such as *Tom Sawyer* or the Brown corpus.[14]

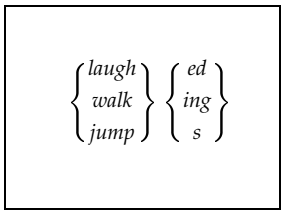
13 Gaussier also offers a discussion of inference of regular morphophonemics, which we do not treat in the present paper, and a discussion in a final section of additional analysis, though without test results. Gaussier aptly calls our attention to the relevance of minimum edit distance relating two potential allomorphs, and he proposes a probabilistic model based on patterns established between allomorphs. In work not discussed in this paper, I have explored the integration of minimum edit distance to an MDL account of allomorphy as well, and will discuss this material in future work.

14 In a final section, Jacquemin considers how his notion of signatures can be extended to identify sets of related suffixes (e.g., *onic/atic/ic*—his example). He uses a greedy clustering algorithm to successively add nonclustered signatures to clusters, in a fashion similar to that of Gaussier (who Jacquemin thanks for discussion, and of course Jacquemin's paper preceded Gaussier's paper by two years), using a



(a) Word list with no internal structure

Total letter count: 57 letters



(b) Word list with morphological structure

Total letter count: 19 letters

Figure 1
Naive description length.

The fourth approach to morphology analysis is top-down, and seeks a globally optimal analysis of the corpus. This general approach is based on the insight that the number of letters in a list of words is greater than the number of letters in a list of the stems and affixes that are present in the original list. This is illustrated in Figure 1. This simple observation lends hope to the notion that we might be able to specify a relatively simple figure of merit independently of how we attempt to *find* analyses of particular data. This view, appropriately elaborated, is part of the minimum description length approach that we will discuss in detail in this paper.

Kazakov (1997) presents an analysis in this fourth approach, using a straightforward measurement of the success of a morphological analysis that we have mentioned, counting 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.[15] 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 (respectively, 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

metric more complex than the familiar minimum edit distance, but no results are offered in support of the choice of the additional complexity.

15 I am grateful to Scott Meredith for drawing my attention to this paper.

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 (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 suffix (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" (p. 35). Brent's corpora were quite different 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 analysed stem and suffix for each word, the goal of the present study.[16] Brent makes clear the relevance and importance of information-theoretic notions, but does not provide a synthetic and overall measure of the length of the morphological grammar.

16 Brent's description of his algorithm is not detailed enough to satisfy the curiosity 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*) 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 two-letter suffix (e.g., the suffix *-al* in French).

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 two-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.

⟨[t he] { ([un it] ed) ([st at] es) }⟩⟨of {a me ([r ic]) a }⟩ (1)

Applying de Marcken's algorithm to a "broken" corpus of a language in which word boundaries are indicated (for example, English) provides interesting results, but none that provide anything even 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.[17] (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 mentioned briefly in note 21. 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. Minimum Description Length Model

The central idea of minimum description length analysis (Rissanen 1989) is composed of four parts: first, a model of a set of data assigns a probability distribution to the sample space from which the data is assumed to be drawn; second, the model can then be used to assign a compressed length to the data, using familiar information-theoretic notions; third, the model can itself be assigned a length; and fourth, the optimal analysis of the data is the one for which the sum of the length of the compressed data and the length of the model is the smallest. That is, we seek a minimally compact specification of *both* the model *and* the data, simultaneously. Accordingly, we use the conceptual vocabulary of information theory as it becomes relevant to computing the length, in bits, of various aspects of the morphology and the data representation.

3.1 A First Model

Let us suppose that we know (part of) the correct analysis of a set of words, and we wish to create a model using that knowledge. In particular, we know which words have no morphological analysis, and for all the words that *do* have a morphological analysis, we know the final suffix of the word. (We return in the next section to how we might arrive at that knowledge.) An MDL model can most easily be conceptualized if we encode all such knowledge by means of lists; see Figure 2. In the present case, we have three lists: a list of stems, of suffixes, and of signatures. We construct a list of the stems of the corpus defined as the set of the unanalyzed words, plus the material that precedes the final suffix of each morphologically analyzed word. We also construct a list of suffixes that occur with at least one stem. Finally, each stem is empirically associated with a set of suffixes (those with which it appears in the corpus); we call this set the stem's signature, and we construct a third list, consisting of the signatures that appear in this corpus. This third list, however, contains no letters (as the other

17 See note 21 below.

A. Affixes: 6

1.	NULL
2.	ed
3.	ing
4.	s
5.	e
6.	es

B. Stems: 9

1.	cat
2.	dog
3.	hat
4.	John
5.	jump
6.	laugh
7.	sav
8.	the
9.	walk

C. Signatures: 4

Signature 1:

$$\left\{\begin{array}{c} SimpleStem : ptr(cat) \\ SimpleStem : ptr(dog) \\ SimpleStem : ptr(hat) \\ ComplexStem : ptr(Sig2): ptr(sav) + ptr(ing) \end{array}\right\} \left\{\begin{array}{c} ptr(NULL) \\ ptr(s) \end{array}\right\}$$

Signature 2:

$$\{SimpleStem : ptr(sav)\} \left\{\begin{array}{c} ptr(e) \\ ptr(es) \\ ptr(ing) \end{array}\right\}$$

Signature 3:

$$\left\{\begin{array}{c} SimpleStem : ptr(jump) \\ SimpleStem : ptr(laugh) \\ SimpleStem : ptr(walk) \end{array}\right\} \left\{\begin{array}{c} ptr(NULL) \\ ptr(ed) \\ ptr(ing) \\ ptr(s) \end{array}\right\}$$

Signature 4:

$$\left\{\begin{array}{c} SimpleStem : ptr(John) \\ SimpleStem : ptr(the) \end{array}\right\}$$

Figure 2
A sample morphology. This morphology 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*.

lists do), but rather pointers to stems and suffixes. We do this, in one sense, because our goal is to construct the smallest morphology, and in general a pointer requires less information than an explicit set of letters. But in a deeper sense, it is the signatures whose compactness provides the explicit measurement of the conciseness of the entire analysis. Note that by construction, each stem is associated with exactly one signature.

