Computational learning of morphology

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

[1]Departments of Linguistics and Computer Science, The University of Chicago, Chicago, USA, 60637; email: goldsmith@uchicago.edu
[2]Department of Linguistics, The University of Chicago, Chicago, USA, 60637
[3]Section des sciences du language et de l'information, Université de Lausanne, Lausanne, Switzerland

Xxxx. Xxx. Xxx. Xxx. YYYY. AA:1–23

This article's doi:
10.1146/((please add article doi))

Keywords

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

Abstract

This paper reviews work 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, continuing to the present moment. 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.

Contents

1. Introduction . . . 2
   1.1. Goals . . . 2
   1.2. Evaluating: precision and recall . . . 3
2. General considerations . . . 4
   2.1. Zipfian sparsity . . . 4
   2.2. Searching grammar space for the best morphological grammar . . . 5
3. Concatenative morphology . . . 8
   3.1. The problem of segmentation . . . 8
   3.2. From stem and paradigm languages to agglutinative languages . . . 12
4. Non-concatenative morphology . . . 13
5. Word similarity without morphemes . . . 15
6. Allomorphy and morphophonology . . . 16
7. Paradigms . . . 16
8. Other considerations . . . 17
   8.1. Language acquisition by children . . . 17
   8.2. Joint learning . . . 17
   8.3. Supervised and semi-supervised learning . . . 18
9. Conclusions . . . 18

1. Introduction

1.1. Goals

In this paper, 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 will 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 converge.

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 also methods of semi-supervised learning, in which small amounts of humanly analyzed material is 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

[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, such as Paice [1994], Hull et al. [1996]. A different approach to a similar problem can be seen in Jacquemin [1997].

linguistics have converged on the belief that the single most important question is how language, with all its richness and variability around the globe, can be learned by humans so rapidly. The problem of learning morphology has struck many researchers as a part of this problem at just the right level: 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 is it that we would like an unsupervised morphology learner to accomplish? It should be able to read in a textual sample from any human language, either in a standard orthography or a phonological transcription, and to develop a program, or a data structure, which allows us to provide a morphological account of any word from that language, and ideally it should be able to do that even if the word was not in the original sample on which the analysis was based. Now, 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 there are morphologists dispute the existence of morphemes and prefer to provide a word and paradigm based account. We adopt an ecumenical perspective in this paper, and therefore consider both approaches deserving of serious attention by those constructing automatic morphology learners. We will return to this question in section [paradigms] below.

A successful morphological learner would provide answers to questions such as the following ones about the words of a language: 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? If so, how many independent dimensions (or morphosyntactic features) are active in each of the paradigms? What combinations 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] Most of the effort so far has been spent on 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 are 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, there are 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

[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 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.

of these borrowings have entered into our own morphology (as with suffixes like *-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 can leave us unsure about what should count as the right analysis: *boisterous, ambassador, annual, poem* (cf. *poet*), *agrarian, armor, benediction, crucial,* or *worn.* It's not merely that we don't have a method to resolve what counts as the right answer in the unclear cases; we don't 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 documents that the user wanted. The proportion of those 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 position 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 of this is given by Hafer and Weiss [1974], who used an information retrieval task in their empirical comparison of several variants of 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 at predicting where morpheme breaks are, but rather predicting which word forms are part of the same lexeme, and an appropriate evaluation measure must be established for that strategy.

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

2. General considerations

2.1. Zipfian sparsity

Since the very first studies of word frequencies it has been 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 1 or 2), and such distributions are often called "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 and Yang [To appear] provide a recent study of the extent of this phenomenon. We will return to this general problem in section 8.1 below.

2.2. Searching grammar space for the best morphological grammar

Most of the more successful work 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 assume that we start in a random location, while other methods allow one to start at a grammar that is reasonably close to the final solution. In this section we will 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 in 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: neither on where the search should begin, nor precisely how the search should proceed. Those decisions are left to the researchers and their particular implementations.

Gibbs sampling, on the other hand, 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 (MDL).

Several researchers have proposed employing Minimum Description Length analysis for learning grammars in the 1990s, including 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, often called

MDL for short, was proposed by Jorma Rissanen in the 1970s. It 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 in addition, 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.

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, *simulated annealing* can be incorporated into the search, by making the decision on each individual parameter choice not be deterministic (i.e., choose the parameter choice which maximizes the objective function), but rather use 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*, 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*, while if p_3 and p_4 are both set to 1 and p_i=0 for

[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 $\frac{1}{1+e^{-\frac{d}{t}}}$. Early in the learning process, we make t, the pseudo-temperature, be 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.

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 depends 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 Gibbs sampling will visit grammar points with a probability equal to the probability of the grammar given the data; if there is a second grammar which 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, Johnson 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 which in effect recomputes the probability of each rule in the grammar with each production, based on the cached counts. By design, it is only the counts that 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 and Blunsom 2013, Kumar et al. 2015]. They naturally generate output that is zipfian in interesting ways, and Gibbs sampling can be used to guide the learning path from a randomly chosen initial hypothesis to one (or more) optimal morphologies.

