Computational Learning of Morphology

John A. Goldsmith,[1,2] Jackson L. Lee,[2] and Aris Xanthos[3]

[1]Department of Computer Science, The University of Chicago, Chicago, Illinois 60637; email: goldsmith@uchicago.edu

[2]Department of Linguistics, The University of Chicago, Chicago, Illinois 60637

[3]Section des sciences du langage et de l'information, Université de Lausanne, CH-1015 Lausanne, Switzerland

Annu. Rev. Linguist. 2017. 3:85–106

First published online as a Review in Advance on November 17, 2016

The *Annual Review of Linguistics* is online at linguist.annualreviews.org

This article's doi: 10.1146/annurev-linguistics-011516-034017

Keywords

morphology, unsupervised learning, language induction, grammar induction, minimum description length, adaptor grammars, Gibbs sampling

Abstract

This article reviews research on the unsupervised learning of morphology, that is, the induction of morphological knowledge with no prior knowledge of the language beyond the training texts. This is an area of considerable activity over the period from the mid 1990s to the present. It is of particular interest to linguists because it provides a good example of a domain in which complex structures must be induced by the language learner, and successes in this area have all relied on quantitative models that in various ways focus on model complexity and on goodness of fit to the data.

1. INTRODUCTION

1.1. Goals

In this article, we review the literature on the computational, unsupervised learning of natural language morphology, and offer our view of the important questions that have been raised and, to some degree, answered. Thus, we look at the efforts to date to devise an algorithm that takes raw textual data as its input and provides a linguistic analysis of the morphological structure of the language from which the text was taken, with no prior knowledge of the language on the part of the algorithm. This is an area where practical and theoretical interests align.

From a practical point of view, there are many real-world uses for an effective morphology learner, ranging from providing useful morphological resources for poorly studied languages that can be integrated in speech recognition software and document retrieval all the way to providing automatic morphological parsing of the new words that are springing up every day by the hundreds in medical, genetic, and chemical publications.[1] For computational linguists concerned with these problems, it makes great sense to explore both methods of unsupervised learning and methods of semisupervised learning, in which small amounts of humanly analyzed material are given to the learner as a good starting point in the process of learning.

The interest in unsupervised learning of morphology is perhaps even greater from a theoretical point of view, as researchers both in mainstream linguistics and in computational linguistics have converged on the belief that the most important question is how language, with all its richness and variability around the globe, can be learned by humans so rapidly. Morphology appears to many linguists—to us, and no doubt to most of the authors cited here—to be an ideal domain in which to study specific hypotheses related to grammar learning because, in most human languages, morphology is complex and therefore difficult to learn, and yet there seem to be perfectly sensible grounds for thinking that we can succeed in some significant ways, if not all ways, in learning morphology automatically.

What would we like an unsupervised morphology learner to accomplish? The learner should be able to read in a textual sample from any human language, either in a standard orthography or in a phonological transcription, and to develop a program, or a data structure, that would allow us to provide a morphological account of any word from that language. Ideally, it should be able to do that even if the word was not in the original sample on which the analysis was based. Linguists who work on morphology are divided on what it means to provide "a morphological account": Some morphologists expect an analysis of a word into component morphemes, but others dispute the existence of morphemes and prefer to provide a word- and paradigm-based account. We adopt an ecumenical perspective in this review, and therefore consider both approaches deserving of serious attention by those constructing automatic morphology learners. We return to this question in Section 6, below.

A successful morphological learner would provide answers to questions such as the following: What are the component morphemes of each word? Are there alternative forms (allomorphs) of any of the morphemes, and if so, under what conditions is each used? Are there alternative forms of morphemes that need to be explained by reference to phonological generalizations in the language? Are there inflectional paradigms in the language, and if so, how many independent dimensions (or morphosyntactic features) are active in each of the paradigms? What combinations

[1]The field of document retrieval contains many studies of methods to automatically extract the stem of an English word, so that documents that share common word stems but not words can be identified. Examples of these studies include Paice (1994) and Hull et al. (1996). Jacquemin (1997) presents a different approach to a similar problem.

of morphosyntactic feature specifications are permitted in the language, and how is each such combination realized morphologically? Are there processes of derivational morphology present in the language?[2] How productive is each of the processes discovered?[3] So far, most of the effort has been devoted to answering the first question, though the others are coming into focus as solutions to the segmentation problem get better.

1.2. Evaluating: Precision and Recall

Quantitative evaluation of computational methods of learning is important for determining success or failure, but surprising though it may be, when we try to determine what the correct morphological analysis of a word is, we find many more unclear cases than we might expect ahead of time (and in this respect, morphology is much more like syntax than we might have expected it to be). English has many borrowings, and many of the affixes of these borrowings have entered into our own morphology (as with suffixes such as *-ize*, *-ist*, *-ism*, and so on), but in many other cases, it is not clear whether the morphology has been integrated into English. Is the final *-es* of *Los Angeles* a suffix in the name of the city? Is *-i* a suffix in the word *alumni*? Here is a list of words that may leave us unsure about what should count as the right analysis: *boisterous*, *ambassador*, *annual*, *poem* (cf. *poet*), *agrarian*, *armor*, *benediction*, *crucial*, and *worn*. Not only do we not have a method to resolve what counts as the right answer in the unclear cases; we do not even have a method to determine what should count as an unclear case!

Measurements of precision and recall are widely used to quantitatively evaluate the results of morphology learning. These terms were originally developed in the context of document retrieval, which consists of a method to take a user's query—typically a set of words, or something of the sort—and retrieve from a library all of the documents that the user wants. The proportion of the documents that were returned that were in fact wanted by the user is the query's precision, and the proportion of those that were returned to all of those that should have been returned is the query's recall (Kent et al. 1955). A natural way to evaluate morphological analysis is to treat each position between letters (phonemes) as a site of a possible morpheme break; if we have a gold standard created by a human with an indication of the true segmentation, then we can evaluate which of the predicted breaks are true and which false, and we can do the same for positions for which breaks were not predicted.

An alternative approach is to evaluate the quality of a morphological learner's output on the basis of how much that analysis improves the results of a larger system in which it is included. An early example is given by Hafer & Weiss (1974), who used an information retrieval task in their empirical comparison of several variants of Zellig Harris's (1955) successor count method. Other commonly used tasks are speech recognition and statistical machine translation. In general, this practice can offer a convenient way of avoiding the difficulty of making explicit what counts as the right morphological analysis in unclear cases. In addition, several researchers aim not to predict where morpheme breaks are, but rather to predict which word forms are part of the same lexeme, and an appropriate evaluation measure must be established for that strategy.

[2]A number of linguists have made strong cases that the distinction between derivational and inflectional morphology is one that neither can nor should be maintained across languages. In the context of unsupervised learning of morphology, however, the distinction is useful.

[3]Characterizing the notion of productivity in morphology is no easy matter, and a formalization of the notion is even harder. A study of this topic can be found in O'Donnell et al. (2011) and, more recently, O'Donnell (2015); see also Snover et al. (2002). Indeed, any system that hopes to make predictions outside of the words observed in the training data is obliged to develop a hypothesis about which generalizations are productive and which are not.

Several papers provide very useful overviews of the preceding literature. Hammarström & Borin (2011) present an outstanding review, and we have profited greatly from it, and encourage the reader to turn there. Goldsmith (2001) discusses some of the earlier research in the field, and Goldsmith (2010) covers the related problem of word discovery in addition to morpheme discovery. Virpioja et al. (2011) provide a helpful discussion of the empirical evaluation of systems.

2. GENERAL CONSIDERATIONS

2.1. Zipfian Sparsity

Since the very first studies of word frequencies, linguists have noticed that in all languages, a small number of words have a high frequency, a modest number of words have an intermediate frequency, and a very surprisingly large number of words have a very low frequency (counts of one or two). Such distributions are often described as zipfian in honor of George Zipf (1935, 1949).

This distribution leads to a particularly striking problem for studies of learning, both studies involving learning algorithms and those involving children. When a lexical entry has a paradigm with dozens or scores of differently inflected entries, it is rare to find a form whose complete paradigm is attested—in fact, it never happens. Instead, the language learner is obliged (or, alternatively, eagerly committed) to finding morphological patterns shared by a large number of stems without finding many stems that illustrate the contrasts (which is to say, the entries) across each paradigm. Lignos & Yang (forthcoming) provide a recent study of the extent of this phenomenon. We return to this general problem in Section 7.1, below.

2.2. Searching Grammar Space for the Best Morphological Grammar

Most of the more successful research on this topic is based fundamentally on the metaphorical understanding that grammar learning consists of a search through grammar space, typically one small step at a time. That is, we can imagine the specification of a grammar as locating it as a point in a space of very high dimensionality, and the task of finding the correct grammar is conceived of as one of traveling through that space. Methods differ as to where in grammar space the search should start: Some methods assume that we start in a random location, whereas other methods allow us to start at a grammar that is reasonably close to the final solution. In this section, we briefly describe three approaches that have been used in this literature: minimum description length (MDL) analysis, Gibbs sampling, and adaptor grammars.