Since *stem*, *suffix*, and *signature* all begin with *s*, we opt for using t to represent a s*t*em, f to represent a su*f*fix, and σ to represent a signature, while the uppercase T, F, Σ represent the sets of stems, suffixes, and signatures, respectively. The number of members of such a set will be represented $\langle T \rangle$, $\langle F \rangle$, etc., while the number of occurrences of a stem, suffix, etc., will be represented as [t], [f], etc. The set of all words in the corpus will be represented as W; hence the length of the corpus is [W], and the size of the vocabulary is $\langle W \rangle$.

Note the structure of the signatures in Figure 2. Logically a signature consists of two lists of pointers, one a list of pointers to stems, the other a list of pointers to suffixes. To specify a list of length N, we must specify at the beginning of the signature that N items will follow, and this requires just slightly more than $\log_2$ N bits to do (see Rissanen [1989, 33–34] for detailed discussion); I will use the notation λ(N) to indicate this function.

A pointer to a stem t, in turn, is of length $-\log \text{prob}\ (t)$, a basic principle of information theory (Li and Vitányi 1997). Hence the length of a signature is the sum of the (inverse) log probabilities of its stems, plus that of its suffixes, plus the number of bits it takes to specify the number of its stems and suffixes, using the λ function. We will return in a moment to how we determine the probabilities of the stems and suffixes; looking ahead, it will be the empirical frequency.

Let us consider the length of stem list T. As we have already observed, its length is $\lambda(\langle T \rangle)$—this is the length of the information specifying how long the list is—plus the length of each stem specification. In most of our work, we make the assumption that the length of a stem is the number of letters in it, weighted by the factor log 26 converting to binary bits, in a language with 26 letters.[18] The same reasoning holds for the suffix list F: its length is $X(\langle F \rangle)$ plus the length of each suffix, which we may take to be the total number of letters in the suffix times log 26.

We return to the question of how long the pointer (found inside a signature) to a stem or suffix is. The probability of a stem is its (empirical) frequency, i.e., the total number of words in the corpus corresponding to the words whose analysis includes the stem in question; the probability of a suffix is defined in parallel fashion. Using W to indicate all the words of the corpus, we may say that the length of a pointer to a stem t is of length

$$\log \frac{[W]}{[t]},$$

a pointer to suffix f is of length

$$\log \frac{[W]}{[f]},$$

18 This is a reasonable, and convenient, assumption, but it may not be precise enough for all work. A more refined measure would take the length of a letter to be -1 times the binary log of its frequency. A still more refined measure would base the probability of a letter on bigram context; this matters for English, where stem final t is very common. In addition, there is information in the linear order in which the letters are stored, roughly equal to

$$\sum_{k=1}^{n} \log_2 k$$

for a string of length n (compare the information that distinguishes the lexical representation of anagrams). This is an additional consideration in an MDL analysis of morphology pressing in favor of breaking words into morphemes when possible.

and a pointer to a signature σ is of length

$$\log \frac{[W]}{[\sigma]}.$$

We have now settled the question of how to determine the length of our initial model; we next must determine the probability that the model assigns to each word in the corpus, and armed with that knowledge, we will be able to compute the compressed length of the corpus.

The morphology assigns a probability to each word w as the product of the probability of w's signature times w's stem, given its signature, and w's suffix, given its signature: prob $(w = t + f)$ = prob (σ) prob $(t \mid \sigma)$ prob $(f \mid \sigma)$, where σ is the signature associated with t: $\sigma = \text{sig}(t)$. Thus while stems and suffixes, which are defined relative to a particular morphological model, are assigned their empirical frequency as their probability, words are assigned a probability based on the model, one which will always depart from the empirical frequency. The compression to the corpus is thus worse than would be a compression based on word frequency alone,[19] or to put it another way, the morphological analysis in which all words are unanalyzed is the analysis in which each word is trivially assigned its own empirical frequency (since the word equals the stem). But this decrease in compression that comes with morphological analysis is the price willingly paid for not having to enter every distinct word in the stem list of the morphology.

Summarizing, the compressed length of the corpus is

$$\sum_{w=t+f} [w] (\log \mathit{prob}(\sigma(w)) + \log \mathit{prob}(t) + \log \mathit{prob}(f \mid \sigma(w))),$$

where we have summed over the words in the corpus, and $\sigma(w)$ is the signature to which word w is assigned. The compressed length of the model is the length of the stem list, the suffix list, and the signature list. The length in bits of the stem list is

$$\lambda(\langle T \rangle) + \sum_{t \in Stems} L_{typo}(t)$$

and the length of the suffix list is

$$\lambda(\langle F \rangle) + \sum_{f \in Suffixes} L_{typo}(f),$$

where $L_{typo}()$ is the measurement of the length of a string of letters in bits, which we take to be $\log_2 26$ times the number of letters (but recall note 18). The length of the signature list is

$$\lambda(\langle \Sigma \rangle) + \sum_{\sigma \in Signatures} L(\sigma),$$

where L(σ) is the length of signature σ. If the set of stems linked to signature σ is T(σ) and the set of suffixes linked to signature σ is F(σ), then

$$\lambda(\langle T(\sigma) \rangle) + \lambda(\langle F(\sigma) \rangle) + \sum_{t \in T(\sigma)} \log \frac{[W]}{[t]} + \sum_{f \in F(\sigma)} \log \frac{[\sigma]}{[\mathit{words}(f) \cap \mathit{words}(\sigma)]}.$$

19 Due to the fact that the cross-entropy is always greater than or equal to the entropy.

(The denominator in the last term consists of the token count of words in a particular signature with the given suffix f, and we will refer to this below more simply as $[f \ in \ \sigma]$.)