2.2.4. Moving intelligently through morphology space. In light of the preceding sections, we can see that models of morphology learning can differ in two ways regarding their conception of arriving at the best analysis in a step-by-step manner: (1) they may differ in whether they begin at a randomly selected point, i.e., a more or less randomly specified initial state, and (2) 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 low level. It is easy for system analyzing English, for example, to fall into the erroneous analysis that assigns the suffixes *-an, -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 re-analyzed, 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 in Goldsmith [2006] and Linguistica 4.) Further discussion of intelligent search strategies can be found, for example, in Snover and Brent [2001, 2002], and in Schone and Jurafsky [2000] and Monson et al. [2004].

3. Concatenative morphology

3.1. The problem of segmentation

Almost all of the work we are dealing with assumes that each word or utterance of the basic data that is observed in a language can be adequately represented as a string of letters, where the letters are either drawn from the standard orthography of the language, or the letters 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 which do not overlap, and which place all letters into exactly one morph (as when the word *prepublishing* is segmented into *pre-publish-ing*). 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 one which 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*, etc.), 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 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, i.e., 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. And so, 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.

Another way of expressing this is to say that 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). Since 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

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

and sound-based properties of utterances.

Let us consider the classic proposal of Zellig 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 will deal with below.[7] Harris proposed in fact not a single method, but a family of closely related methods. His central idea was essentially this: 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 S's *successor frequency.* For example, in one corpus we might find that the successor frequency of the word-inital string *govern* is 6, because it is followed by the letters e, i, m, o, s, and the word-ending boundary marker #; we could write SF(*government*, 6) = 6, meaning that after *government*'s 6th letter, there are 6 possible letter continuations. By way of contrast, the successor frequency of *gover* is just 1 (since only n follows *gover*), SF(*government*, 5) = 1, and SF(*government*, 7) is just 1, since only e follows *governm.* 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 perhaps cutting a word at a point k where SF(*word*, k) $>$ SF(*word*, $k-1$) and SF(*word*, k) $>$ SF(*word*, $k+1$), or perhaps where such a peak is found for *either* successor frequency or predecessor frequency.

Harris's general approach was evaluated by Hafer and Weiss [1974], who explored fifteen different criteria for morpheme breaks that were consistent with the spirit of Harris's idea. They allowed for 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), but ended up with relatively disappointing quantitative results.

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 that the analysis proposed of 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, Zellig Harris, and it was elevated to a central principle in Chomsky (1955). Greedy, local methods of analysis rarely work to understand complex cognitive functions.

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 (or *FSA*) in which edges are labeled with morphemes, and 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 (what is called a *state* corresponds to a *node* in a graphical representation).[8] A word that consists of a prefix, a

[7] Hammarström and Borin [2011] discuss work by Andreev published between 1959 and 1967 in a similar vein which 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 and Dolby [1965, 1966] and Earl [1966] did in their studies of English; their interesting discussions of the problem constituted a sort of proto-theory of the problem of language-independent morphology discovery.

[8] On FSAs and morphology, see Beesley and Karttunen [2003], Sproat [1992], Roark and Sproat

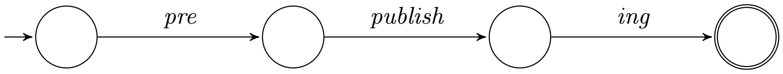

Figure 1

Representing a word's segmentation with an FSA

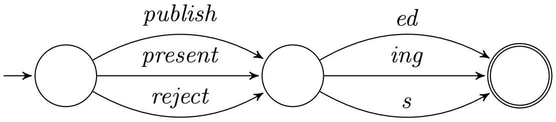

Figure 2

Representing a set of stem's signature with an FSA

stem, and a suffix would thus correspond to a path from the starting state, through two intermediate states, and it would end on a final state, and each of its morphemes would be a label of one of the edges of that path (see Figure 1). There are many such FSAs for any set of words, and 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 will be associated with exactly one single edge.[9]

Imagine that we begin with a wordlist from a language, and build an FSA with only two nodes or states, and in which 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. If we wanted to improve it, 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 was labeled by one of those stems; then we would add edges from that new state to the final state, where each of *those* edges was labeled with one of the suffixes. Having done that, we remove all of the unwanted edges that went straight from the starting state to the final state for those particular bimorphemic words (see figure 2). A moment's thought will convince us that 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 $(M-1) \times N + M \times (N-1)$; we might call that, for a moment, its edge-savings.[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 a node, and establish some threshold value for the edge-savings above which we will insert the node. This is a *greedy* approach, and is to be distinguished from a non-greedy (or *abstemious*) approach, which consider all possible such nodes, and only inserts the very

[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, a good deal of liberty is used 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 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, as a bit of reflection will conclude.

