Abstract

A system for inducing the phonology and inflectional morphology of a natural language

Scott Nathanael McClure
2011

This thesis presents a machine learner that uses morphologically tagged data to induce clusters of words that take on similar inflections, while at the same time identifying sets of morphological rules that are associated with each cluster. The learner also identifies simple phonological alternations. This work is significant because it uses a relatively simple framework to discover prefixes, suffixes, and infixes, each of which may be associated with one or more morphological feature values, while simultaneously discovering certain simple phonological alternations. The learner makes use of Bayesian principles to determine which grammar, out of several, is the most apt, while the search is performed in a greedy manner: starting from an initial state in which every lemma is assigned to its own inflection class, the learner attempts to merge existing inflection classes while improving the posterior probability of the hypothesis. As these inflection classes are merged, the learner develops a more and more accurate picture of the morphological rules associated with each inflection class, as well as of the surface-true phonological alternations that apply throughout the language. This work demonstrates that many of the principles of word formation posited by linguists can indeed be induced using probabilistic methods, and it also serves as a key step in improving the level of detail in the grammars of word formation returned by an automatic learner.

Dissertation Directors: Gaja Jarosz and Stephen R. Anderson

A system for inducing the phonology and inflectional morphology of a natural language

A Dissertation
Presented to the Faculty of the Graduate School
of
Yale University
in Candidacy for the Degree of
Doctor of Philosophy

by
Scott Nathanael McClure

Dissertation Directors: Gaja Jarosz and Stephen R. Anderson

May 2011

Acknowledgments

The carving over the door at the Yale University Department of Linguistics reads φιλότης δυναστεία γλυκητάτη, and Argyro tells me that it means something along the lines of "Friendship is the sweetest regime". I cannot express how grateful I am that my advisors at Yale always kept this in mind. Through their efforts, they turned my years at Yale into the most valuable experience that I can imagine. I owe my sincerest thanks to Gaja Jarosz and Steve Anderson not just for imparting some of their curiosity and tenacity to me, but also for always doing so in a spirit of patience and friendship.

I am also grateful to Bob Frank for his commitment to computational linguistics at Yale and for his many incisive comments. Many other scholars at Yale University and Haskins Laboratories were also generous with their time and efforts, and so doing they made New Haven a fruitful place to work and study; among them are Dana Angluin, Masha Babyonyshev, Louis Goldstein, Dasha Kavitskaya, Jelena Krivokapić, and Tine Mooshammer. Chris McDaniel also did a great deal to make Yale a wonderful place to work and study.

It has also been my pleasure to work with a group of students at Yale who are without peer in intelligence, integrity, and general-purpose likability. I am sad to be leaving the place where I found such unmatched companionship among Raj Dhillon, Argyro Katsika, Jennie Mack, Kelly Nedwick, Mike Proctor, Jodi Reich, and Will Salmon. Special kudos go out to Erich Round, who, in addition to extending his

friendship to me, was kind enough to explain a great many things about phonology and morphology, and to Yael Fuerst, the best classmate I could ever imagine.

I must also offer my sincere thanks to those who have had a hand in developing the software and linguistic resources that have made this research possible. Countless contributions—many of which have been made anonymously, and many of which have been made by unpaid volunteers—to resources released under the various GNU, BSD, and CreativeCommons licenses and into the public domain proved invaluable during the completion of this thesis. In particular, I must extend thanks to those responsible for the Perl programming language, the Perseus Digital Library Project at Tufts University, the Quranic Arabic Corpus at the University of Leeds, and the WikiMedia Foundation's Türkçe Vikisözlük and Türkçe Vikipedi projects.

Finally, I must thank my family. My parents, James and Betsy, and my sister, Kate, have provided immeasurable support, and three felines—Jimmy Jazz, Pablo, and Ferdinand—have brightened many an hour before and after work. Finally, I must offer my deepest and sincerest thanks to my spouse Madeleine. To say that this thesis would not exists were it not for her love, support, and extreme patience would be only a faint echo of the truth.

Contents

Chapter 1

Introduction

This thesis is concerned with the regularities that govern the relationships between word forms in natural human languages. These regularities include morphological relationships, such as singular∼plural pairs in English—such as *pigeon*∼*pigeons*, *parrot*∼*parrots*, and *goose*∼*geese*, for example—but they also include phonological relationships, such as the [t]∼[ɾ] alternation that occurs in the phonological forms of *write*∼*writer*, and the [z]∼[s] alternation in the phonological form of the plural marker that appears in *pigeons* and *parrots*. This thesis presents an automatic learner that uses morphologically tagged data to induce clusters of words that take on similar inflections, while at the same time discovering the morphological rules at play in each cluster and simple phonological alternations that apply across clusters.

This chapter sets the stage by introducing previous research on morphological induction, and by briefly explaining the goals of the present system. Chapter 2 outlines the objective function that the learner uses to compare one grammar against another and to determine which is a more appropriate hypothesis, given the training data, while chapter 3 explains the search procedures that the learner uses to walk through the space of possible grammars in the attempt to find an apt hypothesis. Chapter 4 discusses three case studies, conducted on Classical Arabic, Classical Latin,

and Modern Turkish. Finally, chapter 5 discusses some of the theoretical issues that this work raises, while chapter 6 explains the significance of the present system and indicates the ways in which future research can build upon it.

1.1 A preliminary statement of the problem

The system presented here attempts to take labeled training data—that is, data in which word forms in a natural language have been associated with tags specifying the lemma to which each word form belongs, as well as the morphosyntactic information carried by the form—and to develop a grammar of inflectional morphology similar to that which a human linguist would posit. Such a grammar captures the knowledge that a human speaker of the language would need in order to deploy the inflectional morphology of that language. One hopes that the conventions that human linguists and the present system use when stating grammars in terms of certain rules and representations are similar to the rules and representations that human speakers do, in fact, draw upon while using natural language, but this is not the central issue of this work. The task of grammar induction, in the context of inflectional morphology, involves clustering words into groups, called inflection classes, where all members inflect in a similar way; it also requires identifying the underlying root form of each word, and identifying the inflectional rules that apply within each inflection class. Finally, it involves identifying certain phonological generalizations which exist in the language, and which may allow certain inflection classes to be merged once the generalization is recognized. Chapter 2 presents the specifics of the grammatical formalism in greater detail, while chapter 3 discusses the way in which the present system searches the space of possible grammars for an apt hypothesis grammar for the training data. The following sections present a more detailed picture of the theory of word formation that the present system assumes; from there, the chapter goes on to explore previous work in automatic induction of natural language morphologies.

1.2 Linguistic theories of word formation

According to the model of morphology suggested in Anderson [7], word formation involves the successive application of morphological rules or processes upon a base form. Derivational rules apply so as to change the grammatical category, inflectional category, argument structure, and semantic representation of the base on which they apply. This is in contrast with inflectional rules that apply so as to realize feature values found in a form's morphosyntactic representation. The present work is concerned with the regularities of inflectional morphology, rather than of derivational morphology, and also with certain phonological generalizations that are highlighted by the application of inflectional rules.

The theories of inflectional morphology suggested in Anderson [7], Stump [61], and others have the advantage that a morphological analysis does not seek to assign sound-meaning correspondence to parts of words. Considering the facts about non-concatenative morphology in natural languages, trying to find sound-meaning correspondence at the level of the morpheme is not going to be successful, in the general case—as illustrated by the facts about interdigitation, consonant and vowel mutation, and the morphological deployment of tone. In Stump's terms, derivational and inflectional morphology are both viewed best as "inferential" processes rather than as "lexical" elements, meaning that they are more appropriately modeled as changes that apply to the phonological forms of words than as elements that are introduced into the phonological forms of words. To be sure, prefixation and suffixation make up the great bulk of the morphological processes in the languages of the world, but these processes are not exhaustive, and a complete theory of morphology needs to allow for other kinds of processes, including infixation and mutation.