It is no doubt easy to get lost in the formalism, so it may be helpful to point out what the contribution of the additional structure accomplishes. We observed above that the MDL analysis is an elaboration of the insight that the best morphological analysis of a corpus is obtained by counting the total number of letters in the list of stems and suffixes according to various analyses, and choosing the analysis for which this sum is the least (cf. Figure 2). This simple insight fails rapidly when we observe in a language such as English that there are a large number of verb stems that end in *t*. Verbs appear with a null suffix (that is, in bare stem form), with the suffixes *-s*, *-ed*, and *-ing*. But once we have 11 stems ending in *t*, the naive letter-counting approach will judge it a good idea to create a new set of suffixes: *-t*, *-ted*, *-ts*, and *-ting*, because those 10 letters will allow us to remove 11 or more letters from the list of stems. It is the creation of the lists, notably the signature list, and an information cost which increases as probability decreases, that overcomes that problem. Creating a new signature may save some information associated with the stem list in the morphology, but since the length of pointers to a signature σ is $-\log \ \text{freq} \ (\sigma)$, the length of the pointers to the signatures for all of the words in the corpus associated with the old signature (*-Ø*, *-ed*, *-s*, *-ing*) or the new signature (*-ts*, *-ted*, *-ting*, *-ts*) will be longer than the length of the pointers to a signature whose token count is the sum of the token count of the two combined, i.e.,

$$x \log \left(\frac{[W]}{x} \right) + y \log \left(\frac{[W]}{y} \right) \geq (x + y) \log \left(\frac{[W]}{x + y} \right).$$

3.2 Recursive Morphological Structure

The model presented above is too simple in that it underestimates the gain achieved by morphological analysis in case the word that is analyzed is also a stem of a larger word. For example, if a corpus contains the words *work* and *working*, then morphological analysis will allow us to dispense with the form *working*; it is modeled by the stem *work* and the suffixes *-Ø* and *-ing*. If the corpus also includes *workings*, the analysis *working-s* additionally lowers the cost of the stem *working*. Clearly we would like stems to be in turn analyzable as stems + suffixes. Implementing this suggestion involves the following modifications: (i) Each pointer to a stem (and these are found both in the compressed representation of each individual word in the corpus, and inside the individual signatures of the morphological model) must contain a flag indicating whether what follows is a pointer to a simple member of the stem list (as in the original model), or a triple pointer to a signature, stem, and suffix. In the latter case, which would be the case for the word *[work-ing]-s*, the pointer to the stem consists of a triple identical to the signature for the word *work-ing*. (ii) The number of words in the corpus has now changed, in that the word *[work-ing]-s* now contains two words, not one. We will need to distinguish between counts of a word *w* where *w* is a freestanding 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 nonintegral fashion over the two or more nested word structures found in complex words. We consider the more complex case in the appendix.[20]

20 In addition, the number of words in a corpus will change if the analysis determines that all occurrences of (let us say) *-ings* are to be reanalyzed as complex words, and the stem in question

We may distinguish between those words, like *work* or *working*, whose immediate analysis involves a stem appearing in the stem list (we may call these W_{SIMPLE}) and those whose analysis, like *workings*, involves recursive structure (we may call these $W_{COMPLEX}$). As we have noted, every stem entry in a signature begins with a flag indicating which 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. We also keep track separately of the total number of words in the corpus (token count) that are morphologically analyzed, and refer to this set as W_A; this consists of all words except those that are analyzed as having no suffix (see item (ii) in (2), below).

(2) Compressed length of morphology

(i) $\lambda\langle T\rangle + \lambda\langle F\rangle + \lambda\langle\Sigma\rangle$

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

(ii) *Suffix list*: $\sum_{f \in Suffixes} \left(\log 26^* length(f) + \log \frac{[W_A]}{[f]} \right)$

(iii) *Stem list*: $\sum_{t \in T} \left(\log 26^* length(t) + \log \left(\frac{[W]}{[t]} \right) \right)$

(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:
$\lambda(\langle \text{stems}(\sigma)\rangle) + \lambda(\langle \text{suffixes}(\sigma)\rangle)$

(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

$$\log \frac{[W]}{[W_{SIMPLE}]}$$

(perhaps *work-ing*) did not appear independently as a freestanding word in the corpus; we will refer to these inferred words as being "virtual" words with virtual counts.

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

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

(d) a pointer to each suffix, of total length

$$\sum_{f \in Suffixes(\sigma)} \log \frac{[\sigma]}{[f \text{ in } \sigma]}.$$

(3) Compressed length of corpus

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

MDL thus provides a figure of merit that we wish to minimize, and we will seek heuristics that modify the morphological analysis in such a fashion as to decrease 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.

4. Heuristics for Word Segmentation

The MDL model designed in the preceding section will be of use only if we can provide a practical means of creating one or more plausible morphologies for a given corpus. That is, we need bootstrapping heuristics that enable us to go from a corpus to such a morphology. As we shall see, it is not in fact difficult to come up with a plausible initial morphology, but I would like to consider first an approach which, though it might seem like the most natural one to try, fails, and for an interesting reason.