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 to a corpus of English, the top four edge-saving nodes that emerge correspond to stems followed by (1) the pair of suffixes *-s* and $\emptyset$; (2) the pair of suffixes *'s* and $\emptyset$; (3) the pair of suffixes *-ly* and $\emptyset$; and (4) the set of suffixes *-ed, -ing, -s* and $\emptyset$. Goldsmith [2001] calls these constructs *signatures*; they can be thought of as highly corpus-bound proto-paradigms. Each signature is a set of stems followed by a set of suffixes, for which all pairs of stem plus suffix is found in the corpus.[11]

It is not difficult to find sets of suffixes that lead to signatures with high edge-savings. The simplest is to look at all positions in a word where the successor frequency is greater than 1, and for each such point, with its *word-initial string*, to gather into a set the different strings that follow, right up to the end of the word; call them *ending sets.* Having done that, we determine for each of these ending-sets precisely how many different word-initial strings led 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 (since those two numbers correspond directly to the M and N described just above). A set of signatures derived in this way, each containing at least two word-initial strings (in effect, two stems), produces quite an interesting first approximation of the morphology of the final suffix of an inflecting language, and the larger the edge-savings, the more certain the analyses are.

The emphasis on signatures is motivated by the fact that languages produce many examples of pseudo-generalizations which only appear once or twice: while the pattern *read, reads, reading*, with its signature $\emptyset$*, -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.

Let us reflect on how such an approach might fail, however. If a set of suffixes all begin with the same letter (or letters), it 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 has 39 pairs of words like *affluent, affluence* that are analyzed as having suffixes *t, ce.* More strikingly, while there are about 170 stems like *climb* and *creak* that occur with exactly the suffixes *ed, ing, s* and $\emptyset$, there are there are about 90 that are like *move*, which would be analyzed as having stems such as *mov, embrac, silenc* and the suffixes *-e, -ed, -ed, -ing.* Of course, this allomorphy (loss of stem-final *e* before the suffixes *-ed* and *ing*) no longer reflects spoken English, and 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, or in 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*, $\emptyset$}, or the set {*ation, ed, er, ing, s*, $\emptyset$}, like *confirm.* It is a very difficult computational problem to distinguish between those affixes which are productive and those which are

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

not.[12] In this case, this means determining which of the stems that appear with the suffixes *∅, -ed, -s, -ing* can also appear with *-er* or *-ation*.[13]

3.2. From stem and paradigm languages to agglutinative languages

In a good deal of the work referenced so far, the focus has been on determining *the* appropriate morphological break between stem and suffix (and/or break between prefix and stem). But even in Western European languages, it is not at all uncommon for a word to have several suffixes (*tranform-ation-less, fundament-al-ism-s*), and such languages as Finnish, Turkish, and Hungarian quite commonly have several affixes in a word. Looking beyond Europe, the number of morphemes per word is greater still in good part of the world's languages. How can the methods discussed here be extended to deal with agglutinative languages, with many morphemes per word?

Linguistica 4, the system described in 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 on the third iteration it identified *-al* as a suffix to the left of *-ism*.

The ParaMor system described in 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, administrada-s* (for *-adas, -das, -as, -s* in different schemes), with the final inferred segmentation as *administr-a-d-a-s*.

The Morfessor model family [summarized in Creutz and Lagus 2007] is designed for unsupervised morpheme segmentation of highly inflecting and compounding languages. Initially, two search algorithms were proposed by Creutz and Lagus [2002]. The first considers each word in a corpus successively, evaluates each possible split into two parts using an MDL-based cost function and recursively processes the resulting parts until no further gain can be obtained. The second method starts 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 re-segment the data in a way that maximizes the probability that the model assigns to them. More recent versions of Morfessor improve segmentation results by

[12] This 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 and Yang [To appear].

[13] Truncation has become an important morphological process in virtually all the European languages, as when *stylographe* is truncated in French to *stylo*. Pham and 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' work.

incorporating knowledge of morph categories (e.g., prefix, suffix, stem) into the model.

Linguistica 5 uses a similar method to 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, where the robustness is defined as the length of the morpheme times 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 the stem-final string with the greatest robustness, generating the FSA in Figure 3, where orange lines indicate sets of stems. It uses an abstemious strategy, as explained above, choosing to discovery 100 internal suffixes (suffixes preceding other suffixes) across the entire FSA of English.

4. Non-concatenative morphology

Morph concatenation is by far the most frequent word formation mechanism in languages of the world and it is no surprise that a vast majority of the research on morphology learning has specifically addressed it. 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 are accounted for by positing the existence of *roots* /klb/ and /rml/, i.e. morphs consisting of a sequence [rather than a string, cf. Lee 2015] of consonants, conveying the lexical meaning of the word, and combining with various patterns of vowel qualities and quantities which express inflectional or derivational variations. Ablaut in English strong verbs inflection is another well-known example of a non-concatenative process, albeit by no means as productive as root-and-pattern morphology in Semitic languages.

