From Signatures to Finite State Automata

John Goldsmith and Yu Hu

Department of Computer Science
University of Chicago
Chicago IL 60637
ja-goldsmith@uchicago.edu yuhu@cs.uchicago.edu

1 Introduction

In this paper, we outline the design of a non-deterministic finite state automaton (NFSA) for natural language morphology, and compare it to previous work in unsupervised learning of morphology. In Section 2, we describe the nature of an MDL-based system for unsupervised learning of morphology, using the signature-based model of Goldsmith 2001 as an example, and we describe some drawbacks of the signature-based model. In Section 3, we present an alternative model which is a non-deterministic finite state automaton, distinguishing between convergent and divergent states, a difference that corresponds to inflectional versus derivational morphology and specify an MDL model based it. In Section 4, we review the ways in which a Patricia trie has been used by several authors as a bootstrap means for finding morphemes, and the final sections describe the ways in which we are focusing on obtaining layers of morphological structure.

2 Signatures

2.1 Earlier work

In Goldsmith 2001, a model for unsupervised learning of natural language morphology is presented, employing Minimum Description Length (MDL) analysis as a tool to provide an objective function in selecting the optimal morphology for a natural language corpus. An MDL-based analysis of this sort is, in effect, a specific and explicit mapping M from corpora to a data structure of a particular sort, a data structure that can assign a probability to each word of the corpus and whose treatment of the data can reasonably be interpreted as a morphological analysis (and thus be compared with a morphological gold standard established for particular selected corpora or for a language). Such a data structure is, we may say, a morphology.

In that paper, the data structure was chosen with an implicit bias towards European languages: not intentionally, but in fact, and to some degree for convenience. The morphology that was employed was a set of three lists: a list of stems, a list of suffixes, and a list of *signatures*. A signature is a set of suffixes, all of which appear following one or more stems in the corpus. From a formal point of view, a signature is defined as a pair of lists of pointers: a list of pointers to stems, and a list of pointers to suffixes (or prefixes; we will omit explicit reference to that alternative from here on).

The heart of MDL analysis is the selection of the particular morphology which, given a particular corpus, minimizes the description length of the corpus, where the *description length* is defined as the information length of the morphology plus the compressed length of the corpus when we use the morphology to define a probability distribution whose support includes the corpus. If the morphology is capable of assigning a probability to each word in the corpus, then we calculate the compressed length of the corpus as the sum of the base-2 logarithm of the inverse probability of each word in the corpus, as calculated by the morphology.

MDL informs us that for any given corpus C, we wish to choose the particular morphology that minimizes the description length, but MDL does not suggest how to find it. In Goldsmith 2001, a general strategy composed of a bootstrap heuristic and a sequence of incremental heuristics is outlined, and a detailed description is given in Goldsmith 2004. Open source for this software is available at *linguistica.uchicago.edu*.

2.2 Concerns with a signature-based morphology

A signature-based morphology of the sort we have just described is well-designed to describe the morphology of a language in which most of the morphological complexity is found in words of one or two morphemes. When there is a significant proportion of words with more than two morphemes, this approach is still capable of working well in *some* cases. For example, the process can be reapplied to the stems which have been discovered, thus finding morphological relationships between stems.

However, it is not uncommon in the world's languages (though not in Indo-European languages) for there to be a sequence of four or more morphological positions in a word, each of which can be realized in a different manner, as in a Swahili verb in (1) (this table simplifies the facts a bit for purposes of exposition, in particular leaving out the system of suffixes).

(1) Swahili verb

Subject marker	Tense marker	Object marker	Verb stem
ni 'I' a 's/he' tu 'we' wa 'they'	li 'past' na 'present'	ni 'me' tu 'us' wa 'them'	fanya 'do' sema 'speak' ona 'see'

A signature-based morphology is not well-designed to deal with a language of this sort, for at least two reasons. One reason is that the morphological root is in the center of the word; there is both a very rich prefixal inflectional system, and a reasonably rich derivational suffixal system, details of which we have omitted from (1); we return below to the relevance of the inflectional/derivational distinction. The second reason is that if we successively perform a morphological operation that peels off morphemes from one end (or the other), there will not, in general, be a homogeneity to the morphemes pulled off at any one iteration.[1]