The problem we wish to solve can be thought of as one suited to an expectation-maximization (EM) approach (Dempster, Laird, and Rubin 1977). Along such a line, each word w of length N would be initially conceived of as being analyzed in N different ways, cutting the word into stem + suffix after i letters, $1 \leq i \leq N$, with each of these N analyses being assigned probability mass of

$$\frac{[w]}{N[W]}.$$

That probability mass is then summed over the resulting set of stems and suffixes, and on successive iterations, each of the N cuts into stem + suffix is weighted by its probability; that is, if the *i*th cut of word w, of length l, cuts it into a stem t of length i and suffix of length $l - i$, then the probability of that cut is defined as

$$\frac{pr(stem\ t = w_{1,i})\, pr(suffix\ f = w_{i+1,l})}{\sum_{k=1}^{N} pr(stem\ t = w_{1,k}) pr(suffix\ f = w_{k+1,l})},$$

where $w_{j,k}$ refers to the substring of w from the jth to the kth letter. Probability mass for the stem and the suffix in each such cut is then augmented by an amount equal to the frequency of word w times the probability of the cut. After several iterations (approximately four), estimated probabilities stabilize, and each word is analyzed on the basis of the cut with the largest probability.

This initially plausible approach fails because it always prefers an analysis in which either the stem or (more often) the suffix consists of a single letter. More importantly, the probability that a sequence of one or more word-final letters is a suffix is very poorly modeled by the sequence's frequency.[21] To put the point another way, even the initial heuristic analyzing one particular word must take into account all of the other analyses in a more articulated way than this particular approach does.

I will turn now to two alternative heuristics that succeed in producing an initial morphological analysis (and refer to a third in a note). It seems likely that one could construct a number of additional heuristics of this sort. The point to emphasize is that the primary responsibility of the overall morphology is not that of the initial heuristic, but rather of the MDL model described in the previous section. The heuristics described in this section create an *initial* morphology that can serve as a starting point in a search for the shortest overall description of the morphology. We deal with that process in Section 5.

4.1 First Heuristic

A heuristic that I will call the *take-all-splits* heuristic, and which considers *all* cuts of a word of length l into stem+suffix $w_{1,i} + w_{i+1,l}$, where $1 \leq i < l$, much like the EM approach mentioned immediately above, works much more effectively if the probability is assigned on the basis of a Boltzmann distribution; see (4) below. The function H(.) in (4) assigns a value to a split of word w of length l: $w_{1,i} + w_{i+1,l}$. H does not assign a proper distribution; we use it to assign a probability to the cut of w into $w_{1,i} + w_{i+1,l}$ as in (5). Clearly the effect of this model is to encourage splits containing relatively long suffixes and stems.

$$H(w_{1,i} + w_{i+1,l}) = -(i \log\ freq\ (stem\ = w_{1,i}) + (l - i) \log\ freq\ (suffix\ = w_{i+1,l})) \quad (4)$$

$$prob\ (w = w_{1,i} + w_{i+1,l}) = \frac{1}{Z} e^{-H(w_{1,i} + w_{i+1,l})} \quad (5)$$

21 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 probability is freq (*diplomac*)* freq (*y*), and constrast that value with the corresponding values of two other analyses: freq (*diploma*)* freq (*cy*), and freq (*diplom*)* freq (*acy*). Now, the ratio of the frequency of words that begin with *diploma* and those that begin with *diplomac* is less than 3, while the ratio of the 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.

where

$$Z = \sum_{i=1}^{n-1} H(w_{1,i} + w_{i+1,l})$$

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 five iterations on the entire lexicon.[22] 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 Second Heuristic

The second approach that 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 two and six letters that appear word finally. Thus, for example, the word *elephant#* contains one occurrence of the word-final bigram *t#*, one occurrence of the word-final trigram *nt#*, and so forth; we stop at 6-grams, on the grounds that no grammatical morphemes require more than five letters in the languages we are dealing with. 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 $n_1 n_2 \ldots n_k$ is a grammatical morpheme the measure:

$$\frac{[n_1 n_2 \ldots n_k]}{\textit{Total count of k-grams}} \log \frac{[n_1 n_2 \ldots n_k]}{[n_1][n_2] \ldots [n_k]},$$

which we may refer to as the **weighted mutual information**. 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 incorrect results as well as correct ones; 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 *ings* 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 Evaluating the Results of Initial Word Splitting

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.[23] In addition, if we follow the *take-all-splits* approach, we have many

22 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.

23 Various versions of Harris's method of morpheme identification can be used as well. Harris's approach has the interesting characteristic (unlike the heuristics discussed in the text) that it is possible to impose restrictions that improve its precision while at the same time worsening its recall to unacceptably low levels. In work in progress, we are exploring the consequences of using such an initial heuristic with significantly higher precision, while depending on MDL considerations to extend the recall of the entire morphology.

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 return to the central notion of the signature.