Approaches to the unsupervised learning of non-concatenative morphology have been mostly applied to Arabic, in particular the classical and modern standard written forms (often vowelized). Aside from the early work of De Roeck and 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 attempt to solve. Some of the proposed algorithms [Bati 2002, Rodrigues and 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 that has been used in this context comes from the field of information retrieval. Indeed, in order to cluster words that share the same root, De Roeck and Al-Fares [2000] use a string similarity coefficient initially designed by Adamson and 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 and Al-Fares 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 and Carroll [2013b] have obtained good results on Arabic root identification without recourse to such supervision. Their

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

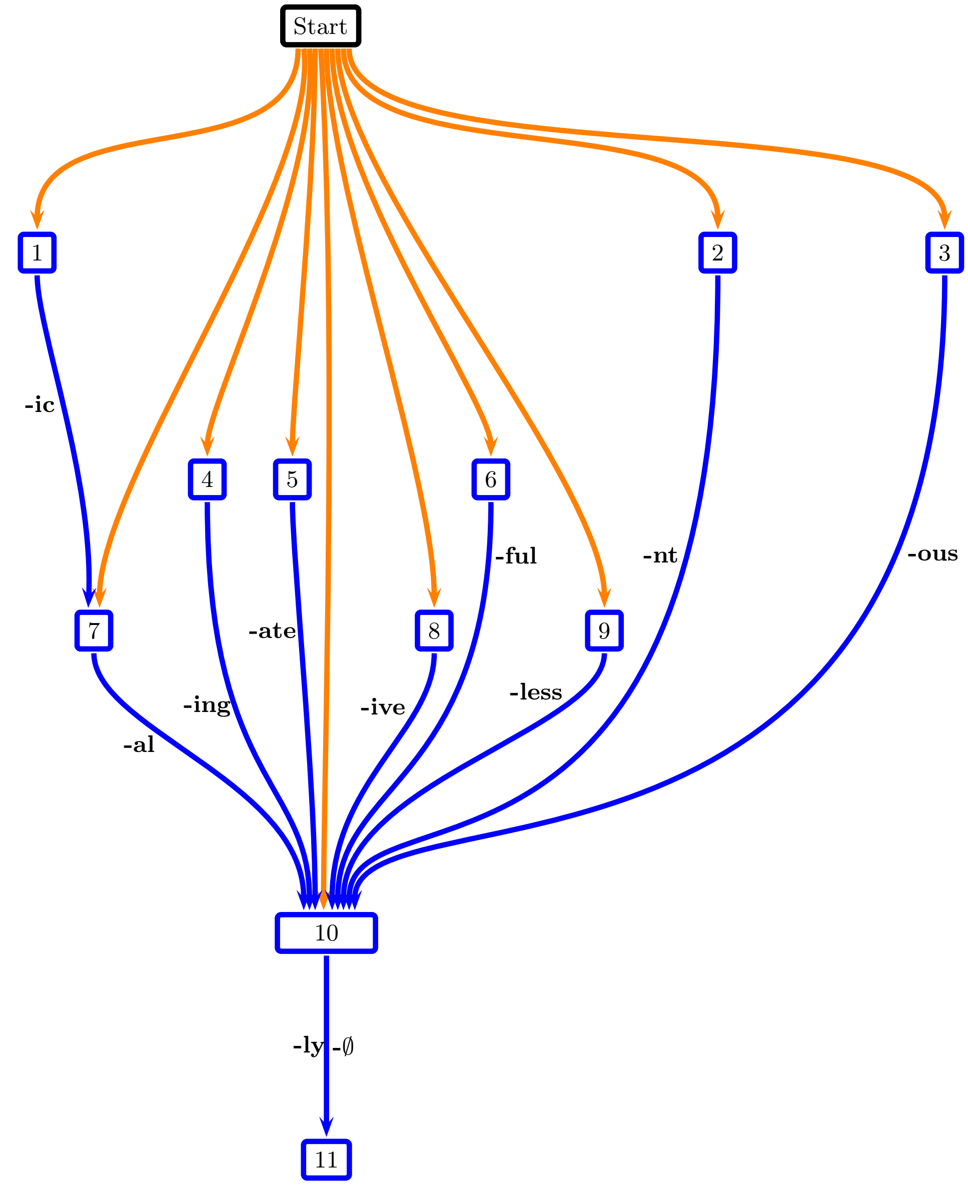

Figure 3

Finding multiple suffixes in Linguistica 5

approach builds on the work of De Pauw and Wagacha [2007], who use *n*-gram features (specified with regard to their initial, internal, or final position in a word) for training a maximum entropy classifier to find relationships between morphologically related words, in order to subsequently identify their prefixes. Khaliq and Carroll show that this approach can be readily adapted to root-and-pattern morphology by defining features as subsequences of (possibly non-adjacent) letters rather than contiguous substrings. In a follow-up to this work, Khaliq and 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 cooccur with frequent patterns, and vice-versa.

Some learning heuristics rely on specific properties of root-and-pattern morphology. Thus Elghamry [2004], later followed by Rodrigues and 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 non-concatenative morphology learning for a given corpus. Xanthos does this by quantifying the compression resulting from modelling the data with a root-and-pattern analysis, which turns out to be order of magnitudes larger for Arabic than for English or French for instance.

The latest proposals in this area, by Fullwood and O'Donnell [2013] and Botha and Blunsom [2013], adopt the non-parametric Bayesian framework of adaptor grammars (see section 2.2.3) pioneered by Johnson et al. [2007b]. Interestingly, these works also have in common that they simultaneously deal with non-concatenative *and* concatenative aspects of Semitic morphology, but they do so in very different ways. Fullwood and O'Donnell 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 described with template $r - r\ r - -$, where r stands for a root consonant and – for a residue component. Botha and Blunsom, on the other hand, use the *range concatenating grammar* formalism [Boullier 2000] to represent concatenation and intercalation operations in a distinct but unified fashion–thus contributing to solve what is arguably one of the main current challenges in the field.