Indeed, concerns along these lines do arise even in the context of European languages. For example, in French and other Romance languages, adjectival stems are followed by a masculine or feminine suffix (French: *-e-* feminine, *-Ø-* masculine), followed by a number marker (French: *-Ø* singular, *-s* plural). If the adjective itself is morphologically complex, this leads to an analysis such as:

(2)

computationn	-ell	-e	-s
computation-	adj.	feminine	plural

In the example in (2), the root is the noun "*computation*", followed by a suffix "*ell*" forming an adjective, followed by the feminine and plural markers. (The case is rendered slightly more complex by the fact that the suffix "*ell*" is chosen before the feminine suffix, and "*el*" is chosen before the masculine suffix, and that the root "*computation*" takes the form "*computationn*" before a vowel-initial suffix).

The signature-based model of morphology serves well to allow us to focus on difficult problems of segmentation between stems and affixes, and the work cited above has used an MDL- and signature-based model to tackle these problems. Nonetheless, the signature-based model does not appear to us, despite our efforts, to generally scale up easily to languages with richer morphological systems. We return to some of the reasons shortly; we turn first to an alternative.

3 Nondeterministic finite-state automata

We will first describe the kind of finite-state automata we are seeking, and then turn to how to learn them.

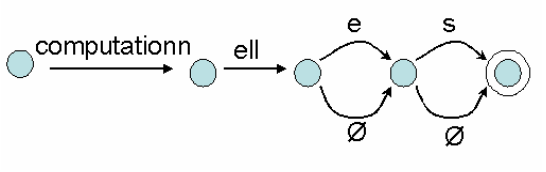
3.1 Convergence and divergence

At least since the work of Koskenniemi 1983 (see Jurafsky and Martin 2003, for example, for

[1] Let us clarify that with a simple example. Suppose there were a language with a set of verbal suffixes {*ing, ed, s*} and a final interrogative suffix {*je*}. Some verbs would be of the form {*X-ing, X-ed, X-s*}, and some of the form {*X-ing-je, X-ed-je, X-s-je*}. The first suffixal iteration would extract the suffixes {*je, ing, ed, s*} and the second would extract the suffixes {*ing, ed, s*}, and neither iteration would capture the generalization that the suffixes {*ing, ed, s*} occur on an inner layer, and while a choice of { *je, Ø* } occurs on an outer layer.

an overview), most computational work on morphology has adopted the nondeterministic finite-state automaton or transducer as the formal model of choice, and some of the work on unsupervised learning recently has followed these lines as well (such as Altun and M. Johnson 2001, and H. Johnson and Martin 2003). Such a model does not appear to have the limitations alluded to above when it encounters languages with richer morphologies (i.e., more morphemes per word); a finite-state device could naturally contain a subgraph that could handle the form in (2). Viewing the problem not as transduction but as generation, we could easily imagine a subgraph as in (3), in which each transition is associated with the emission of a morpheme:

(3)

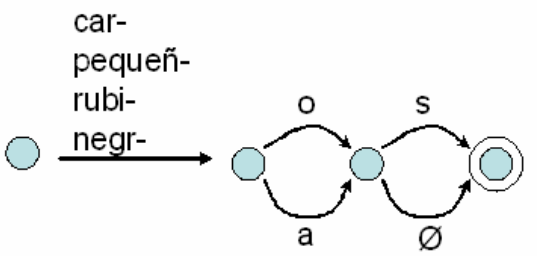


Finite state automata are often described in a fashion that associates emission (or acceptance) of a string with an arc that joins two states, as in (3), but one can also of course develop formalisms of finite state automata that associate emission (or acceptance) of a string with a particular state rather than state-transition. These approaches are equivalent from the point of view of the languages that they generate. However, they are by no means equivalent as natural structures to use as the basis for an MDL analysis,[2] and it turns out that neither is quite right for this purpose; the best model draws on aspects of both, for the following reasons.