Furthermore, facts about multiple exponence and null exponence suggest that inflectional morphology is indeed "realizational", in the sense that particular inflectional processes apply so as to express the morphosyntactic feature values carried by

a particular form. This is in contrast to "incremental" theories in which inflectional processes apply to a word, with each process adding to the set of morphosyntactic features carried by the form, in addition to changing its phonological form. Again, see Stump [61] for a discussion of the terms "incremental" and "realizational" as applied to inflectional morphology. Consider briefly, though, the challenges that multiple exponence and null exponence pose for incremental theories of inflectional morphology. Multiple exponence describes situations in which a single morphosyntactic feature is, apparently, marked at several different positions within a word, and null exponence describes situations in which an inflectional rule appears to mark no particular morphosyntactic feature at all. Null exponence is seen, for example, in many Indo-European languages, in situations where a thematic vowel intervenes between the root and the ending in certain verb forms. This thematic vowel cannot be associated with any morphosyntactic feature, but in many cases it is still plausible to argue that it belongs to inflectional system of the language. It is difficult to analyze these phenomena if one assumes that every inflectional rule must add at least one new morphosyntactic feature value to a word's representation; the alternative approach, in which inflectional rules apply as the reflex of morphosyntactic feature values that are assigned to words by an independent process, is far more appealing.

Note that Roark and Sproat [55] argue that incremental and realizational theories are in fact equivalent. It is true that any incremental model of morphology can be converted into a realizational model that assigns the same morphological features to the same word forms, and, in a similar way, that an equivalent incremental model can be built for any realizational model. However, facts about multiple exponence and null exponence in natural languages often mean that a realizational model can account for naturally occurring data much more straightforwardly than the equivalent incremental model. While the two systems may be formally equivalent, realizational models seem to capture the situation in natural languages much more aptly, often with

a smaller and more natural set of morphological features: the equivalence that Roark and Sproat describe relies on the set of morphological features in the language being fungible—sometimes bizarrely so—in order to accommodate an incremental analysis that is equivalent to a particular realizational analysis. It is much more persuasive to argue that morphological features have some standing in the syntax and semantics of the language, and that a reasonable morphological analysis ought to reflect this truth.

Among theories of morphology that take inflectional morphology to be processes that apply to base forms in response to the morphosyntactic feature values that the forms carry, there are different ways of expressing regularities and sub-regularities that exist among inflected forms. For the purposes of the system being developed here, the assumption is that each base or root is assigned to a single inflectional category; the set of inflectional rules that are active for any one inflectional category are independent from the rules that are active in all other inflectional categories in the language. It is also assumed that the inflectional rules active in each category are organized into rule blocks, and that within any given block, only the most specific rule whose conditions are met will apply. See Anderson [7] but especially Stump [61] for similar, but somewhat more complex, treatments of inflectional categories, rule blocks, and disjunctive rule application. Note in particular that the present system, unlike the model proposed by Stump, does not recognize paradigms as linguistic objects: each root belongs to a particular inflectional category, and all roots that belong to the same inflectional category are inflected according to the same set of rules, but there are no rules of referral that relate more than one cell in a paradigm. There are also no other generalizations, such as those discussed in Carstairs [11], that apply over paradigms in the system being presented here. In the present system, any given rule is assigned to a particular inflection class, and it applies when certain morphosyntactic feature values are found on a word. There is no attempt to suggest

that a single rule ever applies in several different inflection classes, or that a single rule marks several different sets of morphosyntactic feature values.

Another aspect of word formation is the role of phonological generalizations in determining the surface forms provided by a grammar. To cite one very well known example, consider the regular plural marker in English, /z/. When this marker appears on a word ending in a vowel or a voiced sonorant, the form [z] is preserved in the surface form, as happens in *puffins*, but when it appears on a word ending in a voiceless consonant it surfaces as [s], as is the case in *parakeets*. This last generalization is superseded by a generalization in which [ə] (or, for many speakers, [ɨ]) appears to the immediate left of the plural marker, in cases where the plural marker would otherwise fall next to a sibilant: this is the situation in *finches*.

Clearly, it is possible to construct a theory of word formation that does not recognize any connection between the plural marker in *puffins*, *parakeets*, and *finches*. Under a theory in which inflectional classes are recognized, it is possible to assign *puffin*, *kiwi*, and *dove* to a single class in which the plural is marked with /z/, *parakeet*, *duck*, and *lark* to a separate inflection class where it is marked with /s/, while *finch*, *ibis*, and *thrush* are all assigned to another in which it is marked with /əz/ or /ɨz/. This theory is adequate from the point of view of generating the correct surface forms, but it fails as a description of a speaker's knowledge of English for several reasons. First, it fails to capture a clear phonological similarity between [z], [s], and [əz]. Each of these three strings is clearly only a small edit away from the others, and it would not be unreasonable to expect a phonological generalization to be responsible for associating them with one another. Alternatively, one might say that [z], [s], and [əz] are close to one another in some formalized view of the articulatory space, or even the perceptual space, but for the purposes of this argument one might as well think of segments as vectors of phonological features, with each occupying a particular point in the feature space.

This relationship between [z], [s], and [əz] stands in contrast with the relationship between exponents of the same feature values that are drawn from more clearly distinct inflection classes—for example, the edit distance separating *-ibus* and *-orum*, the plural genitive markers in two inflection classes in Latin, is quite large, no matter the edit distance metric that one uses. Second, this assignment of *puffin*, *parakeet*, and *finch* to distinct inflection classes is clearly based on a systematic fact about the phonological form of the root, rather that simply an arbitrary fact that can cross-cut roots of various phonological forms. Together, these facts indicate that the relationship between the markers [z], [s], and [əz] is based on facts about the phonology of the language, and that it is not simply an arbitrary fact about the plural marker. On the basis of these facts, then, it makes sense to suppose that words whose plural markers surface as [z], [s], and [əz] might in fact belong to the same inflection class, and that a fact about the phonology of the language—rather than its inflectional morphology—is responsible for the relationship between these forms.

Without a doubt, of course, natural languages contain many phonological generalizations that cannot be captured just within the model of word formation, since they depend on the context of words in utterances. Such generalizations cannot be modeled as part of word formation, but instead must be modeled as part of the "post-lexical" or "post-cyclic" phonology. That being said, there is a huge set of phonological generalizations in natural languages—including the [z], [s], and [əz] example from English—in which the generalization, while clearly phonological, rather than morphological, can be captured as part of the process of word formation.

There are a number of ways in which word-level phonological generalizations and morphological processes can be related. One of the most empirically appealing ways in which the two grammar components can be related is for phonological generalizations to apply immediately after each application of a morphological process—on this point, see Kiparsky [38], [39], [40] and Kaisse and Hargus [35], among others.

In an ideal world, the present system would model this kind of cyclic rule application; however, the present system makes a simplifying assumption in that it takes all phonological rules to apply once, simultaneously, at the end of a derivation, and that all phonological rules are surface true. This is indeed a great simplification on the kinds of generalizations described in Chomsky and Halle [15], Anderson [5], Kiparsky [38], and elsewhere in the field of generative phonology, but it allows certain basic facts to be captured. (See Prince and Smolensky [52], however, for a theory of phonological generalizations couched in terms of ranked, violable constraints on forms, and especially the discussion in McCarthy [48] regarding the kinds of surface-true and opaque generalizations that can be captured within such a system.)

The present system also assumes that phonological generalizations can be stated as a certain relatively small and simple subset of finite state automata, or regular expressions. This assumption follows the work of Johnson [32], in which it is shown that a comprehensive set of phonological generalizations can be stated in terms of finite state automata or regular expressions. It may or may not in fact be true that all phonological generalizations in the languages of the world can indeed be stated in terms of regular languages, but the assumption of the present system is that phonological generalizations can be stated in terms of a very restricted set of finite state automata, that these automata apply together after all morphological processes have taken place, and that their application is always transparent, rather than opaque.

1.3 Previous work on learning morphology and phonology

The previous work on learning morphology can be seen in terms of several categories, which are helpful, at least up to a point, for making sense of the goals that each author has for his or her own work, and for understanding the assumptions that each author makes about the morphology and phonology of natural languages. The category with

the longest history is one in which researchers have tried to find a procedure by which words or utterances can reliably be divided into morphemes without appealing to any outside information. This approach might fairly be termed "structuralist" because of its focus on finding patterns within a particular kind of representation of the speech signal, rather than on determining what set of knowledge is necessary to generate these forms.