5. Word similarity without morphemes

We noted earlier that not all analyses of words is based on the assumption that words are analyzable into morphs or morphemes, and the computational analysis of word relationships without morphemes has been undertaken as well. The work of [Adamson and Boreham 1974], where words are clustered based on the bigrams they contain (see section 4), is an early example of this. Other systems have used string-edit distance as a method for determining similarity between strings, as [Baroni et al. 2002] do; in a similar vein, some systems have 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, e.g., Mayfield and McNamee [2003]).

Methods that do not directly attack the problem of morpheme discovery within words often focus on distributional information above the word-level, which can be either synat-

[15] See also Goldsmith and Xanthos [2009] for a review.

[16] Bati [2002] was a precursor to this work, although phonological categories were hard-coded in this case.

actic, semantic, or—most commonly—some combination of the two. Schone and Jurafsky [2000] employ latent semantic indexing (LSA) on a set of documents to help to determine of 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 be chance (on LSA, see Deerwester et al. [1990]). Other early work here include Baroni et al. [2002] and Neuvel and Fulop [2002], and additional references can be found in Hammarström and Borin [2011].

6. 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.

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 the learning of the close relationship of [e] and [ie] in Spanish (learning of 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 which 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 the case here, 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 just a handful of the morphology has been inferred, while in other cases it may take a considerable amount of morphological analysis before the morphophonological generalization emerges.

Early work on learning rules of allomorphy include Zhang and Kim [1990]; some additional work on this described in Gaussier [1999]. Goldwater and 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. See also Schone and Jurafsky [2001], Wicentowski [2002], Wicentowski [2004]. This appears to be an area ripe for additional progress.

7. 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 and O'Brien [2006] model inflectional patterns using a connectionist approach, with the nodes in the hidden layer corresponding to the more abstract inflection classes. More recent work treats inflection class inference as a clustering problem in unsupervised learning [Zeman 2009, Brown and Evans 2012, Lee 2014, Beniamine and 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 pre-categorized in distinct inflection classes can act as a gold standard dataset. But if inflection classes are thought of having a hierarchical configuration, evaluation for inflection class inference by clustering is much less clear. Nonetheless, a hierarchical view of inflection classes offers insights with regards to the *partial* similarities and differences across morphological paradigms.

8. Other considerations

8.1. 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 and Yang [To appear] provide an overview of the morphological learning problem in language acquisition, covering issues of data sparsity, productivity, and analogy.

Most published work in computational morphology does not speak directly to the problem of human morphological acquisition, because the datasets 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 proposed. Some recent work, however, does use child directed speech, e.g., Frank et al. [2013] (who also make use of syntactic information, though they do batch learning). Lee and Goldsmith [2016] present preliminary results of incremental morphological learning using child directed speech.

8.2. Joint learning

In principle, linguistic knowledge at multiple levels of grammar can be learned simultaneously, and it is reasonable to ask if such knowledge from different levels may interact or even improve one another. 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 recent work such as Dreyer and Eisner [2011], Lee et al. [2011], Sirts and Alumäe [2012], Frank et al. [2013] has shown that learning morphology and syntax simultaneously does improve results for both components. Joint learning not only leads to fruitful results for the computational tasks at hand, but also provides important insights for theoretical questions, such as those in connection with the architecture of grammar.

8.3. Supervised and semi-supervised learning

Stepping outside of the unsupervised learning paradigm, we note that there has been a substantial amount of work on supervised and semi-supervised learning of morphology. One factor that facilitates research on (semi-)supervised learning of morphology appears to be the increased availability of machine-readable inflection tables. Durrett and DeNero [2013] employ inflectional data from Wiktionary for supervised morphological learning. Other authors such as Wicentowski [2004], Ahlberg et al. [2014] make use of similar resources together with large corpus text for semi-supervised learning tasks. The system by Yarowsky and Wicentowski [2000] does not require inflection pairs or tables, but assume minimal knowledge of root words as well as mapping between parts of speech and expected morphological patterns. Ruokolainen et al. [2016] provides a review comparing unsupervised, supervised, and semi-supervised approaches to morphological segmentation. As compiled and annotated datasets of morphological paradigms – even for low-resource languages – are increasingly more easily available, the semi-supervised learning research paradigm with highly competitive results is likely to become more active in the years to come.

9. Conclusions

Reviewing the work of the past twenty years, we can observe a good deal of success with the problem of word-segmentation and the discovery of word-internal structure. Two generalization come through: the first is that the successes we see would not have been possible without the emergence of machine learning in the last thirty years. The tools developed there have been absolutely essential for the work described here. The second is a bit less obvious, but significant. The successful methods all take the form of developing an explicit objective function that is based on characterizing a grammar and integrated a finite set of data, and then selecting a solution as the *argmin* winner: the learned grammar is the one which minimizes the objective function.