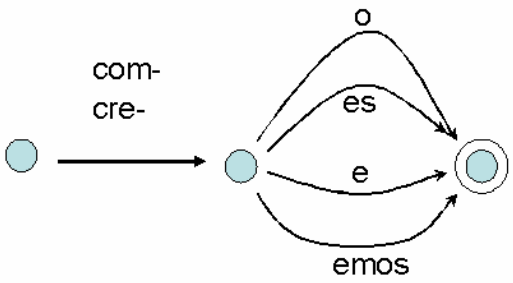
a. *Convergence*: In the arcs of an arc-emitting automaton, there is no reason to expect the arcs that leave a given state S to converge on a common state T. In natural language, however, such convergence *is* the normal case. This is especially true in the case of inflectional morphology, which is the morphological marking of such characteristics as subject agreement and tense marking on verbs; or gender, number, and case marking on adjectives and nouns.

(4) example: Spanish adjectives and verbs.

a. Spanish adjectives



b. Spanish verbs



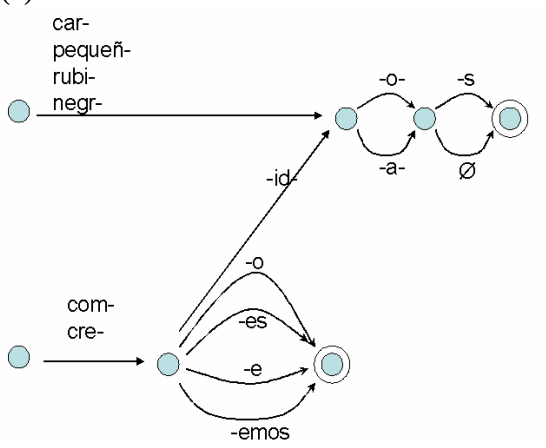
This behavior is easy to model with a state-emission automaton, in which each state is associated with a set of morphemes $\{m_i\}$ and probabilities $\{p_i\}$, and passage through the state is associated with the emission of one of these morphemes m_i with probability p_i.

b. *Divergence*: While convergence is the most common structure in the graph of a morphology, it is not always found, and its alternative, divergence, is typically associated with a morpheme that marks a change of part of speech. (We speak of a state being *divergent* if its out-arcs end at two or more distinct states, and *convergent* otherwise; divergence can be quantified in natural ways.) A typical example is given in (5), which extends the example in (4). The suffix *–id-* is a participial suffix which in effect shifts, or converts, a verb stem to an adjective, thus joining the two graphs in (4). The suffix *–id-* is responsible for the generation of

[2] For essentially this reason: the natural length of the morphology is the sum of the lengths of the component pieces, and multiplying states or arcs unnecessarily has a very significant impact on the calculated length of the morphology.

words such as *com-id-o, com-id-o-s, com-id-a-s*, etc.

(5)



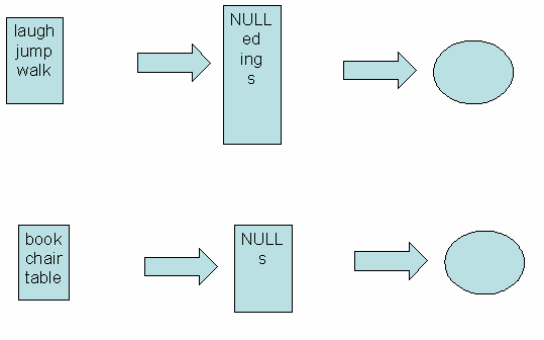
Divergence is easy to express with an arc-emission automaton; as we noted above, a random arc-emission automaton will be filled with states whose out-arcs are divergent, i.e., point to distinct states.

What we need is a model which allows for *both* convergence and divergence: divergence because the facts demand this possibility, and convergence because we wish to assign a smaller description length to morphologies composed primarily of converging states.

Our solution to this eat-your-cake-and-have-it-too problem is an automaton composed in the following way. Each state S_i consists of three components: (1) S_i contains a list of one or more morphemes – or rather, of *pointers* to morphemes in a collection of morphemes (i.e., strings from an alphabet A); we refer to this list as the state's *morpheme-choices* (and this is typically *two* or more, in fact); (2) each morpheme-choice may (but need not) be explicitly associated with a unique arc to another state, which we may call a *morpheme-specific arc*; (3) a state may (but need not) be associated with a *common* arc to another state; this *common arc* is interpreted as being associated to all morpheme-choices that are not explicitly marked as associated with an arc. (Only accepting states may have morpheme-choices that are associated with neither a common arc nor a morpheme-specific arc.) It is in effect the *default* transition associated with each morpheme choice that does not have an arc explicitly associated with it. A simple example from English will clarify these notions.