All of these approaches have been developed in the context of probabilistic models, and involve different aspects of a search algorithm through the space of possible grammars (here, morphologies) to find one or more grammars that score high on a test based on probability. Probability assigned to training data is used as a way to quantify the notion of "goodness of fit," in the sense that the higher the probability is that a grammar assigns to a set of data, the better the goodness of fit. The three approaches are not, strictly speaking, alternatives; one could adopt any subset of the three in implementing a system.

The essence of MDL analysis consists of dividing that probability into two factors, one the probability of the model and the other the probability of the data given the model. But MDL gives no insight into what a natural search method should consist of in the space of possible grammars, nor does it offer guidance on where the search should begin, or precisely how the search should proceed. Those decisions are left to the researchers and their particular implementations.

Gibbs sampling, by contrast, involves a specific style of searching in the space of grammars, and a probability is explicitly computed for the training corpus given each grammar that is explored,

but no constraint on how that probability distribution should be devised. This probability typically includes some consideration for grammar complexity (that is, the probability assigned to a corpus by Grammar 1 may be smaller than that assigned by Grammar 2 based solely on the larger number of parameters in use in Grammar 1), but it does not need to.

Adaptor grammars are models of grammar that keep track of counts of various previous decisions made in the generation of preceding utterances. They are built in such a way that "rich get richer" (i.e., zipfian) distributions arise naturally. Adaptor grammars have been implemented with Gibbs sampling as their method of choice for search.

2.2.1. Minimum description length. Several researchers proposed employing MDL analysis for learning grammars in the 1990s; these include Brent (1996) and de Marcken (1996) in connection with word discovery and Brent et al. (1995) and Goldsmith (2001) in connection with morphology learning. This approach was originally proposed by Jorma Rissanen in the 1970s (see Rissanen 1989, and for developments of these ideas in the context of linguistics, see de Marcken 1996 and Goldsmith 2015).

In its general form, MDL appeals to information theory, and proposes that the information content of a particular grammatical description of a particular set of data D can be calculated as the sum of two quantities: the complexity of the overall grammar G used to provide the description plus the number of bits needed to encode the data D, given G, a probabilistic grammar. The first term measures the complexity of the analysis by measuring its algorithmic complexity, and the second term measures the goodness of fit of the particular analysis of the data given the grammar. The second term can properly be understood as the quantity of information in the corpus that is not explained by the grammar. MDL instructs us to minimize the sum of these two quantities, both of which are measured in dimensionless bits.

MDL can be viewed as a way of quantifying the notion that, when we correctly understand it, we find that a language has done its very best to use and reuse its component pieces as much as possible: *C'est un système où tout se tient.* This is true for two distinct reasons: A grammar with fewer redundancies is preferred because removing redundancies leads to a shorter grammar, and reducing the number of alternatives permitted at each choice point in generating a word (or, more accurately, reducing the entropy at that choice point) increases on average the overall probability of the data. MDL's emphasis on finding the shortest grammar is what gives rise to the requirement to maximize edge savings and letter savings, as discussed below. MDL provides grounds for treating that not as a heuristic, but as a central and essential aspect of finding the optimal account. MDL's emphasis on maximizing the probability of the data given the model provides a satisfying account as to what it means for a grammar to fit the data and, in particular, why a grammar that generates too much provides no explanation. More importantly, perhaps, it provides a sound methodology that does not require negative evidence in order to support a given analysis. These notions are developed in detail by Goldsmith (2015).

2.2.2. Gibbs sampling. The central idea of Gibbs sampling is that we can profit from the fact that the grammar is a point in a space of high dimensionality; that each dimension corresponds to a small but significant property p_i; and that, much of the time, a meaningful local judgment can be made as to whether or not a change in the value of the parameter p_i is likely to contribute to the overall success of the grammar, if we fix all the other parameters. Gibbs sampling consists of a large number of iterations of a process by which we successively consider each of the parameters and, for each parameter, choose a value based on currently assumed values for all of the other parameters. If we iterate through each parameter once before returning to any parameters for a

second time, this is called a sweep.[4] The number of sweeps required may number in the thousands or more. In addition, we can incorporate simulated annealing into the search not by making the decision on each individual parameter choice deterministic (i.e., choosing the parameter choice that maximizes the objective function), but rather by using a logistic function incorporating a "temperature" to decide whether to change a parameter's value.[5]

Gibbs sampling can be applied to this sort of problem in different ways. Typically, the parameters are tied to the analysis of specific points in the data being analyzed. For example, if a corpus begins with the word *jumping*, and if parameter $p_3 = 1$ and $p_i = 0$ for all other values of i, then the model takes there to be a morpheme break after *jum* (i.e., after the third letter) and none after *jump*, whereas if both p_3 and p_4 are set to 1 and $p_i = 0$ for all values of i not equal to 3 or 4, then the word is broken into morphemes both after *jum* and after the *p*. Gibbs sampling, under such an implementation, would pass through all the parameters, each corresponding to a point between two specific letters in the corpus, calculating whether the hypothesis that a break occurs between these letters leads to a higher or a lower probability for the corpus than the hypothesis that there is no break there, given the probability model that flows from all of the other currently assumed word analyses assigned in the rest of the corpus. (This is the crucial point: The probabilities used in calculating the objective function's values at each moment depend completely on all of the other assumptions currently being made for the other data.) A large number of iterations through all possible points would be necessary to arrive at an optimal analysis.

Gibbs sampling does not in principle lead to a single optimal grammar; Gibbs sampling is the orienteering process, so to speak, by which a path through grammar space is undertaken, and it will visit grammar points with a probability equal to the probability of the grammar given the data. If there is a second grammar that assigns a probability that is equal to one-half of that assigned by the best grammar, then the second grammar will be visited by the Gibbs sampling half as often as the best grammar.

2.2.3. Adaptor grammars. Adaptor grammars have been developed in a number of papers, especially by Mark Johnson, Sharon Goldwater, and Thomas Griffiths (Johnson et al. 2007a,b; 2008).

An adaptor grammar is a generalization of a phrase-structure grammar, most easily described in the context of generation, as part of a statistical process. An adaptor grammar contains a memory cache to keep track of the number of times its nodes have been expanded in the previous generation of sentences, and an adaptor grammar contains a family of parameters that in effect recompute the probability of each rule in the grammar with each production, based on the cached counts. By design, only the counts are retained from preceding productions, and the order in which productions occurred plays no role.

These models have been explored by a number of researchers in recent years (Botha & Blunsom 2013, Kumar et al. 2015). They naturally generate output that is zipfian in interesting ways, and

[4]In particular, the value for the parameter is selected according to the marginal probability for that parameter, given the current values of all the other parameters.

[5]What this means in practice is this: Suppose the difference in the objective function between the choice of parameter p being on (i.e., has value 1) and being off (i.e., has value 0) is $d = f(p = 1|\text{all other parameters fixed}) - f(p = 0|\text{all other parameters fixed})$. We then switch parameter p to on with probability $\frac{1}{1+e^{-d/t}}$. Early in the learning process, we make t, the pseudo-temperature, large, so that the system is relatively free to move around in the search space even when the local hypothesis seems to be doing reasonably well. As the temperature lowers, the learner becomes more and more conservative, and ready to change parameter values based only on the result of the computation of the objective function.

Gibbs sampling can be used to guide the learning path from a randomly chosen initial hypothesis to one (or more) optimal morphology.

2.2.4. Moving intelligently through morphology space. Models of morphology learning can differ in two ways regarding their conception of arriving at the best analysis in a step-by-step manner: (*a*) They may differ in whether they begin at a randomly selected point, that is, a more or less randomly specified initial state, and (*b*) they may differ with regard to how domain-specific and intelligent the principles are that control the path taken by the grammar as it improves.

The primary issue seems to be whether the change in the grammar can be at a relatively high level during the search, or whether the changes remain relatively local, or at a low level. It is easy for a system analyzing English, for example, to fall into the erroneous analysis that assigns the suffixes *-an* and *-en* to such stems as *mailm-*, *policem-*, *salesm-*, *fisherm-*, and *garbagem-*. Shifting the stem–suffix break one letter to the left will not seem like a better analysis if only one or two of these words are reanalyzed, but overall the analysis is better if all the words are modified in one step. This suggests that in some cases (indeed, perhaps most cases) it is more effective to evaluate changes in the morphological grammar and their consequences over all of the forms as we move through morphology space. (This is the strategy adopted by Goldsmith 2006 and Linguistica 4.) Further discussion of intelligent search strategies can be found, for example, in Schone & Jurafsky (2000), Snover & Brent (2001, 2002), and Monson et al. (2004).