While that view of "learning as computation of *argmin*" sounds like something that would come from machine learning, it also resonates with some traditions 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 which generate the training data. "Learning as *argmin*" is not a natural perspective from the point of view, for example, of 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 which 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—and there is a *but*—that style of developing morphology does not appear at this time 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 begin 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 work on category induction will help with this task, just as methods of induction of rules of morphophonology will help to provide simpler computational models of morphology *per se*.

LITERATURE CITED

George W. Adamson and Jillian Boreham. The use of an association measure based on character structure to identify semantically related pairs of words and document titles. *Information Storage and Retrieval*, 10(7-8):253–260, 1974.

Malin Ahlberg, Mans Hulden, and Markus Forsberg. Semi-supervised learning of morphological paradigms and lexicons. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 569–578, Gothenburg, Sweden, April 2014. Association for Computational Linguistics.

Marco Baroni, Johannes Matiasek, and Harald Trost. Unsupervised discovery of morphologically related words based on orthographic and semantic similarity. In *Proceedings of the ACL-02 Workshop on Morphological and Phonological Learning*, pages 48–57. Association for Computational Linguistics, jul 2002.

Tesfaye Bayu Bati. Automatic morphological analyzer for Amharic: An experiment employing unsupervised learning and autosegmental analysis approach. Master's thesis, Addis Ababa University, Ethiopia, 2002.

Kenneth R. Beesley and Lauri Karttunen. *Finite State Morphology*. CSLI Publications, 2003.

Sarah Beniamine and Benoît Sagot. Segmentation strategies for inflection class inference. In *Décembrettes 9, Colloque international de morphologie*, 2015.

Jan A. Botha and Phil Blunsom. Adaptor grammars for learning non-concatenative morphology. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 345–356, Seattle, Washington, USA, October 2013. Association for Computational Linguistics.

Pierre Boullier. Range concatenation grammars. In *Proceedings of IWPT 2000*, pages 53–64, 2000.

Michael R Brent. Advances in the computational study of language acquisition. *Cognition*, 61(1): 1–38, 1996.

Michael R Brent, Sreerama K Murthy, and Andrew Lundberg. Discovering morphemic suffixes: A case study in minimum description length induction. *Proceedings of the fifth international workshop on artificial intelligence and statistics*, 1995.

Dunstan Brown and Roger Evans. Morphological complexity and unsupervised learning: Validating Russian inflectional classes using high frequency data. In Ferenc Kiefer, Mária Ladányi, and Péter Siptár, editors, *Current Issues in Morphological Theory: (Ir)regularity, analogy and frequency*, pages 135–162. Amsterdam, Philadelphia, 2012.

Alexander Clark. Supervised and unsupervised learning of arabic morphology. In Abdelhadi Soudi, Antalvan den Bosch, and GÃijnter Neumann, editors, *Arabic Computational Morphology*, volume 38 of *Text, Speech and Language Technology*, pages 181–200. Springer Netherlands, 2007.

Mathias Creutz and Krista Lagus. Unsupervised discovery of morphemes. In *Proceedings of the ACL-02 Workshop on Morphological and Phonological Learning - Volume 6*, MPL '02, pages 21–30, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.

Mathias Creutz and Krista Lagus. Unsupervised models for morpheme segmentation and morphology learning. *ACM Trans. Speech Lang. Process.*, 4(1):3:1–3:34, February 2007.

Carl de Marcken. *Unsupervised Language Acquisition*. PhD thesis, MIT, 1996.

Guy De Pauw and Peter Waiganjo Wagacha. Bootstrapping morphological analysis of Gĩkũyũ using unsupervised maximum entropy learning. In *INTERSPEECH 2007, 8th Annual Conference of the International Speech Communication Association*, pages 1517–1520, 2007.

Anne De Roeck and Waleed Al-Fares. A morphologically sensitive clustering algorithm for identifying Arabic roots. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics (ACL-00)*, 2000.

Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. Indexing by Latent Semantic Analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990.

Arthur P. Dempster, Nan M. Laird, and Donald B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B*, 39:1–38, 1977.

Shilpa Desai, Jyoti Pawar, and Pushpak Bhattacharyya. A framework for learning morphology using suffix association matrix. *COLING 2014*, page 28, 2014.

Markus Dreyer and Jason Eisner. 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*, pages 616–627, Edinburgh, Scotland, UK., jul 2011. Association for Computational Linguistics.

Greg Durrett and John DeNero. 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*, pages 1185–1195, Atlanta, Georgia, June 2013. Association for Computational Linguistics.

Lois L. Earl. Structural definition of affixes from multisyllable words. *Mechanical Translation and Computational Linguistics*, 9:34–37, 1966.

Khaled Elghamry. A constraint-based algorithm for the identification of Arabic roots. In *Proceedings of the Midwest Computational Linguistics Colloquium*, 2004.

Gudrun Flenner. Ein quantitatives Morphsegmentierungssystem für spanische Wortformen. *Computatio linguae II*, pages 31–62, 1994.

Stella Frank, Frank Keller, and Sharon Goldwater. Exploring the utility of joint morphological and syntactic learning from child-directed speech. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2013.

Michelle Fullwood and Tim O'Donnell. Learning non-concatenative morphology. In *Proceedings of the Fourth Annual Workshop on Cognitive Modeling and Computational Linguistics (CMCL)*, pages 21–27, Sofia, Bulgaria, 2013. Association for Computational Linguistics.