English morphology includes a state composed of count noun stems in English, such as *book, car, chair*, etc, which point to a state associated with the morphemes *s* (plural) and *Ø* (singular), as in (6). English morphology also includes a state composed of weak verb stems, such as *laugh* or *walk*, which points to a state whose morpheme choices are *Ø*, *ed*, *ing*, and *s*, the four regular verbal suffixes.

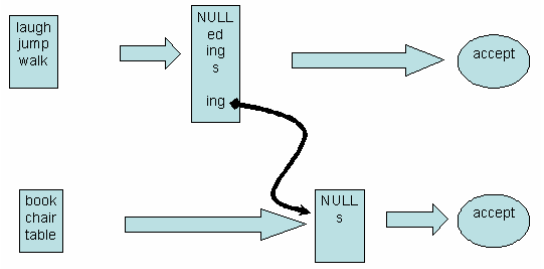
(6)



We adopt the convention that block arrows, as in (6), represent *common arcs*, associated with all morpheme choices to their left which have no morpheme-specific arcs. There are no morpheme-specific arcs indicated in (6).

In (7), however, we indicate the fact that in English morphology, there is a second verbal suffix *–ing* which changes a verb into a noun, as we see in words like *findings, misunderstandings, shootings*, etc. This derivation suffix has a morpheme-specific arc associated with it, as indicated in (7).

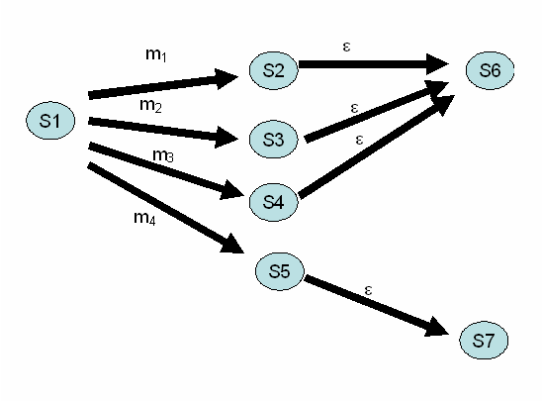
(7)



We call the states in our model *morpheme-choice states*, and each can be thought of, if you

please, as having an internal architecture made up from a familiar arc-emission FSA, as in (8), but we will calculate the description length of the state in a fashion that counts the *common arc* only once, and hence gives priority to analyses in which arcs leaving one state all converge to a single (other) state.

(8)



An MDL model for a morpheme-choice morphology computes its length as follows. Let us use the convention that #(X) refers to the *count* of a set X; X might be the set of words, for example. This convention allows us to consider either type count or token count, as long as we are consistent; this choice will give rise to two somewhat different models. For purposes of expositional simplicity, we make the assumption here that all words are deterministically parsed by the morphology.

Each word in the corpus is associated with a (unique) path from the initial node of the morphology to the (unique) final accepting state. Each state *S* keeps track of how many words' paths pass through it (= #(*S*)) and, for each morpheme choice m_i associated with it, how many times the choice m_i, is taken at that state, which is #(m_i, *S*). Each common arc $a_{i,j}$ and each morpheme-specific arc $b_{i,j}$ passing from State *i* to State *j* keeps track of how many paths pass through them: #($b_{i,j}$ at State *k*) is equal to #(m_i at State *k*), since a morpheme-choice may have no more than one morpheme-specific arc. With a common arc $a_{i,j}$, #($a_{i,j}$) is equal to #(S_i) less the count of all of the morpheme-specific arcs of State S_i.

The probability assigned to a word (which is to say, to the path its analysis takes through the morphology) is the product of the choices made at each state, so the probability of choosing an arc $a_{i,j}$ is $\frac{\#(a_{i,j})}{\sum_k \#(a_{i,k})}$.

The information content of the morphology is the sum of the information content of its component states. This consists of three subparts: a list of pointers to morphemes (these are the morpheme choices), usually but not always a common arc, and possibly a set of morpheme-specific arcs.