3. CONCATENATIVE MORPHOLOGY

3.1. The Problem of Segmentation

Almost all of the research discussed in this review assumes that each word or utterance of the basic data that are observed in a language can be adequately represented as a string of letters, where either the letters are drawn from the standard orthography of the language or they represent a broad phonetic (perhaps phonemic) transcription of the word. In the vast majority of languages, most words can be uncontroversially divided, or partitioned, into a sequence of morphs that do not overlap and that place all letters into exactly one morph (as when the word *prepublishing* is segmented into *pre-publish-ing*)—the case of Semitic languages being a notable counterexample, as discussed in Section 4, below. There is no upper limit on the number of morphs that may appear in a word. In simple terms, the problem is how to split each word up into appropriate, functional subparts.[6] This is the problem of morphological segmentation, and it is the problem that has seen the greatest effort spent on solving it.

If we were given the component meanings and grammatical functions of each word, and we could use that knowledge as we tried to split up each word, the task would be much easier. That is, if we were given the word *prepublishing* and we were informed that it has a tripartite meaning, involving the concept "prior in time," a grammatical function of "nominalization," and a root meaning "ACTION-INVOLVED-IN-PRINTING," and if we had similar information for all the words in our corpus (e.g., *publishes* and *preapprove*), our job would be much easier—it would not be trivial, but it would be much easier. And in some cases of real language learning, it may be realistic to assume that we learn a new word along with at least some syntactic/semantic information. But in general we do not assume that the learning mechanisms from other components of the grammar carry the heavy burden of doing the work and can therefore be called upon by the

[6]To our knowledge, no studies have attempted unsupervised learning of a signed language.

morphology-learning component, and for a very good methodological reason: Some component or components of the general language learning algorithm must be able to bootstrap the language learning process—that is, to get things started. We should not always count on some other component to provide learned structure. As we model each component, we should require of ourselves that we make the very smallest assumption possible about what other components have (so to speak) already inferred about the structure of the language being analyzed. Thus, whereas some parts of the literature do consider distributionally learned semantics as a feature (see Section 7.3, below), most of the work done in this area has (rightly, in our opinion) made no assumption that the learning algorithm has access to any information about the meaning or function of each word.

In other words, we adopt the working hypothesis that it is possible to solve at least some of the problem of morphological analysis without access to meaning (or knowledge of syntax). Given that virtually all functions in morphology are related to meaning in some way, it should be clear that the researcher is under no illusions that this morphological analysis is final or complete; a complete analysis will involve meaning. But the hope is that some aspects of language structure can be learned by reference to formal and sound-based properties of utterances.

3.1.1. Zellig Harris and the risk of being greedy. Let us consider the classic proposal of Harris (1955, 1967) for automatic morphological analysis first, because it was one of the very first to be published, and because it serves as a good point of comparison for other approaches that we discuss below.[7] Harris proposed not a single method but a family of closely related methods. His central idea was that, given a set of words, we scan through each word letter by letter, looking at an increasing word-initial string. After each such word-initial string, we ask how many distinct letters appear anywhere on the list among those words beginning with *S*, and we call this number *S*'s successor frequency (SF). For example, in one corpus we might find that the successor frequency of the word-initial string *govern* is six, because it is followed by the letters *e*, *i*, *m*, *o*, and *s* and the word-ending boundary marker #; we could write SF(*government*, 6) = 6, meaning that after the sixth letter of *government* there are six possible letter continuations. By contrast, the successor frequency of *gover* is only one (because only *n* follows *gover*), SF(*government*, 5) = 1; and the successor frequency of *governm* is only one (because only *e* follows *governm*), SF(*government*, 7) = 1. A mirror-image predecessor frequency can be defined as well. Harris believed that a judicious combination of conditions on successor and predecessor frequency would lead to an accurate discovery procedure for morphemes, such as cutting a word at a point *k* where SF(*word*, *k*) > SF(*word*, *k* − 1) and SF(*word*, *k*) > SF(*word*, *k* + 1), or where such a peak is found for either successor frequency or predecessor frequency.

Harris's general approach was evaluated by Hafer & Weiss (1974), who explored 15 different criteria for morpheme breaks that were consistent with the spirit of Harris's idea. These authors allowed parameters to be learned from the data (such as whether peaks of SF should be sought, or the particular values of the SF threshold above which SF marks a morpheme boundary); today we would say that they used the same data for training and for testing. But they ended up with relatively disappointing quantitative results nonetheless.

The principal lesson that we can learn from carefully studying why Harris's method does not work is this: We can identify an analysis of a language as correct only to the extent that we can see

[7] Hammarström & Borin (2011) discuss similar research by Andreev, published between 1959 and 1967, that has been little noted in the Western literature, though Flenner (1994) describes her development of Andreev's ideas. Other researchers explored algorithmic approaches to identifying affixes in particular languages, as Resnikoff & Dolby (1965, 1966) and Earl (1966) did in their studies of English; their interesting discussions of the problem constituted a sort of prototheory of the problem of language-independent morphology discovery.

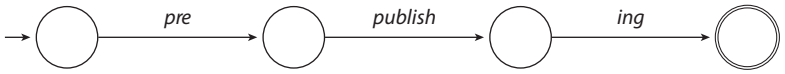

Figure 1

Representing a word's segmentation with a finite state automaton (FSA).

that the analysis proposed for one part of the language fits in as part of a larger whole. It is only the overall coherence of a grammar that provides the confirmation that we have found the right structure. For linguists, this should not be a surprise. This insight was already explicit in writings in the 1940s by linguists working within the circles around Hjelmslev and, ironically enough, Harris, and it was elevated to a central principle by Chomsky (1955). Greedy, local methods of analysis rarely work to understand complex cognitive functions. (We return to the notion of a greedy algorithm below.)

3.1.2. Finite state automata. Today we may say that the linguist's task of uncovering and displaying concatenative morphology in a language is essentially the task of finding a finite state automaton (FSA) in which edges are labeled with morphemes. In such a view, there is an equivalence between the set of all paths from the starting state to one of the final states (technically, accepting states), on the one hand, and all licit words in the language (a state corresponds to a node in a graphical representation), on the other.[8] A word that consists of a prefix, a stem, and a suffix would thus correspond to a path from the starting state through two intermediate states to a final state, and each of its morphemes would be a label of one of the edges of that path (**Figure 1**). There are many such FSAs for any set of words, so we must say explicitly that we seek the FSA that has the smallest number of edges. Indeed, in the best of all grammars, each morpheme in the language is associated with exactly one edge.[9]

Imagine that we begin with a word list from a language and build an FSA with only two nodes or states. Each word on the list is associated with a distinct edge running from the starting state to the single final state. Such an FSA would correctly analyze all and only the monomorphemic words of the language. What single state could we add that would most improve it? Ideally, if we knew that a large number of stems could be followed by exactly the same set of suffixes, we could add a node along with a set of edges to it from the starting state, where each edge is labeled by one of those stems; then we would add edges from that new state to the final state, where each of these new edges is labeled with one of the suffixes. We would then remove all of the unwanted edges that go straight from the starting state to the final state for those particular bimorphemic words (**Figure 2**). This insertion of a single node will greatly decrease the number of edges: If there are M stem edges coming into the new node and N suffix edges coming out of it, the number of edges that have been saved is $MN - (M + N)$. We might call this the edge savings that results from this modification of the FSA.[10]

From an algorithmic point of view, we can distinguish two kinds of approaches to finding such nodes that we might want to insert into the FSA. We can consider all possible places to put such

[8]On FSAs and morphology, see Sproat (1992), Beesley & Karttunen (2003), and Roark & Sproat (2007).

[9]What follows in the text, directly below and elsewhere, when not specifically attributed to other authors, is our opinion on the basis of the models that we have developed. In this section, we take a good deal of liberty in integrating observations from Goldsmith (2001, 2006). Executable code can be found at **http://linguistica.uchicago.edu**.

[10]An even better measure of savings tallies not the number of saved edges but rather the total number of letters associated with each of the saved edges. It is better to save multiple copies of edges associated with long strings than to save multiple copies of edges with short strings.

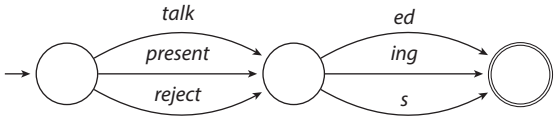

Figure 2

Representing a signature with a finite state automaton (FSA).

a node, and establish some threshold value for the edge savings above which we will insert the node. This is referred to as a "greedy" algorithm, one that makes local decisions but hopes to achieve global optimization. We might call its antithesis (a nongreedy approach) an "abstemious" approach, that is, one that considers all possible such nodes and inserts only the very best one, on the basis of its edge savings. The abstemious way is in virtually every respect a better way to go. If we apply this approach to a corpus of English, the top four edge-saving nodes that emerge correspond to stems followed by (*a*) the pair of suffixes *-s* and Ø; (*b*) the pair of suffixes *-'s* and Ø; (*c*) the pair of suffixes *-ly* and Ø; and (*d*) the set of suffixes *-ed*, *-ing*, *-s*, and Ø. Goldsmith (2001) calls these constructs signatures; they can be thought of as highly corpus-bound protoparadigms. Each signature is a set of stems followed by a set of suffixes, for which all pairs of stem plus suffix are found in the corpus.[11]

It is not difficult to find sets of suffixes that lead to signatures with high edge savings. The simplest approach is to look at all positions in a word where the successor frequency is greater than one and, for each such point, to gather the strings that follow the word-initial string, right up to the end of the word. We call these ending sets. We then determine for each of these ending sets precisely how many different word-initial strings lead up to them. The count of those different starting strings, and the count of the number of strings in the ending sets, gives us the edge savings (as those two numbers correspond directly to the M and N described above). A set of signatures derived in this way, each containing at least two word-initial strings (in effect, two stems), produces an interesting first approximation of the morphology of the final suffix of an inflecting language, and the larger the edge savings is, the more certain the analyses are.