Éric Gaussier. Unsupervised learning of derivational morphology from inflectional lexicons. In *Proceedings of ACL'99 Workshop: Unsupervised Learning in Natural Language Processing*, 1999.

John A. Goldsmith. Unsupervised learning of the morphology of a natural language. *Computational Linguistics*, 27(2):153–198, 2001.

John A. Goldsmith. An algorithm for the unsupervised learning of morphology. *Natural Language Engineering*, 12:353–371, 12 2006.

John A. Goldsmith. Segmentation and morphology. In *The Handbook of Computational Linguistics and Natural Language Processing*, pages 364–393. Wiley-Blackwell, 2010.

John A. Goldsmith and Jeremy O'Brien. Learning inflectional classes. *Language Learning and Development*, 2(4):219–250, 2006.

John A. Goldsmith and Aris Xanthos. Learning phonological categories. *Language*, 85(1):4–38, 2009.

Sharon Goldwater and Mark Johnson. Priors in bayesian learning of phonological rules. In *Proceedings of the Seventh Meeting of the ACL Special Interest Group in Computational Phonology (SIGPHON '04)*, Barcelona, Spain, 2004.

Margeret A. Hafer and Stephen F. Weiss. Word segmentation by letter successor varieties. *Information Storage and Retrieval*, 10:371–385, 1974.

Harald Hammarström and Lars Borin. Unsupervised Learning of Morphology. *Computational Linguistics*, 37(2):309–350, 2011.

Zellig S. Harris. From phoneme to morpheme. *Language*, 31:190–222, 1955.

Zellig S. Harris. Morpheme boundaries within words: Report on a computer test. *Transformations and Discourse Analysis Papers*, 73, 1967.

Derrick Higgins. *A Multi-modular Approach to Model Selection in Statistical Natural Language Processing.* PhD thesis, University of Chicago, 2002.

David A Hull et al. Stemming algorithms: A case study for detailed evaluation. *JASIS*, 47(1):

70–84, 1996.

Christian Jacquemin. Guessing morphology from terms and corpora. In *ACM SIGIR Forum*, volume 31, pages 156–165. ACM, 1997.

Mark Johnson. Using adaptor grammars to identify synergies in the unsupervised acquisition of linguistic structures. In *Proceedings of ACL-08:HLT*, pages 398–406, 2008.

Mark Johnson, Thomas L Griffiths, and Sharon Goldwater. Bayesian inference for PCFGs via markov chain monte carlo. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 139–146, 2007a.

Mark Johnson, Thomas L. Griffiths, and Sharon Goldwater. Adaptor grammars: a framework for specifying compositional nonparametric bayesian models. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 641–648, Cambridge, MA, 2007b. MIT Press.

Allen Kent, Madeline M. Berry, Fred U. Luehrs, and J. W. Perry. Machine literature searching viii. operational criteria for designing information retrieval systems. *American Documentation*, 6(2): 93–101, 1955.

Bilal Khaliq and John Carroll. Induction of Root and Pattern Lexicon for Unsupervised Morphological Analysis of Arabic. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 1012–1016, Nagoya, Japan, oct 2013a. Asian Federation of Natural Language Processing.

Bilal Khaliq and John Carroll. Unsupervised Induction of Arabic Root and Pattern Lexicons using Machine Learning. In *Proceedings of the International Conference Recent Advances in Natural Language Processing RANLP 2013*, pages 350–356, Hissar, Bulgaria, 2013b. INCOMA Ltd. Shoumen, BULGARIA.

Arun Kumar, Lluís Padró, and Antoni Oliver. Learning agglutinative morphology of Indian languages with linguistically motivated adaptor grammars. In *Proceedings of Recent Advances in Natural Language Processing*, pages 307–312. 2015.

Jackson L. Lee. Automatic morphological alignment and clustering. Technical Report TR-2014-07, Department of Computer Science, University of Chicago, 2014.

Jackson L. Lee. 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*, pages 161–167, Denver, Colorado, June 2015. Association for Computational Linguistics.

Jackson L. Lee and John A. Goldsmith. Linguistica 5: Unsupervised learning of linguistic structure. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics*, San Diego, California, June 2016. Association for Computational Linguistics.

Yoong Keok Lee, Aria Haghighi, and Regina Barzilay. Modeling Syntactic Context Improves Morphological Segmentation. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning*, pages 1–9, Portland, Oregon, USA, 2011. Association for Computational Linguistics.

Constantine Lignos and Charles Yang. Morphology and language acquisition. In *Cambridge Handbook of Morphology*. Cambridge University Press, To appear.

James Mayfield and Paul McNamee. Single n-gram stemming. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, SIGIR '03, pages 415–416, New York, NY, USA, 2003. ACM.

Christian Monson, Alon Lavie, Jaime Carbonell, and Lori Levin. 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, pages 52–61, Stroudsburg, PA, USA, 2004. Association for Computational Linguistics.

Christian Monson, Jaime Carbonell, Alon Lavie, and Lori Levin. ParaMor: Finding paradigms across morphology. In *Advances in Multilingual and Multimodal Information Retrieval*, pages 900–907. Springer, 2007.