The state's morpheme choices are pointers to morphemes on a list; this decision allows the morphology to be simpler if it permits the same morpheme to appear in several different locations (states) in the morphology. A particular morpheme *m* which is associated with the i^{th} morpheme choice of a state S_j has a count #(*m*)associated with it that is equal to the sum of the counts of all of the morpheme choices that point to it. That is, #(*m*) = $\sum_S \sum_i \#(m_i, S)\delta(m, m_i)$ where δ(m_i, m_j) = 1 iff i = j. If T = $\sum_i \#(m_i)$, then the information content of m_i = $\log \frac{T}{\#(m_i)}$, and the length of a state's morpheme choices is a sum of the information content for each morpheme choice.

The length of an arc (either common or morpheme-specific) to a state S_j is based on how many word-paths pass through that state, which we have called #(*Sj*). If we Z = $\sum_j \#(Sj)$, then the length (information content) of an arc $a_{i,j}$ is equal to log $\frac{Z}{\#(a_{i,j})}$.

4 Bootstrapping the search for a corpus's morphology using successor frequency and Patricia tries

An MDL-based system for unsupervised learning of morphology is characterized by an explicit objective function, a bootstrap heuristic for finding a rough and ready initial morphological analysis, and a set of incremental heuristics which suggest improvements to the morphology subject to testing by the objective function. The bootstrap heuristic is important simply because the space of possible morphological analyses of a set of data is astronomical, and incremental searches for improvements in a morphology in practice require that the currently hypothesized morphology not be wildly different from the correct analysis. The incremental heuristics need not be intelligent; their responsibility is to suggest changes to the morphology (add a suffix, change the location of the stem/suffix cut, etc.) which can then be evaluated by the MDL-based objective function. In the limit, these modification heuristics could be very dumb, but there is no particular benefit from taking that strategy.

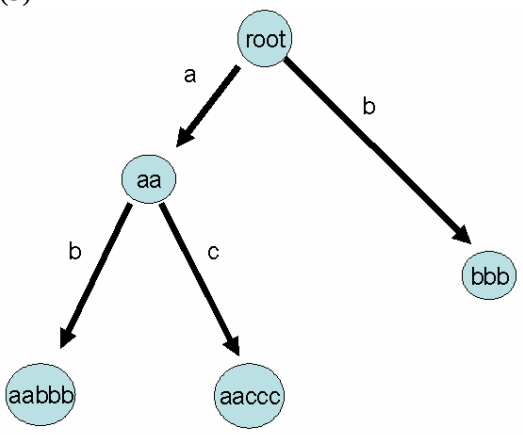
One of the most effective bootstrapping heuristics for establishing an initial morphology depends on an insight of Zellig Harris (1955, 1967). Using the term "prefix" in the computer science sense – that is, a string P is a prefix of a string S iff S=PX for some (possibly null) string X – then Harris proposed a "successor frequency" function on all prefixes P in a corpus, defining the successor frequency S(P) as the number of distinct letters l_i such that the concatenation Pl_i is a prefix of some word in the corpus (we assume by convention that each word is terminated by a designated symbol such as '#'). Thus the successor frequency of a prefix is the number of alternative ways that the prefix can be continued, given the words of the corpus. One can similarly define the "predecessor frequency" for all suffixes of words in the corpus.

Harris believed that some simple function based on successor and predecessor frequency would allow for automatic detection of morpheme boundaries; Hafer and Weiss (1974) showed that this belief was incorrect. Goldsmith (2003) shows how successor frequency can be combined with a signature-based model to produce a useful bootstrap heuristic, roughly as follows. Find the right-most peak of successor frequency for all words in the corpus, and divide each word w_i with such a peak into two pieces, L_i and R_i. Without giving any particular significance to the name, we may refer to *L* as the *stem*, and R as the *suffix*. A given stem L_i may be the first piece in the analysis of several words $\{w_j\}$; in such a case, associate all of the suffix R_j's with that stem L_i, and we may refer to that set of R_j as L_i's *signature*: it is the set of distinct ways that L_i can be completed, and is an initial guess as to L_i's true possible suffixes.