5. Signatures

Each word now has been assigned an optimal split into stem and suffix by the initial heuristic chosen, 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 miniparadigm. For example, in one English corpus, the stems *despair, pity, appeal*, and *insult* appear with the suffixes *ing* and *ingly*. However, they also appear as freestanding words, and so we use the word *NULL*, to indicate a zero suffix. Thus their signature is *NULL.ing.ingly*. Similarly, the stems *assist* and *ignor* 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 nonregular signatures produced by the *take-all-splits* 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 pseudostem 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 (6), 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.)[24]

$$\begin{Bmatrix} stem_1 \\ stem_2 \\ stem_3 \end{Bmatrix} \begin{Bmatrix} suffix_1 \\ suffix_2 \end{Bmatrix} \tag{6}$$

If we have a morphological pattern of five suffixes, let us say, and there is a large set of stems that appear with all five suffixes, then that set will give rise to a regular signature with five suffixal members. This simple pattern would be perturbed by the (for our purpose) extraneous fact that a stem appearing with these suffixes

24 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, 161) as well as Nida (1948, 1949).

should also appear with some other suffix; and if all stems that associate with these five suffixes appear with idiosyncratic suffixes (i.e., each different from the others), then the signature of those five 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 suffixes 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.

To get a sense of what are identified as regular signatures in a language such as English, let us look at the results of a preliminary analysis in Table 2 of 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 σ 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 Table 3 for the final analysis of this text.

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 that I have studied.

These results, then, are derived by the application of the heuristics described above. The overall sketch of the morphology of the language is quite reasonable already in its outlines. Nevertheless, the results, when studied up close, show that there remain a good number of errors that 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 *ts* can lead 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. For example, *s* 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*.

Table 2
Top 81 signatures from *Tom Sawyer*.

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

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.

Table 3
Suffixes from *Tom Sawyer*.

Suffix	Remarks	Suffix	Remarks
s		ted	chat-ted, fit-ted, submit-ted, etc.
ed		est	
ing		ity	
er		ous	
e		ard	drunk-ard
ly		able	
’s		ious	
d		less	
y		ment	
n		id	id.or for stems horr-, splend-, liqu-
on	Spurious (bent-on, rivers-on): triage issue	ure	
es		ive	
t		ty	novel, uncertain, six, proper
st	Signature NULL.ly.st, for stems such as safe-	ence	
en	behold, deaf, weak, sunk, etc.	ily	
le	Error: analyzed le.ly for e.y (stems such as feeb-, audib-, simp-).	ward	
al		ation	
n’t		led	triage problem
nce	Signature nce.nt, for stems fragr-, dista-, indiffere-	’d	
ent	Spurious: triage problem (pot-ent)	ry	error: stems such as glo- with signature rious.ry
tion		rious	error: stems such as glo- with signature rious.ry
r		rs	error: r should be in stem
ter	triage problem	ned	awake-ned, white-ned, thin-ned
k	triage problem	ning	begin-ning, run-ning
ful		age	
ion		h	triage problem
’ll		te	should be -ate (e.g., punctua-te)
an	triage problem	ant	triumph-ant, expect-ant
ness		r’s	error
nt	see above	ance	

6. Optimizing Description Length Using Heuristics and MDL

We can use the description length of the grammar formulated in (2) and (3) 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.[25]

25 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 (see the Appendix). The second way to speed up the task is, again, to 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

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, ists, ians, ened, lines, ents*, and *ively*. Note that what it means to say that "suffixes are checked to see if they are spurious amalgams" is that each suffix is checked to see if it is the concatenation of two independently existing suffixes, and then if that is the case, the entire description length of the corpus is recomputed under the alternative analysis; the reanalysis is adopted if and only if the description length decreases. The same holds for the other heuristics discussed immediately below.[26]

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*.[27] Such signatures are evaluated to determine if the description length improves when such a signature is modified to become *e.ing.s*, etc. 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

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 freestanding 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.

26 This is accomplished by the command *am4* in *Linguistica*.

27 This is accomplished by the command *am5* in *Linguistica*.

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 10-suffix signature is correct, then the discovery of a subsignature 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 *rags* 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 (one letter for none, one letter for one letter, one letter for two 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 meaningful cases, such as these, from the (typically less frequent and) spurious cases of stems whose forms are parallel but ultimately not related.

7. 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 a 500,000-word corpus of English (Table 4), a 350,000-word corpus of French (Table 5), *Don Quijote*, which contains 124,716 words of Spanish (Table 6), a 125,000-word corpus of Latin (Table 7), and 100,000 words and 1,000,000 words of Italian (Tables 8 and 9). 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, compare the signatures obtained in Italian on a corpus of 100,000 words (Table 8) and a corpus of 1,000,000 words (Table 9). When one sees the rich inflectional

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

1. NULL.ed.ing.s	**4. NULL.s**	**7. NULL.ed.ing**
accent	abberation	applaud
add	abolitionist	arrest
administer	abortion	astound
afford	absence	blast
alert	abstractionist	bless
amount	abutment	bloom
appeal	accolade	boast
assault	accommodation	bolster
attempt	accomodation	broaden
		cater
2. 's.NULL.s	**5. e.ed.es.ing**	
adolescent	achiev	**8. NULL.er.ing.s**
afternoon	assum	blow
airline	brac	bomb
ambassador	chang	broadcast
amendment	charg	deal
announcer	compris	draw
architect	conced	drink
assessor	conclud	dwell
association	decid	farm
	describ	feed
3. NULL.ed.er.ing.s		feel
attack	**6. e.ed.er.es.ing**	
back	advertis	**9. NULL.d.s**
bath	announc	abbreviate
boil	bak	accommodate
borrow	challeng	aggravate
charm	consum	apprentice
condition	enforc	arcade
demand	gaz	balance
down	glaz	barbecue
flow	invad	bruise
	liv	catalogue
	pac	costume
		10. NULL.ed.s
		acclaim
		beckon
		benefit
		blend
		blister
		bogey
		bother
		breakfast
		buffet
		burden