The structuralist and "knowledge-free" approaches to learning morphology are very similar, both in terms of the kind of training data that they use, as well as in terms of the grammars that they induce. Indeed, both the structuralist and the knowledge-free techniques are interested in what information about morphology can be obtained by presenting a machine learner with a corpus of untagged transcriptions. The main difference is that so-called knowledge-free techniques usually bring in some kind of statistical information, such as token frequency or comparisons of context words, that can be used to get information about the identity of lexemes in the training data, beyond what can be obtained from simple phonological or orthographic resemblance. This stands in contrast with the structuralist approaches, which generally do not attempt to enrich their representations with any kind of information beyond simple tokenization. These two techniques for inducing morphology are then compared with techniques that employ Bayesian models—often trained with tagged, rather than untagged, data—and with a relatively disparate set of systems that all make phonological generalizations in addition to morphological generalizations.

1.3.1 Structuralist approaches to morpheme segmentation

Work in structuralist approaches to morpheme segmentation shares the goal of dividing an orthographic representation, a phonemic representation, or some other phonological representation into a sequence of morphemes. This work assumes, first and foremost, that word formation is a matter of morpheme concatenation. It also as-

sumes that one has access to an appropriate kind of representation over which the segmentation procedure may be applied. For Harris [31], this is the phonemic representation, which, according to Harris's assumptions, can be recovered unambiguously and in full from the speech signal; the work that follows his usually makes use of orthographic representations instead. The kinds of representations that are used in this structuralist work are significant, because morphological generalizations are always made over representations which are present in (or unambiguously recoverable from) the representation given for training purposes—there is never any attempt to find a more abstract representation, such as an underlying phonological representation.

Harris's [31] procedure for finding morpheme boundaries is this: take a fairly large set of phonemic representations of utterances from the language in question. For each utterance, for n = 1 to the length of that utterance in phonemes, consider the string of phonemes that run from position 1 to position n; then find how many times that string appears at the beginning of other utterances in the set of data being considered, and find how many different phonemes can follow it at position (n + 1). The points in an utterance where there is a high number of continuations are likely to be morpheme boundaries, while the points with only a few continuations are likely not to be boundaries.

Harris gives a few variations on this procedure to make it more accurate. The most obvious is to run the procedure backwards, so that one looks at strings that begin at the right edge of an utterance and continue to the left. This allows Harris to identify morpheme boundaries correctly in cases where an element—perhaps a stem—tends to be followed by a relatively small number of other elements—perhaps drawn from a small set of inflectional markers with which that stem is prone to appear. Another suggestion that Harris makes is to look at how many continuations are available, on average, from position (n + 1) as compared with the number of continuations available at position n. The idea here is to catch situations in which the phonemes

at positions n and (n + 1) belong inside the same morpheme, even though there is a peak in the number of continuations after position n. As Harris explains, there is often a peak in the number of continuations available after a vowel in English, but this can be attributed to the phonotactics of English, and it is not related to morpheme boundaries. These spurious peaks that appear after vowels can be eliminated if one looks ahead to see how many continuations are possible at position (n + 1). If this figure is quite low, in the neighborhood of 1 or 2, a boundary should not be placed after position n. The reasoning here is that while its true that a peak in the number of continuations available at a certain point often indicates a morpheme boundary, it is also the case that the number of continuations available tends not to drop all the way down to 1 or 2 by the time one arrives at the second phoneme in the morpheme.

Harris's work is valuable for laying the groundwork for a huge amount of work in morpheme segmentation and word segmentation that considers the cues provided by bigrams, trigrams, n-grams, and mutual information measured over segments. The fact is, though, that Harris's procedures depend on the principle that a phonemic representation of an utterance is the linguistically significant representation. If one assumes that there is a linguistically significant level of representation that includes contrasts which are sometimes obscured in the phonetic representation—such as the underlying representation of generative phonology—Harris's procedures are less useful, since they cannot give an analysis that recognizes regularities in terms of an underlying representation. His procedures also depend on the principle that word formation takes place through the concatenation of morphemes—anything more elaborate simply cannot be recognized by the continuation-counting procedure.

The Linguistica system, described in Goldsmith [25], represents a much more sophisticated technique for searching a corpus for the appropriate boundaries between stems and suffixes. As with the work of Harris, however, the Linguistica system assumes that it has access to the representation over which morphological general-

izations should indeed be made. Although Linguistica has been tested, first and foremost, on orthographic representations, this assumption is similar to Harris's assumption that the phonemic representation, recoverable from the speech signal, is the appropriate representation over which to make morphological generalizations: morphological generalizations are still being made over the same kind of representation that is presented in the training data.

Linguistica takes an unannotated corpus as its input, and it returns a grammar in which stems and suffixes have been identified, and the signature of each stem is specified. The signature of a stem is the set of suffixes with which it associates; in a typical grammar returned by Linguistica, various stems will often be assigned to a single signature in a way that is more or less comparable with the inflectional categories that a human linguist would identify. One key difference, though, is that Linguistica does not account for allomorphy of any kind. Grammars are evaluated relative to one another according to the principle of minimum description length. The description length is taken to be the length of the grammar plus the length of the data being expressed with that grammar; the goal is to make this entire description as short as possible. In essence, one is searching for the best way to compress the data, without loss, according to certain principles that dictate which compressions are possible, and which are impossible. In the case of Linguistica, compression takes place by recording a particular word in the data not in terms of the phones or graphemes that compose it, but in terms of a valid stem-suffix combination. If the stem and the suffix have been recorded elsewhere in the data, one does not need to give the individual phones or graphemes that build them. Instead, a pointer that refers to the full representation of the stem and suffix will express the same information with a shorter overall description.

The system starts by finding an initial hypothesis for the segmentation of each word into a stem and a suffix, either according to a technique that makes use of

n-grams or expectation maximization. (For more on expectation maximization, see Dempster, Laird, and Rubin [20].) At this point, the preliminary division of the word list into stems and suffixes is used to build signatures: sets of stems which associate exclusively with the same set of suffixes are put together in the same signature. The minimum description length principle then comes into play, as the system finds the number of bits that are required to state the grammar plus the number of bits that are required to express the data in terms of that grammar. The length of the grammar is given by a rather involved formula, but the basic idea is this: counting the bits needed to express a string of phones or graphemes is easy; the number of bits needed to express a pointer to a particular stem or suffix depends on the probability distribution over the set of stems or suffixes. To find the number of bits needed to express the training data, one counts the number of bits that are needed to represent each stem and each suffix as a string, and adds this to the number of bits that are needed to give the pointers that refer to those strings in context.

The system uses a hill-climbing search, whereby it finds the description length of the data under its initial hypothesis, and then perturbs that initial hypothesis in several ways in the search for a shorter description of the data. It keeps the changes that result in a shorter description and it ignores the changes that fail to do so. As part of this search, Linguistica finds hypothesized suffixes that can be taken as the concatenation of two other suffixes that have been hypothesized, and it determines whether analyzing those forms as having two suffixes rather than one results in a better description length. It also looks for signatures for which all the suffixes begin with the same character, and it sees whether those characters are better taken as part of the stem. When successful, this procedure can allow several signatures to be collapsed into one.

The grammars that Linguistica gives as output come quite close to the analyses that would be produced by a human, particularly if one allows for the fact that

grammars in Linguistica do not conform perfectly to the kinds of grammars of word formation discussed earlier in section 1.2. The simplifying assumptions in Linguistica include the assumption that one is not seeking to account for phonologically controlled allomorphy, as well as the assumption that one is finding morphological generalizations over a representation that can be recovered directly from the signal, rather than a more abstract representation. Note, though, that Linguistica is prone to the problems posed by sparse data, in which the corpus provides full paradigms for a few words, but only incomplete paradigms for most other words. The danger here is that words that really all belong to the same paradigm will be grouped into distinct paradigms on the basis of which forms happen to be attested, rather than on the basis of true categories; this fact can be attributed to the minimum description length metric used by Linguistica.

In the years since Goldsmith's first papers on Linguistica (such as [25]) were published, a number of authors have sought to extend this work in one way or another. Of particular note is the work of Goldsmith and Hu [24], which demonstrates a technique for reducing the signatures output by Linguistica to morphological grammars defined in terms of finite state automata. This work makes the connection between morphological learning in the Linguistica system and the work on hand-built finite state morphological grammars (see, for example, Beesley and Karttunen [9]) more clear. Several authors have also attempted to use a morphological learner like Linguistica to induce knowledge of phonological alternations or spelling rules, but see section 1.3.4 for a full discussion of these projects and their significance.