We then count the number of stems a given signature is associated with, and drop any signatures which occur only once. We accept only those divisions of words into a piece L and a piece R which conform to a signature that satisfies this test. This heuristic thus consists of a successor frequency peak test and a signature validation test.

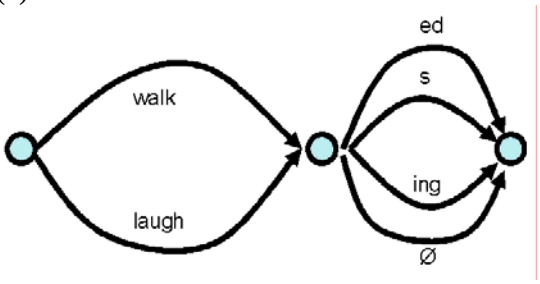
The natural way to store words computationally in order to easily compute successor frequency is as a Patricia trie (Morrison 1968), a compressed trie in which all nodes in the trie have at least two daughter nodes. Thus a Patricia trie storing the strings *aabbb*, *aaccc*, and *bbb* has a structure as in (9). The successor frequency of a prefix represented as the key at a node is the number of daughter nodes, and any other prefix has a successor frequency of 1; predecessor frequency is similarly captured by the number of daughter nodes on a Patricia trie built of the mirror-image of all words. We will henceforth say *trie* when we mean *Patricia trie*.

(9)



Under certain ideal conditions, a trie representation of data might bring one close to a good morphological analysis. For example, given the data D = { *walk, walks, walked, walking, laugh, laughs, laughing, laughed* }, a trie will be constructed in which a node with key *walk* and a node with key *laugh* will be daughters to the root, and from each will depend nodes for *Ø*, *ed*, *ing*, and *s*. It is well-known (Hopcroft and Ullman 1979) that a minimal FSA can be found which generates the same language as a trie, viewed as an FSA, essentially by collapsing the terminal nodes (which are the accepting states), and then successively collapsing subgraphs low in the trie which generate identical sets of strings. For example, the trie which generates D above will contain two subgraphs which generate { *Ø, ed, ing, s* }, and they can be collapsed, to create a minimal FSA as in (9).

(9)



Altun and M. Johnson 2001 and H. Johnson and Martin 2003 exploit this observation to build a minimal FSA from a trie as a morphological model.[3]

[3] Relatively few details are offered in Altun and Johnson 2001. They suggest that their model is based on MDL, but it appears to us that their calculation of the length of the model is limited to counting the number of nodes, placing more computational weight on deciding whether two nodes should be collapsed based on the impact of this collapse on the total probability assigned to the corpus. They suggest that during each update of the FSA, they calculate the change in the description length for each of the approximately n^2 possible state collapses, where n is the number of states, and they report that their system learns the correct morphology for Turkish when presented with 240,000 words, but do not give precision or recall figures.

5 Motivation: to better obtain layered morphology; inflectional and derivational morphology

One of the primary reasons for making the switch to an FSA as the basic model for a morphology is that it is superior as a model for the *layered* characteristics of natural language morphology. What do we mean by *layered* characteristics? This is a notion most easily explained by example, and we have seen several so far: cases where a subpart (typically a large subpart) of the morphology is best viewed as a sequence of states in a state-emission FSA: in short, where the composition of a word consists of walking through a fixed set of states, and choosing one morpheme as we pass through each state.

There appears to be a natural linguistic interpretation of this structure as well, which we are exploring. The basic idea is this: natural language inflectional morphology tends to consist of a sequence of largely non-overlapping realizations of morphosyntactic information, and this is best analyzed as a sequence of convergent morpheme states in an NFSA. We saw a couple of examples of this earlier, in examples (1) through (5); good examples of it are the suffixal options in French and Spanish adjectives(3 and 4a), and the prefix options in the Swahili verb in (1).

Derivational morphology, by contrast, is *divergent*, in the sense that a derivational affix typically shifts the traversal through the FSA from one sequence to another. Each sequence is typically devoted to the morphological realization of one lexical class, and much (though not all) derivational morphology involves a morpheme whose function is to change a word from one category to another (or in the present metaphor, to shift the path that we take in the FSA from one convergent sequence to another).