The emphasis on signatures is motivated by the fact that languages produce many examples of pseudo-generalizations that appear only once or twice. Whereas the pattern *read*, *reads*, *reading*, with its signature Ø, *-s*, *-ing*, occurs frequently (and hence the stem *read-* is well motivated), this stem does not participate in a larger linguistic generalization that relates it to such words as *readily* or *readjust*. Suffixes are well motivated when they occur in signatures, and signatures are well motivated when they occur with many stems.

3.1.3. Additional concerns. Let us reflect on how such an approach might fail, however. If all the members of a set of suffixes begin with the same letter (or letters), they will be analyzed as part of the stem; we have observed corpora in which the analysis {*aborti*, *constructi*} + {*on*, *ve*} was derived. Such an error will appear along with a telltale result: a set of stems that all end in the same letter.

Morphophonology, and phonology reflected in orthography, will also lead this initial algorithm to incorrect results. The Brown corpus (Kučera & Francis 1967) has 39 pairs of words such as *affluent*, *affluence* that are analyzed as having suffixes *-t*, *-ce*. More strikingly, while there are about 170 stems such as *climb* and *creak* that occur with exactly the suffixes *-ed*, *-ing*, *-s*, and Ø, there are about 90 like *move*, which would be analyzed as having stems such as *mov-*, *embrac-*, and *silenc*

[11] Gaussier (1999) explores a similar perspective.

and the suffixes *-e*, *-ed*, *-ed*, and *-ing*. Of course, this allomorphy (loss of stem-final *e* before the suffixes *-ed* and *-ing*) no longer reflects spoken English, so this particular problem would not arise in dealing with a transcription of modern English. However, the problem illustrates what would arise in dealing even with transcribed Middle English, and many other cases.

Such an elementary analysis into stem and suffix (or its mirror image, the analysis into prefix and stem) must be followed by a more careful analysis to separate derivational morphology that is not fully productive. For example, the analysis into signatures will find large classes of stems (*pretend*, *contend*) that are associated with the suffix set {*ed*, *er*, *ing*, *s*, Ø} or the set {*ation*, *ed*, *er*, *ing*, *s*, Ø}, such as *confirm*. It is a very difficult computational problem to distinguish between those affixes that are productive and those that are not.[12] In this case, this means determining which of the stems that appear with the suffixes Ø, *-ed*, *-s*, and *-ing* can also appear with *-er* or *-ation*.[13]

3.2. From Stem and Paradigm Languages to Agglutinative Languages

Much of the research referenced above has focused on determining the appropriate morphological break between stem and suffix (and/or the break between prefix and stem). But even in Western European languages, it is not at all uncommon for a word to have several suffixes (e.g., *tranform-ation-less*, *fundament-al-ism-s*), and such languages as Finnish, Turkish, and Hungarian commonly have several affixes in a word. The number of morphemes per word is greater still in a good number of the languages in the rest of the world. How can the methods discussed here be extended to deal with agglutinative languages, with many morphemes per word?

Linguistica 4, the system described by Goldsmith (2006), can be used to apply the affix identification algorithm iteratively. Once a set of suffixes has been ascertained, a corresponding set of stems is identified; these stems are combined with those words left unanalyzed in the first iteration to form a new set of strings, and this set is analyzed on a second iteration. On a large corpus with the words *fundamental*, *fundamentally*, *fundamentalism*, and *fundamentals*, the system analyzed *-al*, *-ly*, and *-s* as suffixes on the first iteration; it analyzed *-ism* on the second iteration (during which it also identified *-al* as a suffix to the left of *-ly* and *-s*); and it identified *-al* as a suffix to the left of *-ism* on the third iteration.

The ParaMor system, described by Monson et al. (2007), achieves the induction of multiple word-internal morpheme boundaries by hypothesizing multiple stem–suffix divisions for a given word form. At the heart of the ParaMor algorithm is the search for schemes or partial paradigms, data structures with a set of suffixes associated with stems. Crucially, a word type can have multiple hypotheses of stem–suffix divisions. The Spanish word *administradas* 'administered (feminine, plural)' can be segmented as *administr-adas*, *administra-das*, *administrad-as*, *or administrada-s* (for *-adas*, *-das*, *-as*, or *-s* in different schemes), with the final inferred segmentation as *administr-a-d-a-s*.

The Morfessor model family (summarized by Creutz & Lagus 2007) is designed for unsupervised morpheme segmentation of highly inflecting and compounding languages. Initially, Creutz & Lagus (2002) proposed two search algorithms. The first considers each word in a corpus successively, evaluates each possible split into two parts by using an MDL-based cost function, and recursively processes the resulting parts until no further gain can be obtained. The second method starts

[12] This problem is related to the challenge an algorithmic learner is faced with when a suffix is rare, addressed directly by Desai et al. (2014) working on Konkani (India); see also Lignos & Yang (forthcoming).

[13] Truncation has become an important morphological process in virtually all European languages, as when *stylographe* is truncated in French to *stylo*. Pham & Lee (2014) select the truncation site in Brazilian Portuguese as involving a balance between deleting as much as possible and preserving as much as possible, inspired by successor and predecessor frequencies in Harris's work.

with breaks at random intervals and uses an expectation-maximization (EM) algorithm (Dempster et al. 1977): It iteratively estimates morph probabilities based on the current segmentation of the data, then uses the estimated distribution to resegment the data in a way that maximizes the probability the model assigns to them. More recent versions of Morfessor improve segmentation results by incorporating knowledge of morph categories (e.g., prefix, suffix, stem) into the model.

Linguistica 5 uses a similar method as Linguistica 4 for finding the rightmost suffix (or leftmost prefix), but uses a different method to find additional affixes closer to the root. It uses a local measure of robustness to measure the plausibility of a morpheme hypothesis; robustness is defined as the length of the morpheme multiplied by the number of times it appears in distinct cases. Thus, for example, after finding a large set of words that appear both with and without a suffix *-ly* in English, it inspects the resulting stem set and looks for the stem-final string with the greatest robustness, generating the FSA shown in **Figure 3**. It uses an abstemious strategy, as explained above, choosing to discover 100 internal suffixes (suffixes preceding other suffixes) across the entire FSA of English.

4. NONCONCATENATIVE MORPHOLOGY

Morph concatenation is by far the most frequent word formation mechanism in languages around the world, and it is no surprise that a vast majority of the research on morphology learning has specifically addressed it. This kind of analysis assumes that morphs consist of concatenated (hence adjacent) segments, just as a word consists of concatenated (hence adjacent) morphs. Yet an important class of productive morphological phenomena cannot be conveniently expressed in terms of operations bearing on contiguous strings. Thus, in Semitic languages word stems are typically formed by intercalation rather than by concatenation, as illustrated by such pairs as Arabic /kalb/ 'dog' ~ /kilaab/ 'dogs' and /raml/ 'sand' ~ /rimaal/ 'sands.' Traditionally, such observations have been accounted for by positing the existence of roots /klb/ and /rml/, that is, morphs consisting of a sequence of consonants; conveying the lexical meaning of the word; and combining with various patterns of vowel qualities and quantities that express inflectional or derivational variations. In such an analysis, the relative order of segments is preserved between the morph and the word, but adjacency is not: Segments that are adjacent in the morph may be nonadjacent in the word. We say that the sequence /klb/ is maintained in the word /kalb/ but that the string /klb/ is not (cf. Lee 2015). Ablaut in English strong-verb inflection is another well-known example of a nonconcatenative process, albeit by no means as productive as the root-and-pattern morphology in Semitic languages.

Approaches to the unsupervised learning of nonconcatenative morphology have been applied mostly to Arabic, in particular the classical and modern standard written forms (often vowelized). Aside from the early research of De Roeck & Al-Fares (2000), who seek to find clusters of morphologically related words, identifying the root of a word is the problem that all approaches reviewed here have attempted to solve. Some of the proposed algorithms (Bati 2002; Rodrigues & Ćavar 2007, 2005; Clark 2007; Xanthos 2008) aim to leverage this result and provide paradigmatic accounts of root-and-pattern morphology.