The work of de Marcken [18], [19] represents an unusual approach to the problem of finding constituents in linguistic data. His system identifies units at various levels of organization, from clusters of phones that happen to appear together frequently, to morphs, all the way up to words and sets of words that are particularly common in sequence. The idea here is that linguistic structures, from the level of the phone

all the way up to the level of the sentence, are built according to principles that can combine them regularly (or "compositionally", in de Marcken's terms), but that there are some units of organization which, although they are built of regular, composed, pieces, may impart some degree of irregularity or perturbation on the form at that level.

The easiest example to imagine is that of phones being put in sequence to form words: sometimes $\mathrm{p}(\mathrm{phone}_a, \mathrm{phone}_b)$—by which the probability of phone_a being followed by phone_b is notated—can be given by $\mathrm{p}(\mathrm{phone}_a) \cdot \mathrm{p}(\mathrm{phone}_b)$, but in other cases $\mathrm{p}(\mathrm{phone}_a, \mathrm{phone}_b)$ is significantly higher or lower than what would be predicted by $\mathrm{p}(\mathrm{phone}_a) \cdot \mathrm{p}(\mathrm{phone}_b)$. In the case that $\mathrm{p}(\mathrm{phone}_a, \mathrm{phone}_b)$ is particularly high, one might guess that the sequence $[\mathrm{phone}_a,\ \mathrm{phone}_b]$ represents a common word or morph in the language, or at the very least that it happens to be a string from which larger structures are often built. The same principles can be used to talk about larger structures, at the level the sentence, where principles of syntactic organization operate, but the situation is far less straightforward, since it is harder to suppose that structures are built simply by concatenating one unit after another at this level of organization.

As long as one is looking at the principles that combine phones into morphs and words, however, one can more easily suppose that all composition takes place through concatenation, and this allows the problem to be simplified in a convenient way. Rather than search for something that resembles a morphological analysis, de Marcken searches for the set of "items" that allow the corpus to be maximally compressed, where an item is a concatenation of two phones, or the concatenation of two items, or the concatenation of an item and a phone. He shows that the boundaries of these items quite often coincide with word boundaries and morph boundaries. However, de Marcken's items also include information about phonotactics, and at least some information about organization at levels higher than that of the word, and it is not at all obvious which items correspond with morphs, with words, and with some other

level of organization in the training corpus. This work is perhaps most notable in the way that it confronts the problem of regularities at different levels causing interference when one is searching for generalizations at some particular level, but it does not seem that this kind of analysis is particularly helpful for getting insight into the morphology of natural languages.

Johnson and Martin [33] point out that much of the previous work that places morpheme boundaries based on some measure of how many continuations are possible from a given point in a word, such as Harris [31] and some of the knowledge-free work, can be seen as variations on a procedure in which one builds a finite-state machine that can generate full words, and which one then searches for nodes that have a particularly high number of arcs leading either in or out. Such nodes can be taken to be morpheme boundaries. These authors point out that it is actually possible to build a minimal finite-state machine from a list of surface forms in the language, since an algorithm exists for converting a trie into the minimal FSM which can generate all and only the words in the trie, which has one start node and one end node, and in which all sequences which can be continued in the same set of ways share a node at the point from which those identical continuations begin. The authors observe that such an FSM can be searched for nodes that have many arcs leading in (as this is likely the beginning of a morpheme) or for many arcs leading out (as this is likely the end of a morpheme). These nodes with many arcs leading out are essentially what Harris, Goldsmith, and others have looked for, whether or not they have used the terminology of finite-state machines to describe their work. Given the rather low performance that Johnson and Martin's hub-searching methods seem to show for Inuktitut, however, it is not clear that the authors have a technique that is more suitable for agglutinative languages, such as Inuktitut or Turkish, than Goldsmith's Linguistica.

Like most of the previous work on discovering a morphology discussed so far, this

work assumes that word formation is a matter of concatenating morphemes to build words. It is not immediately clear how to apply this work if one is interested in inducing a morphology in which the surface forms of words are taken to be the result of the application of morphological and phonological rules on an underlying representation of a root. Most knowledge-free techniques for inducing morphology make a similar assumption about the kinds of representations over which morphological generalizations are to be made, and about the ways in which words are formed.

1.3.2 Knowledge-free induction of morphology

This work is actually very much like the work of Harris, Goldsmith, and the other authors that take an essentially structuralist approach to the induction of morphology as a segmentation problem. The thing that sets these authors apart is the fact that they suppose that the context of words may provide important information for finding a morphological analysis, and that they are interested in finding a simple, knowledge-free way to have access to that information.

Schone and Jurafsky [57] use a knowledge-free technique not altogether unlike the one found in Harris [31], and refine its results using latent semantic analysis (LSA). The procedure works as follows: from a text, take a list of word types, and build a trie from them. A potential stem exists over each string that extends from the base node of the trie to a juncture, and a potential suffix exists over each string that extends from a juncture to a terminal node. Consider only the stem/suffix combinations that include one of the more common suffixes found in the trie. Then, for each stem/suffix combination that has an identical string as the stem, determine whether a morphological relationship truly exists, or whether a morphological relationship only appears to exist, by performing a latent semantic analysis over the words in the text. This is done by building an $n \times 2n$ matrix out of the n most common words: each row is assigned to one of those n words as targets, and each column is

used to represent the number of times a particular word from this set of n appears in a large window either before or after the corresponding row word in the text. It is then possible to perform a singular value decomposition of this matrix. One can compare any two rows of the resulting matrix by taking their cosine: this allows one to determine how semantically alike any two words are by comparing, in this way, the frequency of the words that typically appear in the neighborhood of the target words. Words that share an identical stem and that are judged to be semantically similar by this measure are considered to be morphologically related, while words that share an identical stem and that are judged not to be semantically similar are removed from consideration. From this, one can arrive at a list of stems and a list of suffixes that are used to associate morphologically related words.

This paper represents a novel improvement over and above the work of Harris [31]. Harris recognizes that finding continuations is sure to include some false positives in the set of words that would be identified as being morphologically related, and he has ways to eliminate some of those false positives by, for example, finding continuations backwards, as well as forwards, through the data. Schone and Jurafsky address this problem in a slightly different way, however: rather than relying solely on facts about the ways in which phones are combined to form words to detect such false positive, Shone and Jurafsky find a way in which one can use facts about the distribution of words in a text to get some kind of information about the semantic content of each word. With this piece filled in, it is possible to use a very simple technique for building tries to find stems and suffixes without admitting as many false positives. Although these results are interesting from a technical point of view, it does not seem that their system has any more linguistic insight than the one that Harris proposes. Using latent semantic analysis provides a convenient way to get information about which words are semantically related without resorting to tagged data, but many of Harris's initial assumptions are left unaddressed.

Yarowsky and Wicentowski [66] perform knowledge-free induction of morphology that is similar, in certain ways, to the work of Schone and Jurafsky. They do, however, use a wider array of measures to gain insight as to whether or not two words are indeed morphologically related, beyond the simple measures of semantic and orthographic similarity that Schone and Jurafsky use. Yarowsky and Wicentowski are also interested in finding morphological relationships that involve stem changes as well as relationships that involve affixation, meaning that they are interested in a set of morphological relationships that the literature discussed so far has ignored.

In order to capture these morphological relationships, they measure orthographic similarity not in terms of sharing material near the root of a trie, but rather in terms of Levenshtein edit distance, with a phonologically informed set of edit costs: consonant-consonant substitutions as well as insertions and deletions are assigned a cost of 1, whereas vowel-vowel substitutions have a cost of 0.5. (For details on the Levenshtein edit distance, see Gusfield [30].) This difference in costs between consonant-consonant and vowel-vowel substitutions certainly works well for the Germanic languages that Yarowsky and Wicentowski consider, but it is not clear that it ought to be a good choice in the general case—after all, the non-concatenative inflectional morphology in Athabaskan and Celtic languages involves consonant rather than vowel mutations.

These authors also make use of the relative frequencies of words in the corpus to determine whether particular pairs of words have the kind of relative frequencies that are typical of forms that are known to be related by a particular morphological relationship. This allows them to correctly identify the pairs *sing*∼*sang* and *singe*∼*singed* without having to appeal to any kind of measure of semantic relatedness. Yarowsky and Wicentowski do also deploy semantic relatedness, however, according to cosine measures in a way very similar to that used in Schone and Jurafsky, and they find that it can be used to give better results when used in concert with relative frequency.