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

1. NULL.e.es.s	**4. NULL.e.es**	**7. NULL.e**
abondant	acquis	accueillant
abstrait	aéropostal	acharné
adjacent	afghan	admis
approprié	albanais	adsorbant
atteint	allongé	albigeois
bantou	anglais	alicant
bleu	appelé	aliénant
brillant	arrondi	alléchant
byzantin	bavarois	amarant
	carthaginois	ambiant
2. NULL.s		
abandonnée	**5. NULL.e.s**	**8. NULL.es.s**
abbaye	adhérent	antioxydant
abdication	adolescent	bassin
abdominale	affilié	civil
abélienne	aîné	craint
aberration	assigné	cristallin
abolitionniste	assistant	cutané
abordée	bovin	descendant
abrasif	cinglant	doté
abréviation	colorant	emulsifiant
		ennemi
3. NULL.ment.s	**6. NULL.ne.s**	
administrative	abélien	**9. a.aient.ait.ant.e.ent.er.es.èrent.é.ée.és**
agressive	acheuléen	contrôl
anatomique	alsacien	jou
ancienne	amérindien	laiss
annuelle	ancien	rest
automatique	anglo-saxon	
biologique	araméen	**10. NULL.es**
chimique	aristotélicien	adopté
classique	athénien	âgé
		allié
		annulé
		apparenté
		apprécié
		armé
		assiégé
		associé
		attaché

pattern emerging, as with the example of the 10 suffixes on first-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.[28]

28 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 (seven such verb stems are identified). Signature 3 is a subset of Signature 1, composed of stems accidentally not found in the feminine plural form. 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.

Table 6
Top 10 signatures, 130,000-word Spanish 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	
cubiert	acompañaba	**8. NULL.es**
cuy	acordaba	ángel
delicad	aguardaba	animal
		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	
acontecimiento	fuerza	**9. da.do.r**
acosado	hubiera	amanceba
acostumbrado	miente	ata
3. a.o.os	**6. a.as.o**	averigua
afligid	agradezc	colga
ánim	anch	emplea
asalt	atónit	feri
caballeriz	confus	fingi
desagradecid	conozc	heri
descubiert	decill	pedi
despiert	dificultos	persegui
dorad	estrech	
enemig	extrañ	**10. NULL.le**
flac	fresc	abrazó
		acomodar
		aconsejó
		afligióse
		agradeció
		aguardar
		alegró
		arrojó
		atraer
		besó

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 ninth among signatures. It contains a large part of the suffixal pattern from the most common regular conjugation, the first 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-grained look at the results and quantify them. To

Table 7
Top 10 signatures, 125,000-word Latin 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	
die	popul	**9. NULL.e.m.s**
ezechia		azaria
facultate	**6. e.em.es.i.ibus.is.um**	banaia
fide	fratr	esaia
fimbria	greg	iosia
	homin	iuda
3. a.ae.am.as.is	reg	lucusta
ancill	vic	massa
aqu	voc	matthathia
lucern		pluvia
parabol		sagitta
plag		
puell		**10. i.o.um**
stell		brachi
synagog		carmel
tabul		cenacul
tunic		damn
		evangeli
		hysop
		lectul
		liban
		offici
		ole

do this, we have selected from the English and the French analyses a set of 1,000 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 Tables 10 and 11.

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 *abolish*. 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.

Table 8
Top 10 signatures, 100,000-word Italian 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, 1,000,000-word Italian 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.ó.	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%

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 an error, but *John-son* (and a bit less obviously *Wat-son*) have not been treated as errors.[29] 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 second-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, Good indicates 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 that are analyzed, but incorrectly analyzed, by the algorithm. Failed to Analyze are the cases of words that are bimorphemic but

29 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.

for which no analysis was provided by the algorithm, and Spurious Analysis are the cases of words that 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 Tables 10 and 11. For English, these figures correspond to precision of $829/(829 + 52 + 83) = 85.9\%$, and recall of $829/(829 + 52 + 36) = 90.4\%$.

8. 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 that 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 straightforward 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 that 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. 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). 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 suffixes, and there were 1,490 signatures altogether. Application of triage leads to the loss

of only 240 signatures. The single-suffix signatures that were eliminated were: *ide, it, rs, he, ton, o*, and *ie*, all of which are spurious. However, a number of signatures that 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 that were in fact correct, the percentage of the words whose analysis was incorrectly eliminated by triage was 21.9% (236 out of 1,077 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 that departs from the MDL-based optimization in certain cases, which I shall identify in a moment. I believe that when the improvements identified in Section 10 below are made, 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 five, 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.

9. 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.[30] When two signatures as we have defined them here are collapsed, there are two major effects on the description length: pointers to the merged signature are shorter—leading to a shorter total description length—but, in general, predicted frequencies of the com-

30 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 $\{a_i^1\}$ in one case, and $\{a_i^2\}$in the other, maximal compression is obtained by choosing the case $(k = 1, 2)$ for which

$$\sum_i \log(a_i^k)$$

is the greatest.

posite 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.[31] 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.[32]

10. 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 semiregular 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' *écri* and *écriv*, each used in distinct forms. It would also be reasonable, in this case, to analyze the latter stem *écriv* as composed of *écri* plus a suffix *-v*, although in this case, there are no additional benefits to be gained from the more fine-grained analysis.

31 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 ith 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 $\mathbf{b} = \{b_k\}$ 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:

$$\sum_{t_n \in stems(\sigma)} \prod_k (1 - b_k)(1 - prob(f_i))^{[t_n]}.$$

Thus all that is necessary is to estimate the hidden parameters of the frequencies of the individual suffixes in the entire paradigm. See the following note as well.

32 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.