One of the first learning methods used in this context comes from the field of information retrieval. Indeed, in order to cluster words that share the same root, De Roeck & Al-Fares (2000) use a string similarity coefficient initially designed by Adamson & Boreham (1974) for identifying semantically related documents. This coefficient relies on a representation of strings as bags of letter *n*-grams and essentially quantifies the degree of overlap between these *n*-gram distributions. Finding that the original method does not successfully handle Arabic data, De Roeck & Al-Fares

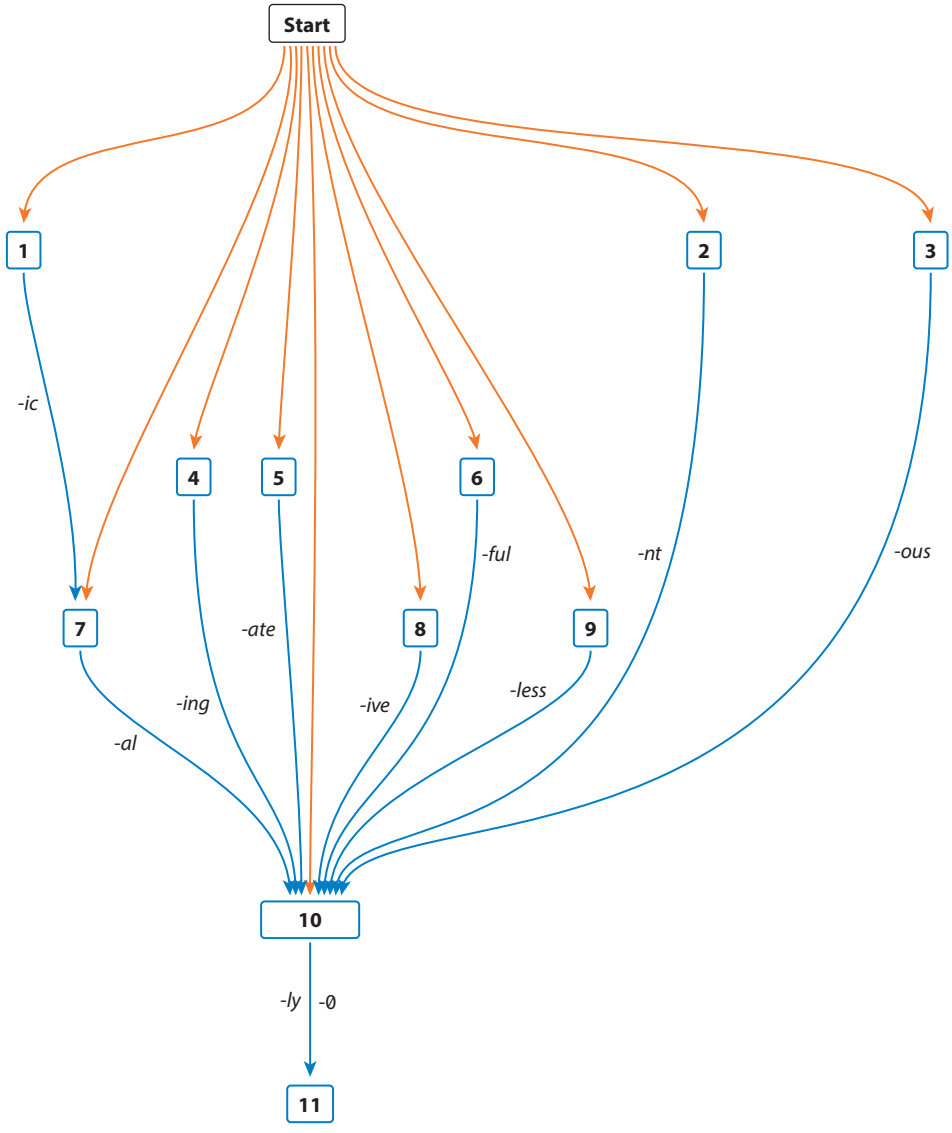

Figure 3

Finding multiple suffixes in Linguistica 5. The orange lines indicate sets of stems; the blue lines represent suffixes.

(2000) propose various ways of adapting it, mostly in the sense of including hard-coded, language-specific knowledge, such as phonological biases[14] and affix inventories.

In a recent contribution to this line of research, Khaliq & Carroll (2013b) obtain good results on Arabic root identification without recourse to such supervision. Their approach builds on the work

[14]In particular, weak consonants (glides and glottal stop) are processed in a distinct fashion.

of De Pauw & Wagacha (2007), who use machine learning techniques to find relationships between morphologically related words. In particular, De Pauw & Wagacha (2007) adopt a representation of each word in terms of all the *n*-grams that occur in initial position (e.g., for *word*: *w*, *wo*, *wor*, and *word*), internal position (*o*, *r*, and *rd*), and final position (*d*, *rd*, *ord*, and *word*); each such representation is then treated as an instance of a class that is taken to be the word itself. A maximum entropy classifier is trained on the basis of these data and used to reclassify them, which amounts to clustering words whose representation is most similar, and that are thus potentially morphologically related. Khaliq & Carroll (2013b) show that this approach can be readily adapted to root-and-pattern morphology by defining features as subsequences of (possibly nonadjacent) letters rather than contiguous substrings. In a follow-up to this study, Khaliq & Carroll (2013a) present a conceptually simpler yet similarly efficient method, based on the principle of contrastive scoring, whereby hypothetical roots are iteratively scored in proportion of their tendency to co-occur with frequent patterns, and vice versa.

Some learning heuristics rely on specific properties of root-and-pattern morphology. Thus, Elghamry (2004), followed by Rodrigues & Ćavar (2005, 2007), sets explicit constraints on the maximum distance between letters forming a triliteral Arabic root. The algorithm determines how often each letter type occurs in subsequences that either satisfy or do not satisfy these constraints in a corpus, and integrates these counts to select the most likely root for each word. Xanthos (2008) describes several techniques for learning the consonant–vowel distinction[15] in an unsupervised fashion and uses the result to decompose Arabic words into a consonantic root and a vocalic pattern.[16] Such approaches are bound to make spurious inferences when applied to languages without root-and-pattern morphology, which in a truly language-independent setting underscores the importance of evaluating the global relevance of nonconcatenative morphology learning for a given corpus. Xanthos does so by quantifying the compression resulting from modeling the data with a root-and-pattern analysis, which was orders of magnitudes larger for Arabic than for English or French, for instance.

The latest proposals in this area, by Fullwood & O'Donnell (2013) and Botha & Blunsom (2013), adopt the nonparametric Bayesian framework of adaptor grammars (see Section 2.2.3) pioneered by Johnson et al. (2007a). Interestingly, these papers also have in common that they simultaneously deal with both nonconcatenative and concatenative aspects of Semitic morphology, but they do so in very different ways. Fullwood & O'Donnell (2013) represent affixes on the same level as vocalic patterns, so that an Arabic form like /zawjah/ 'wife' (where /-ah/ is usually thought of as a feminine suffix) is decomposed into root /zwj/ and "residue" /aah/, and their intercalation is described with the template *r* – *r r* – –, where *r* stands for a root consonant and – for a residue component. Botha & Blunsom (2013), by contrast, use the range concatenating grammar formalism (Boullier 2000) to represent concatenation and intercalation operations in a distinct but unified fashion, thereby helping to solve what is arguably one of the greatest challenges in the field.

5. ALLOMORPHY AND MORPHOPHONOLOGY

The problem of dividing a word into its component morphs (or morphemes) is directly connected with the problems of allomorphy and morphophonology.

[15] See also Goldsmith & Xanthos (2009) for a review.

[16] Bati (2002) represents a precursor to this research, although phonological categories were hard-coded in this case.

Paradoxically, it appears that the learning of allomorphy and morphophonology must both precede and follow the learning of morphological segmentation. On the one hand, if we already have knowledge that [e] and [ie] are closely related in Spanish morphology, and that two morphemes differ only by that string-wise difference, then a learning algorithm could without difficulty construct and test the (morphological) hypothesis that *ten-er* and *ten-emos* are in the same relationship to *tien-e* as *sab-er* and *sab-emos* are to *sab-e*. But on the other hand, learning the close relationship between [e] and [ie] in Spanish (learning morphophonology) is most easily accomplished if we know that there are a large number of verbal lexemes in Spanish whose paradigms contain pairs of stem morphemes that are identical except that one has an [e] where the other has an [ie] (which assumes knowledge of morphological structure). Intermediate positions are imaginable, to be sure: With no knowledge of morphophonology, an automatic learner can build incomplete paradigms, one for each version of the stem (in this case, the stem *ten-* and the stem *tien-*), and these paradigms will be much less complete than that built for the more regular stem *sab-*. That scenario imagines some morphological analysis being followed by some phonological analysis, which in turn can be used by the morphological learner to extend and simplify the overall morphology.