Furthermore, the output of the system is not just a list of pairs with a measure of

how likely they are to be morphologically related: once a number of pairs of related words have been identified, the system finds the probability with which a stem will undergo a particular change when a particular set of morphological feature values appears on it, given the last three characters of the stem. The discussion here is restricted to certain cases of ablaut and suffixation in Germanic languages, where word forms can be related by suffixation, a vowel change, or both. The fact that Yarowsky and Wicentowski even consider stem changes in addition to concatenation makes this work distinct from other structuralist and knowledge-free work.

Note that the present system differs from Yarowsky and Wicentowski's in certain key respects, though: for one, Yarowsky and Wicentowski find morphological rules that relate surface forms that belong to the same lexeme, rather than morphological rules that build surface forms from an underlying representation through the serial application of rules. Also, Yarowsky and Wicentowski have different assumptions about the kinds of operations that a morphological rule can perform on a base form. The present system assumes that a morphological rule can perform a single prefixation, suffixation, or infixation operation; a single rule cannot make two changes to a word form at two non-contiguous points in that word form. This means that a mapping from an underlying form to a surface form that includes vowel gradation and suffixation—as is common in many modern Indo-European languages—must be treated by the system being presented here as a rule of infixation that inserts the vowel, as well as a rule of suffixation; Yarowsky and Wicentowski simply treat each surface form-to-surface form mapping as a single process. Yarowsky and Wicentowski are essentially interested in the kinds of mappings that relate surface forms of a lexeme, whereas the present system seeks to correctly identify the morphosyntactic feature values that each distinct morphological rule realizes.

The work of Baroni, Matiasek, and Trost [8] is also similar to that of Schone and Jurafsky; the main difference is that they do not use tries to establish pairs of

forms that share a good deal of orthographic material at the left or right edge of forms. Instead, they use a variation on the Levenshtein edit distance to determine how orthographically alike two surface forms are, and on the basis of this edit distance as well as measures of semantic similarity, they find pairs that can be considered to be morphologically related. This is some of the most interesting work on knowledge-free induction of morphology, particularly in the sense that the authors get good results across a range of concatenative and non-concatenative morphologies, and with very simple measures of orthographic and semantic similarity. Note in particular that the measure of edit distance that these authors use is one in which all edits are simply assigned a cost of 1. The measure for semantic similarity is one that makes use of vectors of mutual information between the each target word and each context word. The mutual information of a target word and a context word is the log of the probability of observing both words within some window, given the fact that one or the other of the two words has been observed. It is found using this formula:

$$\mathrm{mi}(\mathrm{target}, \mathrm{context}) = \log \frac{\mathrm{p}(\mathrm{target}, \mathrm{context})}{\mathrm{p}(\mathrm{target}) \cdot \mathrm{p}(\mathrm{context})}.$$

For any two target words that one wants to compare, one can make a vector out of the measure of mutual information that each target word shares with each context word. The semantic "alikeness" of any two words can be found by taking the cosine of their vectors. Baroni, Matiasek, and Trost use the Levenshtein edit distance and this measure of semantic relatedness to return pairs of words that are likely to be morphologically related. It would then be possible to make generalizations about how these pairs are related to one another in a given language, but Baroni, Matiasek, and Trost do not take this step themselves—the system simply returns a list of morphologically related pairs.

The work of Chan [12], [13] is an approach to knowledge-free induction of mor-

phology that makes use of the relative frequency of surface forms in a corpus in a unique way. The key postulate in Chan's work is that the most frequent morphological configuration of a word in the training corpus can be taken as the base form of that word, and that the surface forms of all other morphological configurations of that word can be derived from that particular surface form, or "base". In order to maintain this postulate, Chan's system allows for a large and diverse set of morphological rules to exist, because it is often necessary for a rule to delete a certain amount of material, and then replace it with some other material. This work is entirely concerned with the kinds of rules that can be put in place to relate one surface form to another, rather than to relate a single base form with several surface forms. It is also a knowledge-free system, in the sense that it does not make use of any knowledge about the morphosyntactic features that a word carries when it formulates rules. Thus, there are several fundamental differences between Chan's system and the present system: Chan's system attempts to identify morphological rules on the basis of surface-to-surface correspondences, whereas the present system attempts to identify morphological rules on the basis of mappings between an underlying form and surface form. The present system also identifies morphosyntactic features that are associated with each rule, whereas Chan's system simply imagines each cell in an inflectional paradigm as a unique form with no particular relationship to any other form except the base form. Chan's system is also unique in that it is particularly well suited to incremental learning, because a hypothesis for the analysis for each word in the training corpus can be formed on the basis of some small initial corpus, and then revised every time the learner encounters a surface form.

For all the differences in the assumptions that underlie Chan's system and the present system, however, they both face a similar challenge in terms of the evaluation of the output grammar. Chan's system assigns morphological rules to words, and words that are assigned the same morphological rules can be grouped together; in a

similar way, the present system groups words into inflection classes based on those words taking on the same morphological rules. In both cases, one of the most natural ways to evaluate the output grammars is in terms of the clusters of words that they recognize, rather than in terms of the stem-suffix boundaries that are easy to identify and evaluate in grammars that describe morphological grammars in terms of a set of stems and a set of affixes that may be attached to those stems. Chan's system and the present system both search for morphological rules, rather than some other unit of inflectional morphology that might be imagined, and they group words that take on similar rules—but they are still based on different assumptions about the nature of the system being described, and the resulting analyses reflect this fact.

1.3.3 Morphological induction using Bayesian techniques

Ockham's razor and common sense both dictate that a simple model is preferred to a more complicated one, supposing that both models cover the observed data equally well. According to MacKay [44], this intuition can be seen more formally in terms of Bayes's theorem: the best model, given some data, is the model that has the highest value for the product of the a priori probability of the generative model and the likelihood of generating the training data under that model. A model with fewer parameters will, all else being equal, have a higher prior probability assigned to it, since it contains fewer degrees of freedom than a model with more parameters. In this way, Bayes's theorem provides a straight-forward way to balance the need to find a model that can generate the observed data in a constrained way against the danger of finding a model that has been over-fit to the training data.

To be sure, there are other ways to capture the notion of simplicity in a model of a cogntive system. As mentioned in section 1.3.1, Goldsmith [25] uses the notion of minimum description length as a measure of the simplicity or aptness of a morphological system; Chater and Vitányi [14] discuss the use of Bayesian techniques and

Kolmogorov complexity as measures of simplity in cognitive science more generally.

Note, however, that measures of simplicity in grammars can be invoked in two distinct ways: as a measure of which grammar, out of several grammars all describing the same data, and written using the same conventions, is the most apt, and as a measure of which theory of grammar, out of several, is the most apt way to describe the languages of the world. The position in Anderson [5], [6] is that evaluation metrics in generative grammar are useful for determining what kinds of abbreviatory conventions should be recognized in a theory of phonology, since abbreviatory conventions that recognize a key fact about the language faculty will prove useful time and time again when writing grammars for a wide variety of languages, while spurious conventions will not. It is stressed that feature-counting metrics are useful as a way to make decisions about what kinds of conventions should be recognized in a theory of grammar, but that they are of no particular value for determining which of several possible analyses is "correct".

In the case of the present system, and of all the learners described in this section, though, the evaluation metric is used as a tool that can be used to find an apt grammar for some set of data, and which is hoped to capture many of the same generalizations that are captured in humans' knowledge of their native languages. The Bayesian objective function measures the suitability of a grammar in terms of its ability to generate the observed data with high probability, and in terms of the prior probability of the grammar in the space of possible grammars. In other words, in the present work, and in the work described in this section, Bayes's theorem is being used as a tool for evaluating the relative merits of candidate grammars within a particular system. The output of the Bayesian objective function is not being used at the higher level, described in Anderson [5], [6], to determine the kinds of abbreviatory conventions that the theory of grammar should allow. (It should be noted, however, that hierarchical Bayesian models do allow one to compare models in

a way similiar to the manner in which Anderson describes, although the work in this area does not specifically address morphology or phonology. For more on hierarchical Bayesian models, see MacKay [44].)