6 Our algorithm

In developing a morpheme-choice based morphology from a trie, our first goal was to ensure that we could incorporate as much of the intelligence of the signature-based MDL analysis as possible. Our procedure is as follows.

Given a trie of the vocabulary, consider each node in the trie as a possible morpheme break (as noted above, each node in the trie is a case where the successor frequency is greater than or equal to 2), starting with the 5th letter of the word.

At each such point, consider the set of sequences of letters that follow that point; this corresponds directly to the signature. For example, if the words *book* and *books* are in the corpus, then a node in the trie will occur that is pointed to by the string *book* and which encodes the sequence of letters *Ø* and *s*. We keep track of these signatures in a separate data structure, and evaluate them as follows.

The credibility of a signature is based on three characteristics: the number of stems that precede it, and the number and length of the individual suffixes in the signature. Suffixes of length 1 are the least reliable, and signatures made up of suffixes that are all of length 0 or 1 are highly unreliable. Nonetheless they may be real and valid (indeed, the signature *Ø.s* is valid in English, French, and Spanish; the signature *Ø.e* is valid in French, and the signature *a.o* is valid in Spanish.). We accept signatures with such short suffixes only if they are supported by at least 25 stems, and signatures with longer suffixes if they are supported by at least 5 stems. We cut words into morphemes on the basis of the signatures we have identified in this way. Thus, in a corpus of French with the adjectives *petit*, *petits*, *petite*, and *petites*, we will make cuts after *petit*, and also after *e*, and so on.[4]

We build a finite state automaton using morpheme-choice states, as we have described above, on the basis of these morpheme cuts, and minimize the number of states in the following way. All accepting states are identified as a single state, and then we successively collapse all states whose set of morpheme-choices (its "signature", in effect) are identical and which point to the same following states.

7 Collapsing states and suffix-sublanguages; poor results with probability of corpus.

We have been interested for quite some time in morphological analysis of French and Spanish. The languages are similar in a number of respects. Both have systems of adjectives in which an adjectival stem is followed by either a masculine or a feminine gender marker, followed by either a singular or a plural number marker. See (10), where the tables should be read as menu-style FSAs : construct an adjective by choosing one morpheme from each column.

(10). a. French adjectives

Stem	Gender	Number
petit "small"	Ø (masculine)	Ø (singular)
grand "large"	e (feminine)	s (plural)

Example: *petit* (masc. sg.), *petite* (fem. sg.), *petits* (masc. pl.), *petites* (fem. pl.)

b. Spanish adjectives

Stem	Gender	Number
pequeñ "small"	o (masculine)	Ø (singular)
lind "pretty"	a (feminine)	s (plural)

Example: *lindo* (masc. sg.), *linda*(fem. sg.), *lindos* (masc. pl.), *lindas* (fem. pl.)

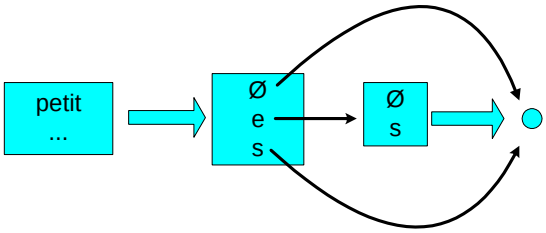
In our earlier signature-based analysis, these forms have been analyzed as displaying four suffixes: French *Ø*, *e*, *s*, and *es*; Spanish *o*, *a*, *os*, and *as*. This is not linguistically correct, but follows from some not unreasonable assumptions that are built into the design.[5]

[4] This differs from the algorithm in Goldsmith 2004, which does not make a cut unless the successor frequency function is unambiguous about the location of the morpheme break, i.e., a break is made at location *i* only if the successor frequency function at position *i-1*, *i*, and *i+1* is (resp.) 1, N, 1 for some N>1. This conservative cutting inhibits it from finding word-internal suffixes of length 1. It was motivated by a wish not to place a morpheme boundary both before and after the *i* in words such as *construction*, which typically occur in corpora both with *construct*, *constructs*, and *constructed*, on the one hand, and with *constructing*, on the other.