It seems to us that the overall resolution of this apparent paradox is that there is no prior ordering of components that can be established for the unsupervised learner of language, and that each component must look for what is often called low-hanging fruit—that is, complexities that can be identified after relatively little learning has taken place. In some cases, a morphophonological regularity will be learned quickly, after only a handful of the morphology has been inferred, whereas in other cases it may take a considerable amount of morphological analysis before the morphophonological generalization emerges.

Zhang & Kim (1990) present some early research on learning rules of allomorphy; Gaussier (1999) describes some additional work on this topic. Goldwater & Johnson (2004) take the induced morphological signatures from Goldsmith's Linguistica as a starting point and propose a Bayesian approach to learning morphophonological transformation rules, a direction also described by Goldsmith (2006). As an example of rule learning, the signature Ø, *-ed*, *-er*, *-ing* for stems such as *work* and *roll* can be related to the similar signature *-e*, *-ed*, *-er*, *-ing* for *din* and *bik* by hypothesizing an *e*-deletion rule for the stems of the latter signature in the contexts of suffixal *-ed*, *-er*, and *-ing*. See also Schone & Jurafsky (2001) and Wicentowski (2002, 2004). This area appears to be ripe for additional progress.

6. PARADIGMS

In many languages, inflectional paradigms are traditionally partitioned into distinct inflection classes (conjugation classes for verbs, declension classes for nouns and adjectives) according to how similarly the lexemes inflect. The notion of inflection classes has attracted attention from researchers who ask if inflection classes can be learned from a given set of paradigms. Goldsmith & O'Brien (2006) model inflectional patterns using a connectionist approach, in which the nodes in the hidden layer correspond to the more abstract inflection classes. More recent research treats inflection class inference as a clustering problem in unsupervised learning (Zeman 2009, Brown & Evans 2012, Lee 2014, Beniamine & Sagot 2015). Apart from the particular clustering algorithms being used, proposals differ in whether inflection classes are in a flat or hierarchical structure. In the case of a flat structure, inflectional paradigms precategorized in distinct inflection classes can act as a gold standard data set. Linguists often provide a hierarchical account of the paradigms found in a language, but it is difficult to translate those proposals into a format that can serve as a veritable gold standard for evaluation. Nonetheless, a hierarchical view of inflection classes offers insights into the partial similarities and differences across morphological paradigms.

7. OTHER CONSIDERATIONS

7.1. Language Acquisition by Children

Unsupervised learning is of great interest to linguists and cognitive scientists because it closely resembles the learning situation faced with humans acquiring their first language. A child acquiring English would not know at birth that *-ing* is a morph, and must learn it based on the linguistic input. Lignos & Yang (forthcoming) provide an overview of the morphological learning problem in language acquisition, covering issues of data sparsity, productivity, and analogy.

Most published research in computational morphology does not directly address the problem of human morphological acquisition, because the data sets used are mostly raw corpus text from adult language that is very much unlike child-directed speech, and because a batch learning algorithm, as opposed to incremental learning for data of increasing sizes, is used. Some recent research, however, does use child-directed speech; an example is Frank et al. (2013), who also make use of syntactic information, though they do batch learning. Lee & Goldsmith (2016) present preliminary results of incremental morphological learning using child-directed speech.

7.2. Word Similarity Without Morphemes

As noted above, not all analyses of words are based on the assumption that words are analyzable into morphs or morphemes, and computational analyses of word relationships without morphemes have been attempted. An early example is research by Adamson & Boreham (1974), who cluster words on the basis of the bigrams they contain (see Section 4). Other systems have used string-edit distance as a method for determining similarity between strings, as Baroni et al. (2002) do; similarly, some systems have used the length of the longest shared substring as a measure of similarity (Jacquemin 1997 does the latter, focusing on longest shared initial substrings; see also, for instance, Mayfield & McNamee 2003). Such methods, which do not directly attack the problem of morpheme discovery within words, often focus on distributional information above the word level, which can be syntactic, semantic, or—most commonly—some combination of the two.

7.3. Joint Learning

In principle, linguistic knowledge at multiple levels of grammar can be learned simultaneously, and it is reasonable to ask if such areas of knowledge from different levels may interact with or even improve one another. In practice, not only does this approach lead to fruitful results for the computational tasks at hand, but it also provides important insights into theoretical questions, such as those connected with the architecture of grammar. Salient examples of this trend in the broader context of computational linguistics include Singla & Domingos (2006), Poon & Domingos (2007), and McCallum (2009).

For morphology in the context of unsupervised learning, the intuition is that knowledge akin to syntax that could be induced from a raw text ought to improve results in morphological learning, and vice versa. Higgins (2002) combines unsupervised morphological induction with the task of part-of-speech induction, couched within frameworks in theoretical linguistics for a parallel architecture of grammar. More recently, Dreyer & Eisner (2011), Lee et al. (2011), Sirts & Alumäe (2012), and Frank et al. (2013) have shown that learning morphology and syntax simultaneously does improve results for both components. An early semantics-based approach is that of Schone & Jurafsky (2000), who employ latent semantic indexing (LSA) on a set of documents to help determine whether two distinct words should be treated as morphologically related, on the

reasonable assumption that pairs of semantically related words with shared substantive roots (i.e., from the same lexeme) should appear much more often in a document than they would by chance (on LSA, see Deerwester et al. 1990). Other notable examples of such research are Baroni et al. (2002) and Neuvel & Fulop (2002), as well as, more recently, Soricut & Och (2015). Additional references can be found in Hammarström & Borin (2011).

7.4. Supervised and Semisupervised Learning

There has been a great deal of research on supervised and semisupervised learning of morphology. A factor that facilitates such research appears to be the increased availability of machine-readable inflection tables. Durrett & DeNero (2013) employ inflectional data from Wiktionary for supervised morphological learning. Other authors, such as Wicentowski (2004) and Ahlberg et al. (2014), use similar resources along with large corpus texts for semisupervised learning tasks. The system created by Yarowsky & Wicentowski (2000) does not require inflection pairs or tables, but assumes minimal knowledge of root words as well as mapping between parts of speech and expected morphological patterns. Ruokolainen et al. (2016) provide a review comparing unsupervised, supervised, and semisupervised approaches to morphological segmentation. As compiled and annotated data sets of morphological paradigms—even for low-resource languages—become more easily available, the semisupervised learning research paradigm, with highly competitive results, is likely to become more active in the years to come.

8. CONCLUSIONS

Reviewing the research of the past 20 years, we can observe a good deal of success with the problem of word segmentation and the discovery of word-internal structure. We can make two generalizations. The first is that the successes we see would not have been possible without the emergence of machine learning in the last 30 years. The tools developed there have been absolutely essential for the research described here. The second is a bit less obvious, but significant. All of the successful methods develop an explicit objective function that is based on characterizing a grammar and integrating a finite set of data, and then selecting a solution as the *argmin* winner: The learned grammar is the one that minimizes the objective function.

Although that view of "learning as computation of *argmin*" sounds like something that would come from machine learning, it also resonates with some traditions that are strictly internal to linguistics, most notably Chomsky's view of generative grammar, before the view from the 1970s that he called Principles and Parameters. In the earlier view, grammar selection was modeled as the task of finding the shortest grammar from among permissible grammars that generate the training data. "Learning as *argmin*" is not a natural perspective from the point of view of, for example, Optimality Theory, despite its name, nor from the point of view of more familiar mainstream models of grammar, where advantages are generally presented as being based on a descriptive range that is great enough to model the complexities found in well-studied languages. That is, of course, an admirable goal and way of evaluating a theory of morphology, or grammar more generally, and linguists must always be engaged in that activity; languages thrive on complexities that seem mysterious until linguists crack them open with new analytic techniques. But that style of developing morphology does not appear to have a natural hook, so to speak, to methods of inducing morphology from data.

From a practical point of view, we need to better understand exactly how well our current methods of morpheme segmentation work, based on some reliable measurements in several dozen languages. In addition, we need to address the challenge of learning the morphosyntactic features

that organize both the inflectional morphology and the interface between syntax and morphology. Current and recent research on category induction will help with this task, just as methods of induction of rules of morphophonology will help provide simpler computational models of morphology per se.

DISCLOSURE STATEMENT

The authors are not aware of any affiliations, memberships, funding, or financial holdings that might be perceived as affecting the objectivity of this review.

LITERATURE CITED

Adamson GW, Boreham J. 1974. The use of an association measure based on character structure to identify semantically related pairs of words and document titles. *Inf. Storage Retr.* 10:253–60

Ahlberg M, Hulden M, Forsberg M. 2014. Semi-supervised learning of morphological paradigms and lexicons. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 567–78. Gothenburg, Swed.: Assoc. Comput. Linguist.

Baroni M, Matiasek J, Trost H. 2002. Unsupervised discovery of morphologically related words based on orthographic and semantic similarity. In *Proceedings of the ACL-02 Workshop on Morphological and Phonological Learning*, pp. 48–57. Stroudsburg, PA: Assoc. Comput. Linguist.