For a concrete example of Bayesian techniques applied to morphological induction, see the work of Snover [59] and Snover, Jarosz, and Brent [60]. In these papers, a list of surface forms is taken as the input. The goal is to divide these forms into linguistically appropriate stems and final suffixes while using a particular method for assigning probabilities to any particular way of dividing the list into stems and suffixes, as well as a particular method for searching the space of possible stem-suffix divisions. In this regard, the form of the grammars resembles the form of the grammars in Goldsmith's Linguistica. The idea is to find the best hypothesis given the data, which can be expressed as

$$\mathrm{p}(\text{hypothesis}|\text{data}) = \frac{\mathrm{p}(\text{hypothesis}) \cdot \mathrm{p}(\text{data}|\text{hypothesis})}{\mathrm{p}(\text{data})}.$$

This can be reduced to

$$\mathrm{p}(\text{hypothesis}|\text{data}) = k \cdot \mathrm{p}(\text{hypothesis}),$$

where k is a constant across all hypotheses, or

$$\mathrm{p}(\text{hypothesis}|\text{data}) \propto \mathrm{p}(\text{hypothesis}),$$

since, under this system, p(data) and p(data|hypothesis) are constant across the whole set of hypotheses that may be entertained.

Probabilities are assigned to hypotheses according to the following principles: a hypothesis having fewer stems and suffixes is favored slightly over a hypothesis having more stems and suffixes, with the distribution of probabilities following the

series of inverse squares. This series of inverse squares provides a convenient way to get a distribution over the set of natural numbers so that small numbers are favored over larger numbers, but which is relatively flat at at very high values. The system being proposed here makes extensive use of this distribution. Another term reflects the probability of the phonological form of each of these stems and suffixes, given the relative frequency of phones in the language. Then, for any hypothesis, the probability of there being some particular number of paradigms between 1 and the number of stems is given by a flat distribution over that interval. The probability that some number of suffixes is associated with each paradigm is also given by the flat distribution, as is the probability that particular suffixes are associated with a particular paradigm. Finally, the probability that a particular suffix is associated with a particular paradigm is given by the relative number of stems that is assigned to each paradigm. This system shows how one can break a grammar down into certain parameters, and assign a flat or nearly flat probability distribution to each one.

The space of possible stem-suffix divisions is too big to search exhaustively for an input list of any reasonable size. However, it should be the case that good morphological generalizations can be found even in subsets of the full list, and that these generalizations can then be carried over to the full word list. The search method described in this paper takes advantage of this fact: sub-hypotheses are formed over small sets of words, with the additional restriction that each stem posited by the sub-hypothesis must be combinable with every suffix that it identifies, and each suffix must be combinable with every stem, to form a word that appears somewhere in the full word list. This is essentially the same thing as assuming that each sub-hypothesis has to be built over a set of words that all belong to the same paradigm. The likelihood of one of these sub-hypotheses can be compared with the likelihood of the null hypothesis in which all words under consideration are taken to be a stem followed by

the null suffix. The value

$$\frac{\text{p(novel hypothesis)}}{\text{p(null hypothesis)}}$$

expresses how good the novel hypothesis is compared with the null hypothesis. The papers present a method by which these sub-hypotheses can be arranged into a directed graph, in which a hypothesis that includes a superset of the stems and affixes posited by some less complete hypothesis is placed in a downstream position from the less complete hypothesis. It is then possible to begin a search at the sub-hypothesis that posits zero stems and zero suffixes in order to find the best sub-hypotheses. The paper then gives a method by which these sub-hypotheses—each of which essentially represents a part of a paradigm found in the word list—can be built into a single, more comprehensive hypothesis.

The grammars that Snover and Snover, Jarosz, and Brent return are rather like the grammars found by Goldsmith's system. As mentioned above, this kind of approach to morphology is not the best approximation of what happens in natural languages in the general case, since it is clear that many languages make use of non-concatenative morphology, and that it is often necessary to make morphological generalizations at level of representation that is not directly accessible from the speech signal. However, the way in which these papers define grammars of word formation in terms of certain parameters can easily be extended to a more articulated model of word formation. See chapter 2 for a description of the parameters and the distributions over parameter values that the present system uses.

1.3.4 The role of phonology in morphological learning

Most of the work on learning morphology has taken word formation to be a matter of concatenating morphemes, which do not then need any kind of phonological readjustment. This means that instances of allomorphy that may have a phonological explanation—such as the sets [s]∼[z]∼[əz] and [t]∼[d]∼[əd] in English—must simply

be treated as three distinct endings that happen to appear on arbitrarily defined sets of bases. The learners simply cannot recognize phonological generalizations, because they assume that phonological processes do not intervene between constructing the word out of morphemes and the surface forms of words that can be observed. The following papers all deal with the search for phonological generalizations in one way or another, but they do not represent the same kind of cohesive research program as the work on learning morphology discussed earlier.

Gildea and Jurafsky [22] attempt to induce a finite state transducer that is equivalent to the phonological rule of flapping in American English:

$$\mathrm{t} \rightarrow \mathrm{ɾ} \;/\; \acute{\mathrm{V}}\ (\mathrm{r})\ \_\_\ \mathrm{V}\ .$$

They are initially unsuccessful with the OSTIA algorithm, but they find that this is because the OSTIA algorithm, as a general-purpose algorithm for inducing finite state transducers, lacks certain biases that make for shorter, more general, and arguably more natural, SPE-style phonological rules. Specifically, they cite the notion that output characters should, in general, resemble the corresponding input characters in some phonologically meaningful way; they refer to this notion as "faithfulness". They also cite the notion that similar segments should undergo similar changes in the mappings that apply to them; they call this "community". These ideas seem obvious to linguists who are used to stating phonological generalizations in terms of the natural classes to which certain generalizations apply, and in terms of phonological feature values that undergo the change. The moral of the story is that while it may be possible to state an SPE-style rule in terms of a finite state transducer, it does not follow that a system for learning finite state transducers over segments, and without any knowledge of natural classes in phonology, will necessarily be able to make reasonable phonological generalizations, even when it is trained on natural

language data. The natural language data may support SPE-style generalizations in a particular learner, particularly if that learner is sensitive to natural classes of segments and the phonological features that different segments carry, but the learner will need at least some amount of phonological knowledge to induce generalized phonological rules correctly.

Goldwater and Johnson [28] take the output of Linguistica as the input to their system, and they attempt to find phonological (or orthographic, really) generalizations that can be used to compress sets of signatures into a single signature. Looking at the sample Linguistica outputs that Goldsmith gives makes it clear that many signatures end up serving as classes that recognize different inflectional markers not because they truly represent different paradigms or inflectional categories, but rather because some signatures account for phonologically controlled allomorphy, whereas others really do account for distinct paradigms or inflection classes. For example, Linguisitca would put the forms *cry*, *cries*, *cried* and *crying* in one signature and the forms *shoo*, *shoos*, *shooed* and *shooing* in another, even though another approach—and one favored by most linguists—would be to recognize these two words as inflecting according to the same pattern, but with certain orthographic regularities that involve verb-final vowels disrupting the perfect division of these forms into stems and suffixes drawn from the same set. Of course, similar techniques might be used to find a generalization about the alternation between [əz], [z], and [s] in English, but the concern in this paper is with allomorphy in orthography, not allomorphy in phonological representations. Goldwater and Johnson want to find generalizations that can be used to reduce these instances of allomorphy to a single signature and a rule of orthographic readjustment that applies throughout the language.

In this system, the search for new hypotheses proceeds in a fairly straight-forward way, by examining the set of signatures for sets that may be collapsed if an orthographic rule is included in the new hypothesis. The particularly interesting thing in

this work is Goldwater and Johnson's findings about the prior probabilities that must be assigned to hypotheses that may include phonological rules. They find that simply using Goldsmith's MDL measurement for the aptness of a grammar actually prefers grammars in which allomorphy is treated by dividing stems in various inflectional categories, rather than grammars in which the number of signatures is reduced in favor of treating similar signatures as orthographically related versions of the same signature or inflectional class. Goldwater and Johnson identify two reasons for this preference: first, they point out that shifting characters (such as *y*, in *cry*∼*cries*, *fly*∼*flies*, and *try*∼*tries*) from a suffix to a stem is generally dispreferred, since if a character is shifted into the stem for many words, it will be counted many times more in the various stems than it would if it only appears in a few suffixes. The other reason that Goldwater and Johnson identify is the fact that the cost (measured in bits) associated with signatures must be balanced against the cost associated with phonological rules. Unless the system used to assign prior probabilities to hypothesized grammars is constructed appropriately, the evaluation metric may well choose grammars in which facts about orthographic regularities are simply listed as allomorphic variants (in this case, as signatures) rather than as phonological generalizations. Once Goldwater and Johnson account for these facts, however, their system successfully discovers orthographic alternations, and these alternations allow it to collapse certain signatures in a way that the basic Linguistica system cannot.