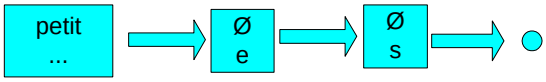
[5] Because of the strategy described in footnote 4, splitting adjectives into stem plus suffix was

In our present FSA-based morphology, adjectives with all four forms in the corpus emerge from the collapsing of the trie into a minimal FSA as in (11), which is not what we would hope for (what we hope for would be the structure in (4a)). When, as in (11a), a state S points to both a daughter state and the daughter's daughter(s), and is hence divergent, then we test the subgraph, to see whether there is an alternative restructuring in which S can be rewritten as a convergent state. We do this by removing the morpheme-specific arcs to the granddaughter node, and see what strings now fail to be generated, and check to see if this set of strings can be generated by adding one or more morphemes to the state on the assumption that the state becomes a convergent state. This will typically decrease the description length of the grammar, since we will be certain to reduce the number of pointers in the grammar in so doing. This incremental heuristic offers us the alternative, and preferred (because shorter) structure in (11b).

(11)
a. before

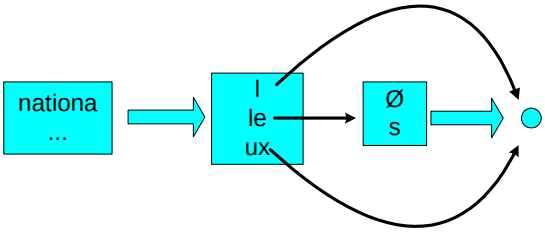


b. after



We in effect identify sequences of converging states in our FSA as islands of reliability.

In work in progress, we are attempting to use these convergent sequences of states in order to learn allomorphy, which is rampant in French morphology. Adjectival stems that end in the morpheme *–al-* do not appear with the masculine suffix Ø and the plural suffix –s-; instead, the combination *–aux* is found (e.g., *nationaux* 'national, masc. pl.'). This allows us to arrive at an "imperfect" submorphology of this form:
(12)



Using a similar strategy as that which drove the restructuring from (11a) to (11b), we are working on a method to reduce (12) to (11b) followed by a rewrite of *al-s-#* to *aux*. This method depends on the reliability in general of layered, convergent subparts of the morphology.

References

Altun, Y. and M. Johnson (2001). Inducing SFA with epsilon-transitions using Minimum Description Length. Finite State Methods in Natural Language Processing 2001 ESSLLI Workshop, Helsinki.

Goldsmith, J. (2001). "Unsupervised Learning of the Morphology of a Natural Langauge." *Computational Linguistics* 27(2): 153-198.

Goldsmith, J. a. M. B. (2002). Using eigenvectors of the bigram graph to infer grammatical features and categories. *Proceedings of the Morphology/Phonology Learning Workshop of ACL-02.*, Philadelphia PA, Association for Computational Linguistics.

Goldsmith, J. (2004). "An algorithm for the unsupervised learning of morphology." Manuscript.

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

discovered initially when only two suffixes occurred on a single stem (which might be any two of the four combinations, including Spanish "as" or French "es" as a single item). These four possibilities, having been "discovered", would then be found in groupings of 3 and 4 on other adjectival stems. From some practical points of view, such as stemming for document retrieval, this analysis is perfectly acceptable – but it is not linguistically correct.

Harris, Z. (1955). “From Phoneme to Morpheme.” *Language* 31: 190-222.

Harris, Z. (1967). Morpheme boundaries within words: report on a computer test. *Transformations and Discourse Analysis Papers* 73. Reprinted in Harris 1970.

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

Hopcroft, J. E. and J. D. Ullman (1979). *Introduction to automata theory, languages, and computation*. Reading, Mass., Addison-Wesley.

Johnson, H. and J. Martin (2003). Unsupervised learning of morphology for English and Inuktitut. Human Language Technology Conference, Edmonton.

Jurafsky, D. and J. Martin (2000). *Speech and Language Processing.* Upper Saddle River, NJ, Prentice Hall.

Koskenniemi, K. (1983). *Two-level morphology: a general computational model for word-form recognition and production*. Helsinki, Department of General Linguistics, University of Helsinki.

Morrison, D. (1967). “PATRICIA--Practical Algorithm To Retrieve Information Coded in Alphanumeric.” *Journal of the ACM (JACM)* 15(4): 514-534.