Bati TB. 2002. *Automatic morphological analyzer for Amharic: an experiment employing unsupervised learning and autosegmental analysis approach*. MS thesis, Dep. Inf. Sci., Addis Ababa Univ., Ethiop. 90 pp.

Beesley KR, Karttunen L. 2003. *Finite State Morphology*. Stanford, CA: Cent. Study Lang. Inf.

Beniamine S, Sagot B. 2015. *Segmentation strategies for inflection class inference*. Presented at Décembrettes 9: Colloq. Int. Morphol., Toulouse, France

Botha JA, Blunsom P. 2013. Adaptor grammars for learning non-concatenative morphology. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing* (*EMNLP13*), pp. 345–56. Stroudsburg, PA: Assoc. Comput. Linguist.

Boullier P. 2000. Range concatenation grammars. In *Proceedings of the Sixth International Workshop on Parsing Technologies (IWPT 2000)*, pp. 53–64. Trento, Italy: Inst. Sci. Technol. Res.

Brent MR. 1996. Advances in the computational study of language acquisition. *Cognition* 61:1–38

Brent MR, Murthy SK, Lundberg A. 1995. Discovering morphemic suffixes: a case study in minimum description length induction. In *Proceedings of the 5th International Workshop on Artificial Intelligence and Statistics*, ed. D Fisher, H-J Lenz, pp. 264–71. New York: Springer

Brown D, Evans R. 2012. Morphological complexity and unsupervised learning: validating Russian inflectional classes using high frequency data. In *Current Issues in Morphological Theory: (Ir)regularity, Analogy and Frequency*, ed. F Kiefer, M Ladányi, P Siptár, pp. 135–62. Amsterdam/Philadelphia: Benjamins

Clark A. 2007. Supervised and unsupervised learning of Arabic morphology. In *Text, Speech and Language Technology*, vol. 38: *Arabic Computational Morphology*, ed. A Soudi, A Bosch, G Neumann, pp. 181–200. Amsterdam: Springer

Creutz M, Lagus K. 2002. Unsupervised discovery of morphemes. In *Proceedings of the ACL-02 Workshop on Morphological and Phonological Learning*, pp. 21–30. Stroudsburg, PA: Assoc. Comput. Linguist.

Creutz M, Lagus K. 2007. Unsupervised models for morpheme segmentation and morphology learning. *ACM Trans. Speech Lang. Process.* 4:3

de Marcken CG. 1996. *Unsupervised language acquisition*. PhD thesis, Dep. Comput. Sci., MIT, Cambridge, MA.

De Pauw G, Wagacha PW. 2007. Bootstrapping morphological analysis of G kũyũ using unsupervised maximum entropy learning. In *Proceedings of the 8th Annual Conference of the International Speech Communication Association* (*INTERSPEECH 2007*), pp. 1449–52. Red Hook, NY: Curran

De Roeck A, Al-Fares W. 2000. A morphologically sensitive clustering algorithm for identifying Arabic roots. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pp. 199–206. Stroudsburg, PA: Assoc. Comput. Linguist.

Deerwester S, Dumais ST, Furnas GW, Landauer TK, Harshman R. 1990. Indexing by latent semantic analysis. *J. Am. Soc. Inf. Sci.* 41:391–407

Dempster AP, Laird NM, Rubin DB. 1977. Maximum likelihood from incomplete data via the EM algorithm. *J. R. Stat. Soc. B* 39:1–38

Desai S, Pawar J, Bhattacharyya P. 2014. A framework for learning morphology using suffix association matrix. In *Proceedings of the 5th Workshop on South and Southeast Asian Natural Language Processing* (*WSSANLP 2014*), pp. 28–36. Red Hook, NY: Curran

Dreyer M, Eisner J. 2011. Discovering morphological paradigms from plain text using a Dirichlet process mixture model. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing* (*EMNLP11*), pp. 616–27. Stroudsburg, PA: Assoc. Comput. Linguist.

Durrett G, DeNero J. 2013. Supervised learning of complete morphological paradigms. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1185–95. Stroudsburg, PA: Assoc. Comput. Linguist.

Earl LL. 1966. Structural definition of affixes from multisyllable words. *Mech. Transl. Comput. Linguist.* 9:34–37

Elghamry K. 2004. A constraint-based algorithm for the identification of Arabic roots. In *Proceedings of the 1st Midwest Computational Linguistics Colloquium*. Bloomington: Indiana Univ.

Flenner G. 1994. Ein quantitatives Morphsegmentierungssystem für spanische Wortformen. *Comput. Linguist.* 2:31–62

Frank S, Keller F, Goldwater S. 2013. Exploring the utility of joint morphological and syntactic learning from child-directed speech. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing* (*EMNLP13*), pp. 30–41. Stroudsburg, PA: Assoc. Comput. Linguist.

Fullwood MA, O'Donnell TJ. 2013. Learning non-concatenative morphology. In *Proceedings of the 4th Annual Workshop on Cognitive Modeling and Computational Linguistics*, pp. 21–27. Sofia, Bulg.: Assoc. Comput. Linguist.

Gaussier É. 1999. Unsupervised learning of derivational morphology from inflectional lexicons. In *Proceedings of the ACL'99 Workshop: Unsupervised Learning in Natural Language Processing*, pp. 24–30. Stroudsburg, PA: Assoc. Comput. Linguist.

Goldsmith JA. 2001. Unsupervised learning of the morphology of a natural language. *Comput. Linguist.* 27:153–98

Goldsmith JA. 2006. An algorithm for the unsupervised learning of morphology. *Nat. Lang. Eng.* 12:353–71

Goldsmith JA. 2010. Segmentation and morphology. In *The Handbook of Computational Linguistics and Natural Language Processing*, pp. 364–93. New York: Wiley-Blackwell

Goldsmith JA. 2015. Towards a new empiricism for linguistics. In *Empiricism and Language Learnability*, ed. A Clark, A Perfors, JA Goldsmith, N Chater, pp. 58–105. Oxford, UK: Oxford Univ. Press

Goldsmith JA, O'Brien J. 2006. Learning inflectional classes. *Lang. Learn. Dev.* 2:219–50

Goldsmith JA, Xanthos A. 2009. Learning phonological categories. *Language* 85:4–38

Goldwater S, Johnson M. 2004. Priors in Bayesian learning of phonological rules. In *Proceedings of the 7th Meeting of the ACL Special Interest Group in Computational Phonology* (*SIGPHON04*), pp. 35–42. Stroudsburg, PA: Assoc. Comput. Linguist.

Hafer MA, Weiss SF. 1974. Word segmentation by letter successor varieties. *Inf. Storage Retr.* 10:371–85

Hammarström H, Borin L. 2011. Unsupervised learning of morphology. *Comput. Linguist.* 37:309–50

Harris ZS. 1955. From phoneme to morpheme. *Language* 31:190–222

Harris ZS. 1967. Morpheme boundaries within words: report on a computer test. *Transf. Discourse Anal. Pap.* 73. 24 pp.

Higgins D. 2002. *A multi-modular approach to model selection in statistical natural language processing*. PhD thesis, Dep. Linguist., Univ. Chicago. 208 pp.

Hull DA, et al. 1996. Stemming algorithms: a case study for detailed evaluation. *J. Assoc. Inf. Sci. Technol.* 47:70–84

Jacquemin C. 1997. Guessing morphology from terms and corpora. In *Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ed. NJ Belkin, AD Narasimhalu, P Willett, W Hersh, F Can, E Voorhees, pp. 156–67. New York: ACM

Johnson M. 2008. Using adaptor grammars to identify synergies in the unsupervised acquisition of linguistic structures. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics*, pp. 399–406. Stroudsburg, PA: Assoc. Comput. Linguist.

Johnson M, Griffiths TL, Goldwater S. 2007a. Adaptor grammars: a framework for specifying compositional nonparametric Bayesian models. In *Advances in Neural Information Processing Systems 19: Proceedings of the 2006 Conference*, ed. B Schölkopf, J Platt, T Hoffman, pp. 641–48. Cambridge, MA: MIT Press

Johnson M, Griffiths TL, Goldwater S. 2007b. Bayesian inference for PCFGs via Markov chain Monte Carlo. In *Human Language Technologies 2007: Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 139–46. Rochester, NY: Assoc. Comput. Linguist.

Kent A, Berry MM, Luehrs FU, Perry JW. 1955. Machine literature searching. VIII. Operational criteria for designing information retrieval systems. *Am. Doc.* 6:93–101

Khaliq B, Carroll J. 2013a. Induction of root and pattern lexicon for unsupervised morphological analysis of Arabic. In *Proceedings of the 6th International Joint Conference on Natural Language Processing*, pp. 1012–16. Nagoya, Jpn.: Asian Fed. Nat. Lang. Proc.