This work is followed up by the work of Naradowsky and Goldwater [50], which also attempts to formulate spelling rules for English verbs, but which additionally employs a single Bayesian utitility function to evaluate systems that include both a morphological analysis and a set of re-write rules. In this regard, the system described in Naradowsky and Goldwater and in Goldwater, Griffiths, and Johnson [27] is very similar to the present system, although there are certain key differences. For one, Naradowsky and Goldwater's system only considers insertions and deletions as possi-

ble orthographic re-write rules, whereas the present system makes use of phonological rules of insertion, deletion, and substitution, all over phonological forms, rather than orthographic forms. Naradowsky and Goldwater's system is also a knowledge-free learner, in which the various proposed morphological analyses are tied only to a division in the word between stem and suffix. This means that the evaluation metric used in Naradowsky and Goldwater is similar to the evaluation metric used in the present system, insofar as they both use Bayes's theorem to favor simpler grammars of word formation. However, the search system and the grammars themselves have several key differences. The biggest difference is that the the system in Naradowsky and Goldwater is a knowledge free system, so there is no need to associate particular suffixes with particular sets of morphosyntactic feature values. As a result, Naradowsky and Goldwater can perform Markov Chain Monte Carlo sampling over the set of all possible parses, because the set of possible parses is significantly simpler and smaller in their system.

The minimum generalization learner (MGL) in Albright [3] and Albright and Hayes [4] also comes very close to doing the kind of learning as the system being proposed here. Their work assumes that the training data has been tagged for inflectional information, and that the problem for the learner is to figure out what kinds of phonological changes are associated with various inflectional information on various bases, in the course of the presentation of the training data. The learner may posit rules that select one fully-inflected surface form from a paradigm as the base, and that derive the other members of the paradigm from it, or the learner may abstract to a base form which never appears as a surface form, but which instead is the underlying representation of the root.

The learner works by constructing, as pairs of training data are presented, rules that perform the mapping from the base form to the derived form. These rules are stated in terms of a structural description and a structural change, and the structural

description is, at least initially, assumed to be fully specified. As training goes on, however, and more and more rules accumulate, the learner posits certain more general rules that perform the same mappings but with more general structural descriptions. All morphological rules are evaluated according to the same metric—a measure of "reliability", which in this case is given a formal definition:

$$\text{reliability} = \frac{\text{forms derived correctly by the rule}}{\text{forms matching the rule's SD}}.$$

The same measure is used for phonological rules, which are arrived at by essentially the same procedure. The result is a grammar in which base forms—either underlying or surface—are mapped to surface forms by means of morphological and phonological rules, with the decision of which rules to apply being made on the basis of each rule's probability, given the context: the probability is proportional to reliability for each rule for which the structural description indicates that it potentially could apply. The minimum generalization learner finds a grammar in which all morphological and phonological rules are associated with some probability, with that probability assigned by how often that particular structural change applies, given the structural description. The grammar treats inflectional categories and irregular forms as properties that can be inferred from the phonological form of the base, with the correct treatment of each being handled by information that is present in the structural description of the rules. This aspect can be contrasted with systems—such as the system being proposed in this thesis—in which membership in an inflectional category is taken to be an arbitrary property of a base form. Notice that this allows the minimum generalization learner to be fed unattested forms, so that it can respond with the word inflected for some other set of morphosyntactic features. This allows the MGL to perform well in something like a "wug" test, and in a way that is more or less like human speakers, who often seem to use something like phonological neighborhood

when they determine how to inflect a novel form.

1.4 The place of the present system

Much of the previous work on the machine learning of morphology has assumed that it is appropriate to imagine that the surface forms of words may be decomposed into constituent morphemes, and that learning morphology is a matter of identifying how these morphemes are represented, and perhaps finding something out about the way in which they combine. However, as mentioned in section 1.2, the theoretical work in morphology and phonology indicates that morphology is more accurately modeled not in terms of the concatenation of morphemes, but in terms of the successive application of morphological and phonological rules to a base. Much of this earlier work on morphological induction also assumes that it is appropriate to make morphological generalizations over phonemic or orthographic representations, whereas it is probably more appropriate to recognize an underlying phonological form from which surface forms are built. The primary goal of the present system is to demonstrate an automatic learner that induces grammars of phonology and inflectional morphology of a type that are appropriate, given the discussion of natural languages in section 1.2.

It is in the interest of representing the inflectional morphology of natural human languages accurately that the present system allows for the inflection of a word form to proceed through an arbitrary number of morphological levels in order to take on its surface form, and that each inflection rule, with the exception of thematic rules, is associated with a particular set of morphosyntactic feature values. The present system also allows for a fairly wide set of morphological rules to operate on surface forms: it allows prefixation, suffixation, and infixation, although ideally an even larger set of morphological rules would be admitted. Furthermore, the present system searches for phonological alternations as it searches for a suitable grammar of inflectional morphology. This seems appropriate, because many lexical and word-internal phonological

alternations are only evident when one examines morphological variants, while these phonological alternations make certain morphological generalizations more evident as well. Recall the discussion of the plural marker in English: the fact that it is a single rule that marks the plural in *doves*, *parrots*, and *thrushes* is only evident when the appropriate phonological rules are known.

While much of the previous work on inducing morphology has focused on morpheme segmentation, and while much of this work has been performed in the context of knowledge-free induction, this is not universally true. Along the same lines, the previous work on grammar induction has also, by and large, separated the search for morphological generalizations from the search for phonological generalizations, but exceptions do exist. For example, Chan [12], [13] and Yarowsky and Wicentowski [66] describe learners that are capable of discovering a wide variety of morphological rules, but they perform this learning in a knowledge-free context. The work of Yarowsky and Wicentowski does not tie the morphological rules that they discover to particular morphosyntactic features, and the work of Chan is aimed specifically at discovering surface form-to-surface form mappings. The present system learns morphological rules that are drawn from a slightly more restricted set of possible edits than the systems presented in these learners—but it is certainly unique insofar as it supports morphological derivations of several cycles. The use of morphological tags in the training data allows the present system to associate particular edits with particular morphosyntactic features, and it also is undoubtedly closer to the situation faced by a human learner than learning in the knowledge-free context: it is unreasonable to think that a human learner does not make use of other linguistic knowledge when positing the morphological and phonological grammar in place for a given language.

The work of Goldwater and Johnson [28] and Naradowsky and Goldwater [50] is also significant because, like the present system, it discovers certain edit rules that allow morphological generalizations to be carried further. In the case of this work by

Goldwater and Johnson, as well as that by Naradowsky and Goldwater, the object is to discover rules of spelling alternation rather than of phonological alternation, but they are very similar insofar as they are searching for edits that are no longer than a single character, and that are conditioned by the surrounding characters that may have been introduced by the application of morphological rules. The present system goes beyond this work in several ways, however. For one thing, the grammars of inflectional morphology are more complex that the stem/suffix analyses discovered in these systems. The present system also deploys a richer concept of phonological rules. Naradowsky and Goldwater recognize only rules of insertion and deletion, and this is in fact appropriate for capturing a large number of the spelling alternations that take place in English—such as the deletion of *e* before the *-ing* suffix in the *quake* → *quaking* alternation. The present system recognizes phonological rules that insert and delete single characters, as well as rules that substitute one character for another. These rules are represented as edit rules that apply not to particular characters, but rather as edits that apply to natural classes of phonological segments, and that are similarly conditioned by natural classes of phonological segments in their context rather than particular characters.