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*.[33]

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. Identifying 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

33 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^*:

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

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 $D(\sigma^* \parallel \sigma)$:

$$\begin{aligned} [t] \log[t] - \sum c_i \log \left(\frac{c_i}{\varphi_i} \right) &= [t] \log[t] - [t] \sum d_i \log \left(\frac{[t] d_i}{\varphi_i} \right) \\ &= [t] \log[t] - [t] D(\sigma^* \parallel \sigma) - [t] \sum d_i \log([t]) \\ &= [t] \log[t] - [t] D(\sigma^* \parallel \sigma) - [t] \log[t] \\ &= -[t] D(\sigma^* \parallel \sigma). \end{aligned}$$

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 that 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 $\langle x \rangle$ which deletes a preceding character x, and with the mechanism, we can establish a single signature *NULL.*$\langle e \rangle$*ed.*$\langle e \rangle$*ing.s*, composed of familiar suffixes *NULL* and *s*, plus two suffixes $\langle e \rangle$*ed* and $\langle e \rangle$*ing*, which delete a preceding (stem-final) *e* if one is present.

11. Conclusion

Linguists face at the present time the question of whether, and to what extent, information-theoretic notions will play a significant 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 mainstream 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 that simplifies the task of morpheme identification? It is the belief that the answer to this question is *yes* that 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) that 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 nonanalyzed, 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 as an optimization problem, of which MDL is one particular style.

Chomsky's early conception of generative grammar (Chomsky 1975 [1955]; henceforth *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 this: the *LSLT* approach conjectured that the grammar of a language could be factored into two parts, one universal and one language-particular; and when we look for the simplest grammatical description of a given corpus (the child's input) it is only the language-particular part of the description that contributes to complexity—that is what the theory stipulates. By contrast, the MDL approach makes minimal universal assumptions, and so the complexity of everything comprising the description of the corpus must be counted in determining the complexity of the description. 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 it is less significant as language-particular grammars grow 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 below.

There is a striking point that has so far remained tacit 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.[34] 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, 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 grammars 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, and the MDL measurement, which makes the final judgment call, deciding whether to accept a modification proposed by the heuristics, on the other.

On so much, the 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 insurmountable, because this corpus 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) that 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

34 Though see Dobrin (1999) for a sophisticated look at this problem.

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.[35]

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 2001). Notions of probabilistic grammar due to 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

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 (7)–(11), calculating the length of S_1 minus the length of S_2. The general formulas derived in (7)–(11) are not of direct computational interest; they serve rather as a template that 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. This template 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 nonzero 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 formulas in

35 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.

(7)–(11) 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 (7) 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

$$\log \frac{M_1}{x} - \log \frac{M_2}{y} = \log \frac{M_1}{M_2} - \log \frac{x}{y};$$

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

$$\log \frac{x_{state1}}{x_{state2}}$$

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; if no confusion would result, we write Δx.[36]

Let us review the terms listed in (7)–(11). 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 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 (7)–(11) where the two may be different. ΔW_S and ΔW_C are similar measures in the change of words that have 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 first the case of those morphemes that 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 (8), 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 the third term involves suffixes present only in S_1 and the fourth term involves suffixes present only in S_2. This format will appear in all of the components of this computation. Recall that the function L_{typo} specifies the length of a string in bits, which we may take here to be simply $\log(26)$ times the number of characters in the string.

In (9), we derive the corresponding formula for the stem component.

The general form of the computation of the change to the signature component (10) 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

36 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 "$\Delta\langle$W$\rangle$" for "$\Delta(\langle$W$\rangle)$".

as in (10a). But the heart of the matter is the treatment of the stems and suffixes within the signatures, given in (10b)–(10d).

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 (10d), and is relatively straightforward, but the treatment of stems (10c) 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 that 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 (10c.i). In (10c.ii), we need change in the total length of the pointers to the stems, and this has actually already been computed, during the computation of (9).[37] Finally in (10c.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 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 (11).

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

$$\Delta\langle\text{Suffixes}\rangle + \Delta\langle\text{Stems}\rangle + \Delta\langle\text{Signatures}\rangle$$

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

$$\Delta W\langle \mathit{Suffixes}\rangle_{(1,2)} - \sum_{f\in \mathit{Suffixes}_{(1,2)}} \Delta f + \sum_{f\in \mathit{Suffixes}_{(1,\sim 2)}} \left[\log \frac{[W]_1}{[f]} + L_{typo}(f)\right] - \sum_{f\in \mathit{Suffixes}_{(\sim 1,2)}} \left[\log \frac{[W]_2}{[f]} + L_{typo}(f)\right]$$

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

$$\Delta W\,\langle \mathit{Stems}\rangle_{(1,2)} - \sum_{t\in \mathit{Stems}_{(1,2)}} \Delta t + \sum_{t\in \mathit{Stems}_{(1,\sim 2)}} \left[\log \frac{[W]_1}{[t]} + L_{typo}(t)\right] - \sum_{t\in \mathit{Stems}_{(\sim 1,2)}} \left[\log \frac{[W]_2}{[t]} + L_{typo}(t)\right]$$

37 The equivalence between the number computed in (9) and the number needed here is not exactly fortuitous, but it is not an error either. The figure computed in (9) describes an aspect of the complexity of the morphology as a whole, whereas 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 that permitted the same stem to appear in several distinction signatures, and in that case, the computation here would not reduce to (9). But the assumption made in the text is entirely reasonable, and simplifies the construction for us.