Sylvain Neuvel and Sean A. Fulop. Unsupervised learning of morphology without morphemes. *CoRR*, cs.CL/0205072, 2002.

Timothy O'Donnell, Jesse Snedeker, Joshua B. Tenenbaum, and Noah Goodman. Productivity and reuse in language. In *Proceedings of the 33th Annual Meeting of the Cognitive Science Society*, 2011.

Timothy J O'Donnell. *Productivity and reuse in language: A theory of linguistic computation and storage*. MIT Press, 2015.

Chris D. Paice. An evaluation method for stemming algorithms. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 42–50. Springer-Verlag New York, Inc., 1994.

Mike Pham and Jackson L. Lee. Combining successor and predecessor frequencies to model truncation in Brazilian Portuguese. Technical Report TR-2014-15, Department of Computer Science, University of Chicago, 2014.

H. L. Resnikoff and J. L. Dolby. The nature of affixing in written English. *Mechanical Translation and Computational Linguistics*, 8:84–89, 1965.

H. L. Resnikoff and J. L. Dolby. The nature of affixing in written English, part ii. *Mechanical Translation and Computational Linguistics*, 9:23–33, 1966.

Brian Roark and Richard William Sproat. *Computational approaches to morphology and syntax*. Oxford University Press Oxford, 2007.

Paul Rodrigues and Damir Ćavar. Learning Arabic Morphology Using Information Theory. In *Proceedings from the Annual Meeting of the Chicago Linguistics Society*, volume 41, pages 49–58, 2005.

Paul Rodrigues and Damir Ćavar. Learning Arabic morphology using statistical constraint-satisfaction models. In Elabas Benmamoun, editor, *Perspectives on Arabic Linguistics: Papers from the Annual Symposium on Arabic Linguistics*, pages 63–75, 2007.

Teemu Ruokolainen, Oskar Kohonen, Kairit Sirts, Stig-Arne Grönroos, Mikko Kurimo, and Sami Virpioja. A comparative study on minimally supervised morphological segmentation. *Computational Linguistics*, 2016.

Patrick Schone and Daniel Jurafsky. Knowledge-free induction of morphology using Latent Semantic Analysis. In *Fourth Conference on Computational Natural Language Learning, CoNLL 2000, and the Second Learning Language in Logic Workshop, LLL 2000, Held in cooperation with ICGI-2000, Lisbon, Portugal, September 13-14, 2000*, pages 67–72, 2000.

Patrick Schone and Daniel Jurafsky. Knowledge-free induction of inflectional morphologies. In *Proceedings of the second meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies*, pages 1–9. Association for Computational Linguistics, 2001.

Kairit Sirts and Tanel Alumäe. 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*, pages 407–416, Montréal, Canada, 2012. Association for Computational Linguistics.

Matthew G. Snover and Michael R. Brent. A bayesian model for morpheme and paradigm identification. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics (ACL-01)*, pages 490–498, 2001.

Matthew G Snover and Michael R Brent. A probabilistic model for learning concatenative morphology. In *Advances in Neural Information Processing Systems*, pages 1513–1520, 2002.

Matthew G Snover, Gaja E Jarosz, and Michael R Brent. 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-Volume 6*, pages 11–20. Association for

Computational Linguistics, 2002.

Richard William Sproat. *Morphology and computation.* MIT press, 1992.

Sami Virpioja, Ville T Turunen, Sebastian Spiegler, Oskar Kohonen, and Mikko Kurimo. Empirical comparison of evaluation methods for unsupervised learning of morphology. *Traitement Automatique des Langues*, 52(2), 2011.

Richard Wicentowski. *Minimally supervised morphological analysis by multimodal alignment.* PhD thesis, 2002.

Richard Wicentowski. Multilingual noise-robust supervised morphological analysis using the wordframe model. In *Proceedings of the Seventh Meeting of the ACL Special Interest Group in Computational Phonology*, pages 70–77, Barcelona, Spain, July 2004. Association for Computational Linguistics.

Aris Xanthos. *Apprentissage automatique de la morphologie: Le cas des structures racine-schème*, volume 88. Peter Lang, 2008.

David Yarowsky and Richard Wicentowski. Minimally supervised morphological analysis by multimodal alignment. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, ACL '00, pages 207–216, Stroudsburg, PA, USA, 2000.

Daniel Zeman. Using unsupervised paradigm acquisition for prefixes. In *Evaluating Systems for Multilingual and Multimodal Information Access, 9th Workshop of the Cross-Language Evaluation Forum, CLEF 2008, Aarthus, Denmark, September 17-19, 2008, Revised Selected Papers*, pages 983–990. Springer-Verlag, Berlin, 2009.

Byoung-Tak Zhang and Yung-Taek Kim. Morphological analysis and synthesis by automated discovery and acquisition of linguistic rules. In *Proceedings of the 13th conference on Computational linguistics-Volume 2*, pages 431–436. Association for Computational Linguistics, 1990.

George Kingsley Zipf. *The Psychobiology of Language: An Introduction to Dynamic Philology.* M.I.T. Press, Cambridge, MA, 1935.

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