Khaliq B, Carroll J. 2013b. Unsupervised induction of Arabic root and pattern lexicons using machine learning. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing* (*RANLP 2013*), pp. 350–56. Hissar, Bulg.: Incoma

Kučera H, Francis WN. 1967. *Computational Analysis of Present-Day American English*. Providence, RI: Brown Univ. Press. **http://clu.uni.no/icame/manuals/BROWN/INDEX.htm**

Kumar A, Padró L, Oliver A. 2015. Learning agglutinative morphology of Indian languages with linguistically motivated adaptor grammars. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing* (*RANLP 2015*), pp. 307–12. Stroudsburg, PA: Assoc. Comput. Linguist.

Lee JL. 2014. *Automatic morphological alignment and clustering*. Tech. rep. TR-2014-07, Dep. Comput. Sci., Univ. Chicago

Lee JL. 2015. Morphological paradigms: computational structure and unsupervised learning. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pp. 161–67. Stroudsburg, PA: Assoc. Comput. Linguist.

Lee JL, Goldsmith JA. 2016. *Linguistica 5: unsupervised learning of linguistic structure*. Presented at Conf. N. Am. Chapter Assoc. Comput. Linguist, San Diego

Lee YK, Haghighi A, Barzilay R. 2011. Modeling syntactic context improves morphological segmentation. In *Proceedings of the 15th Conference on Computational Natural Language Learning*, pp. 1–9. Stroudsburg, PA: Assoc. Comput. Linguist.

Lignos C, Yang C. Morphology and language acquisition. In *Cambridge Handbook of Morphology*. Cambridge, UK: Cambridge Univ. Press

Mayfield J, McNamee P. 2003. Single *n*-gram stemming. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (*SIGIR03*), pp. 415–16. New York: ACM

McCallum A. 2009. Joint inference for natural language processing. In *Proceedings of the 13th Conference on Computational Natural Language Learning* (*CoNLL*), p. 1. Stroudsburg, PA: Assoc. Comput. Linguist. **http://www.aclweb.org/anthology/W09-1101**

Monson C, Carbonell J, Lavie A, Levin L. 2007. ParaMor: finding paradigms across morphology. In *Advances in Multilingual and Multimodal Information Retrieval*, pp. 900–7. Berlin: Springer

Monson C, Lavie A, Carbonell J, Levin L. 2004. Unsupervised induction of natural language morphology inflection classes. In *Proceedings of the 7th Meeting of the ACL Special Interest Group in Computational Phonology: Current Themes in Computational Phonology and Morphology* (*SIGMorPhon '04*), pp. 52–61. Stroudsburg, PA: Assoc. Comput. Linguist.

Neuvel S, Fulop SA. 2002. Unsupervised learning of morphology without morphemes. In *Proceedings of the 6th Workshop of the ACL Special Interest Group in Computational Phonology* (*SIGPHON02*), pp. 31–40. Stroudsburg, PA: Assoc. Comput. Linguist.

O'Donnell TJ, Snedeker J, Tenenbaum JB, Goodman ND. 2011. Productivity and reuse in language. In *Proceedings of the 33rd Annual Meeting of the Cognitive Science Society*, ed. L Carlson, C Hoelscher, TF Shipley, pp. 1613–18. Austin, TX: Cogn. Sci. Soc.

O'Donnell TJ. 2015. *Productivity and Reuse in Language: A Theory of Linguistic Computation and Storage*. Cambridge, MA: MIT Press

Paice CD. 1994. An evaluation method for stemming algorithms. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 42–50. New York: Springer

Pham M, Lee JL. 2014. *Combining successor and predecessor frequencies to model truncation in Brazilian Portuguese*. Tech. rep. TR-2014-15, Dep. Comput. Sci., Univ. Chicago

Poon H, Domingos P. 2007. Joint inference in information extraction. In *Proceedings of the 22nd National Conference on Artificial Intelligence* (*AAAI07*), pp. 913–18. Palo Alto, CA: AAAI

Resnikoff HL, Dolby JL. 1965. The nature of affixing in written English. *Mech. Transl. Comput. Linguist.* 8:84–89

Resnikoff HL, Dolby JL. 1966. The nature of affixing in written English, part II. *Mech. Transl. Comput. Linguist.* 9:23–33

Rissanen J. 1989. *Stochastic Complexity in Statistical Inquiry*. Singapore: World Sci.

Roark B, Sproat RW. 2007. *Computational Approaches to Morphology and Syntax*. Oxford, UK: Oxford Univ. Press

Rodrigues P, Ćavar D. 2005. Learning Arabic morphology using information theory. In *Proceedings of the 41st Annual Meeting of the Chicago Linguistics Society*, pp. 49–60. Chicago: Chicago Univ.

Rodrigues P, Ćavar D. 2007. Learning Arabic morphology using statistical constraint-satisfaction models. In *Perspectives on Arabic Linguistics 19: Papers from the 19th Annual Symposium on Arabic Linguistics*, ed. E Benmamoun, pp. 63–75. Amsterdam/Philadelphia: Benjamins

Ruokolainen T, Kohonen O, Sirts K, Grönroos SA, Kurimo M, Virpioja S. 2016. A comparative study on minimally supervised morphological segmentation. *Comput. Linguist.* 42:91–120

Schone P, Jurafsky D. 2000. Knowledge-free induction of morphology using latent semantic analysis. In *Proceedings of the 4th Conference on Computational Natural Language Learning* (*CoNLL 2000*) *and the 2nd Learning Language in Logic Workshop* (*LLL 2000*), ed. C Cardie, W Daelemans, C Nedellec, ETK Sang, pp. 67–72. New Brunswick, NJ: Assoc. Comput. Linguist.

Schone P, Jurafsky D. 2001. Knowledge-free induction of inflectional morphologies. In *Proceedings of the 2nd Meeting of the North American Chapter of the Association for Computational Linguistics on Language Technologies*, pp. 1–9. Stroudsburg, PA: Assoc. Comput. Linguist.

Singla P, Domingos P. 2006. Entity resolution with Markov logic. In *Proceedings of the 6th International Conference on Data Mining* (*ICDM06*), pp. 572–82. Piscataway, NJ: IEEE

Sirts K, Alumäe T. 2012. A hierarchical Dirichlet process model for joint part-of-speech and morphology induction. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 407–16. Stroudsburg, PA: Assoc. Comput. Linguist.

Snover MG, Brent MR. 2001. A Bayesian model for morpheme and paradigm identification. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, pp. 490–98. Stroudsburg, PA: Assoc. Comput. Linguist.

Snover MG, Brent MR. 2002. A probabilistic model for learning concatenative morphology. In *Advances in Neural Information Processing Systems*, ed. S Thrun, LK Obermayer, pp. 1513–20. Cambridge, MA: MIT Press

Snover MG, Jarosz GE, Brent MR. 2002. Unsupervised learning of morphology using a novel directed search algorithm: taking the first step. In *Proceedings of the ACL-02 Workshop on Morphological and Phonological Learning*, pp. 11–20. Stroudsburg, PA: Assoc. Comput. Linguist.

Soricut R, Och F. Unsupervised morphology induction using word embeddings. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 1627–37. Stroudsburg, PA: Assoc. Comput. Linguist.

Sproat RW. 1992. *Morphology and Computation*. Cambridge, MA: MIT Press

Virpioja S, Turunen VT, Spiegler S, Kohonen O, Kurimo M. 2011. Empirical comparison of evaluation methods for unsupervised learning of morphology. *Trait. Autom. Lang.* 52:45–90

Wicentowski R. 2002. *Minimally supervised morphological analysis by multimodal alignment*. PhD thesis, Dep. Comput. Sci., Johns Hopkins Univ., Baltimore, MD

Wicentowski R. 2004. Multilingual noise-robust supervised morphological analysis using the WordFrame model. In *Proceedings of the 7th Meeting of the ACL Special Interest Group in Computational Phonology: Current Themes in Computational Phonology and Morphology*, pp. 70–77. Stroudsburg, PA: Assoc. Comput. Linguist.

Xanthos A. 2008. *Sciences pour la communication 88: Apprentissage automatique de la morphologie. Le cas des structures racine-schème.* Frankfurt: Peter Lang

Yarowsky D, Wicentowski R. 2000. Minimally supervised morphological analysis by multimodal alignment. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pp. 207–16. Stroudsburg, PA: Assoc. Comput. Linguist.

Zeman D. 2009. Using unsupervised paradigm acquisition for prefixes. In *Evaluating Systems for Multilingual and Multimodal Information Access: Proceedings of the 9th Workshop of the Cross-Language Evaluation Forum*, pp. 983–90. Berlin: Springer

Zhang BT, Kim YT. 1990. Morphological analysis and synthesis by automated discovery and acquisition of linguistic rules. In *Proceedings of the 13th Conference on Computational Linguistics*, pp. 431–36. Stroudsburg, PA: Assoc. Comput. Linguist.

Zipf GK. 1935. *The Psychobiology of Language: An Introduction to Dynamic Philology*. Cambridge, MA: MIT Press

Zipf GK. 1949. *Human Behavior and the Principle of Least Effort: An Introduction to Human Ecology*. Cambridge, MA: Addison-Wesley