(10) Difference in description length for the signature component of the morphology: (a) + (b) + (c) + (d)

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

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

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

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

(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:

$$\begin{aligned} &\langle W_{SIMPLE}\rangle_{1,2}\,(\Delta W - \Delta W_{SIMPLE}) \\ &\quad + \langle W_{COMPLEX}\rangle_{1,2}{}^{*}\,(\Delta W - \Delta W_{COMPLEX}) \\ &\quad + \langle W_{SIMPLE}\rangle_{1,\sim 2} \log \frac{[W]_1}{[W_{SIMPLE}]_1} \\ &\quad - \langle W_{SIMPLE}\rangle_{\sim 1,2} \log \frac{[W]_2}{[W_{SIMPLE}]_2} \\ &\quad + \langle W_{COMPLEX}\rangle_{1,\sim 2} \log \frac{[W]_1}{[W_{COMPLEX}]_1} \\ &\quad - \langle W_{COMPLEX}\rangle_{\sim 1,2} \log \frac{[W]_2}{[W_{COMPLEX}]_2} \end{aligned}$$

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

$$\Delta W \langle Stems\rangle_{(1,2)} - \sum_{t \in\ Stems_{(1,2)}} \Delta t + \sum_{t \in\ Stems_{(1,\sim 2)}} \log \frac{[W]_1}{[t]} - \sum_{t \in\ Stems_{(\sim 1,2)}} \log \frac{[W]_2}{[t]}$$

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

$$\Delta W \langle W_{COMPLEX}\rangle_{(1,2)} + \sum_{w \in W_{COMPLEX\ (1,2)}} \Delta stem(w)$$

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

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

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

(11) Change in compressed length of corpus

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

References

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

Andreev, Nikolai Dmitrievich, editor. 1965. *Statistiko-kombinatornoe modelirovanie iazykov*. Nauka, Moscow, Russia.

Baroni, Marco. 2000. Paper presented at the annual meeting of the Linguistics Society of America. Chicago, IL.

Bloomfield, Leonard. 1933. *Language*. H. Holt and Company, New York.

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, Lawrence Erlbaum Associates, Hillsdale, NJ.

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

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

de Marcken, Carl. 1995. *Unsupervised Language Acquisition*. Ph.D. dissertation, MIT, Cambridge, MA.

Dempster, Arthur Pentland, Nan M. Laird, and Donald 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*. Ph.D. dissertation, Department of Linguistics, University of Chicago, Chicago, IL.

Dzeroski, Saso and Tomaz Erjavec. 1997. Induction of Slovene nominal paradigms. In Nada Lavrac and Saso Dzeroski, editors, *Inductive Logic Programming, 7th International Workshop, ILP-97*, pages 17–20, Prague, Czech Republic, September. Lecture Notes in Computer Science, Vol. 1297. Springer, Berlin.

Flenner, Gudrun. 1994. Ein quantitatives Morphsegmentierungssystem für spanische Wortformen. In Ursula Klenk, editor, *Computatio Linguae II*. Steiner Verlag, Stuttgart, pages 31–62.

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

Gaussier, Eric. 1999. Unsupervised learning of derivational morphology from inflectional lexicons. In *Proceedings of the Workshop on Unsupervised Learning in Natural Language Processing*, pages 24–30. Association for Computational Linguistics.

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

Goldsmith, John. 2001. On information theory, entropy, and phonology in the 20th century. *Folia Linguistica* XXXIV(1–2): 85–100.

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

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

Hafer, Margaret A. and Stephen F. Weiss. 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. *Transformations and Discourse Analysis Papers* 73, Department of Linguistics, University of Pennsylvania. Reprinted in Harris 1970.

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

Jacquemin, Christian. 1997. Guessing morphology from terms and corpora. *Proceedings of SIGIR 97*, pages 156–165, ACM, Philadelphia.

Janssen, Axel. 1992. Segmentierung französischer Wortformen in Morphe ohne Verwendung eines Lexikons. In Ursula Klenk, editor, *Computatio Linguae*. Steiner Verlag, Stuttgart, pages 74–95.

Karttunen, Lauri. 1993. Finite state constraints. In John Goldsmith, editor, *The Last Phonological Rule*. University of Chicago Press, pages 173–194.

Kazakov, Dimitar. 1997. Unsupervised learning of naïve morphology with genetic algorithms. In W. Daelemans, A. van den Bosch, and A. Weijtera, editors, *Workshop Notes of the ECML/Mlnet Workshop on Empirical Learning of Natural Language Processing Tasks, April 26, 1997*.

Klenk, Ursula. 1992. Verfahren morphologischer Segmentierung und die Wortstruktur im Spanischen. In Ursula Klenk, editor, *Computatio Linguae*. Steiner Verlag, Stuttgart.

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

Koskenniemi, Kimmo. 1983. Two-level morphology: A general computational model for word-form recognition and production. Publication no. 11, Department of General Linguistics, University of Helsinki, Helsinki.

Langer, Hagen. 1991. Ein automatisches Morphosegmentierungsverfahren für deutsche Wortformen. Manuscript.

Li, Ming and Paul Vitányi. 1997. *An Introduction to Kolmogorov Complexity and its Applications* (2nd ed.). Springer, New York.

Nida, Eugene. 1948. The identification of morphemes. In Martin Joos, editor, *Readings in Linguistics I*. University of Chicago Press, Chicago, IL, pages 255–271.

Nida, Eugene. 1949. *Morphology: The Descriptive Analysis of Words*. University of Michigan, Ann Arbor.

Pacak, Milos and Arnold 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, Singapore.

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

Wothke, Klaus and Rudolf Schmidt. 1992. A morphological segmentation procedure for German. *Sprache und Datenverarbeitung*, 16(1):15–28.