One should note, in passing, that the present system is not built with parsing novel forms in mind, although it is certainly possible to use it for that purpose, if certain restrictions are kept in mind. The output grammars produced by the present system provide a set of inflection classes, and it is possible to determine whether a novel, morphologically tagged surface form might or might not belong to any particular inflection class. In many cases, however, it will not be possible to determine which single inflection class the novel form must necessarily belong to—it may well be that one or even several surface forms of a novel word do not provide enough information in order to determine which single inflection class is the only inflection class that can parse these novel forms correctly. In this regard, the work of Albright [3] and

Albright and Hayes [4] shows a clear advantage: in many languages, it is possible to use information about the phonological form of a novel word in order to determine the inflection class to which it belongs. In some situations, this assignment may be complete and accurate, and in other cases it may be possible to make only a fuzzy assignment. The key fact, however, is that the present system sees inflection classes strictly in terms of certain rules that are used for morphological generation and morphological parsing, whereas the work of Albright and Albright and Hayes is meant specifically to recognize the role of phonological forms in determining the assignment of a word to its inflection class.

It is the task of this dissertation to build and present a system that uses Bayesian techniques to find an appropriate set of morphological and phonological generalizations on the basis of a set of tagged training data. The present system induces grammars of inflectional morphology of a kind that has been discussed in the theoretical literature, and represented computationally, but never before induced automatically; it also induces certain phonological alternations at the same time. While these phonological alternations are less complex than those that have been suggested to exist in the literature of theoretical linguistics, they still offer significantly more phonological detail than those that have been induced as part of a morphological learner in the past.

Chapter 2

The Bayesian model for evaluating grammars

This chapter describes the way in which Bayes's theorem can be used to find the probability of a hypothesized grammar of word formation, given a set of observed training data. Using Bayes's theorem this way requires that one have a method that can be used to assign prior probabilities to grammars of word formation; it also requires that one have a method for finding the probability of generating the observed training data with such grammars. This chapter lays out the objective function that the present system uses to evaluate grammars of word formation in terms of their prior probabilities, as well as to evaluate the probability they assign to the observed training data. Once this objective function is in place, it can be used to evaluate the relative suitability of grammars of word formation, given a set of training data that they are know to have generated. In practice, the search for an appropriate grammar begins with a hypothesis about the grammar of word formation that is found, using a very simple procedure, from the tagged training data that is provided to the system. The system then searches for incremental improvements that can be made on this initial hypothesis, according to the objective function described here—but for more

on the specifics of the search procedure, see chapter 3.

Section 2.2 describes, in greater detail than chapter 1, the way in which grammars of word formation are stated, and the way in which they can be used to generate the phonological forms of words associated with particular lemma and morphosyntactic tags. Section 2.3 describes how the probability of some set of observed data, given a particular hypothesis about the grammar, can be found. Section 2.4 lays out the model that generates and assigns probabilities to these grammars of word formation, for the purposes of finding the a priori probability of any given grammar stated in these terms, while section 2.5 shows how this regime for assigning prior probabilities indeed prefers linguistically informed grammars. This chapter begins, however, with an introduction to Bayes's theorem, in which it is shown how the the terms p(data|hypothesis) and p(hypothesis) can be used to be determine which grammar, out of several possible grammars, is best suited to a set of observed data.

2.1 Introduction to Bayes's theorem

Bayes's theorem can be used to relate the conditional and marginal probabilities of two random, but not necessarily independent, events. The probability of event A, given the occurrence of event B, can be given by a formula that relates p(A|B) to the values p(A ∩ B) and p(B):

$$\mathrm{p(A|B)} = \frac{\mathrm{p(A \cap B)}}{\mathrm{p(B)}}.$$

In other words, the probability of event A, given event B, can be found by finding the probability that events A and B both occur, and dividing that value by the probability that event B occur in the first place. A similar formula gives the probability of event B, given event A:

$$\mathrm{p(B|A)} = \frac{\mathrm{p(A \cap B)}}{\mathrm{p(A)}}.$$

One can use these two formulae to relate the conditional and marginal probabilities directly, without mentioning $p(A \cap B)$ explicitly, by first isolating the term $p(A \cap B)$ in both:

$$p(A|B) \cdot p(B) = p(A \cap B);$$
$$p(B|A) \cdot p(A) = p(A \cap B).$$

At this point, $p(A|B) \cdot p(B)$ and $p(B|A) \cdot p(A)$ can be related to one another by the transitivity of equality:

$$p(A|B) \cdot p(B) = p(B|A) \cdot p(A).$$

This equation gives the crucial relationship between p(A), p(B), p(A|B), and p(B|A), although it is often rewritten so that p(A|B) is isolated:

$$p(A|B) = \frac{p(B|A) \cdot p(A)}{p(B)}.$$

This relationship can be used to evaluate the suitability of a particular generative hypothesis as the explanation for a set of observed data. Under this approach, one treats the observed data as one event and the generative hypothesis as another event. The generative hypothesis is associated with some prior probability, which represents the likelihood of it being selected out of the blue, from the set of all imaginable hypotheses, without reference to any observed data. Once such a distribution over the prior probabilities of the set of hypotheses has been set, however, one can use Bayes's theorem to find a value that represents the probability of the hypothesized model being the model that did, indeed, generate the observed data:

$$p(\text{hypothesis}|\text{data}) = \frac{p(\text{data}|\text{hypothesis}) \cdot p(\text{hypothesis})}{p(\text{data})}.$$

Taking this line of reasoning one step further, one can evaluate a number of possible generative hypotheses, all drawn from the same distribution of prior probabilities, and then select as the winner whichever hypothesis results in the greatest value for p(hypothesis|data). This is the approach adopted in the present work, but an example in which the space of hypotheses is more limited may make this method more clear.

Suppose that the prevalence of some disease in the general population is six individuals per every ten thousand. This means that the a priori probability of a person being infected is six ten-thousandths, 0.0006. A blood test for the disease gives a false positive one out of every forty times that it is administered, and it never gives a false negative. The events "infected" and "not infected" can be taken as the set of hypotheses that generate the data, which ranges over the two discrete values "positive" and "negative". Some probability is associated with each mapping from a generative hypothesis to a value in the observed data. Suppose that the test result for an individual is "positive". One can use Bayes's theorem to find the exact probability that the "infected" hypothesis, rather than the "not infected" hypothesis, indeed generated the observed "positive" data:

$$\begin{aligned}
\text{p(hypothesis|data)} &= \frac{\text{p(data|hypothesis)} \cdot \text{p(hypothesis)}}{\text{p(data)}} \\
\text{p(infected|positive)} &= \frac{\text{p(positive|infected)} \cdot \text{p(infected)}}{\text{p(positive)}} \\
&= \frac{1 \cdot 0.0006}{0.0006 + (0.9994 \cdot 0.025)} \\
&\approx 0.02345124.
\end{aligned}$$

Note that the value for p(positive) is found by summing the probabilies for each route by which the positive result can be generated in an out-of-the-blue context: either the individual is infected (with probability 0.0006), and consequently tests positive $(1 \cdot 0.0006)$, or the individual is not infected (with probability 0.9994), but still has

a one-in-forty chance of testing as a false positive ($0.9994 \cdot 0.025$). One can work out the probability for the alternative hypothesis in the same way:

$$\begin{aligned}
\text{p(hypothesis|data)} &= \frac{\text{p(data|hypothesis)} \cdot \text{p(hypothesis)}}{\text{p(data)}} \\
\text{p(not infected|positive)} &= \frac{\text{p(positive|not infected)} \cdot \text{p(not infected)}}{\text{p(positive)}} \\
&= \frac{0.025 \cdot 0.9994}{0.0006 + (0.9994 \cdot 0.025)} \\
&\approx 0.97654875.
\end{aligned}$$

The test carries what seems to be only a small risk of a false positive, but the facts about the incidence of the disease in the general population mean that the probability that the test is being administered to a healthy individual is much greater than the probability that the test is being administered to an infected individual. This effect is so great that, in practice, the test returns many more false positives than true positives. One can use Bayes's theorem to find the probability that an individual who has tested positive indeed carries the disease, 0.02345124, and this can be compared with the probability that an individual who has tested positive is in fact healthy: 0.97654875. Even if an individual has tested positive for the disease it is still far more probable that the individual is not infected, in the absence of other evidence for infection.

In the present work, Bayes's theorem is used in a similar way to evaluate the suitability of various grammars of word formation as the model responsible for generating a set of training data. Note that the value for p(data) can be treated as a constant for the purposes of comparing two grammars relative to one another: the term p(data) simply refers to the probability of the observed training data, and the training data is constant for all the grammars that might be hypothesized to explain it. Thus the term p(data) is a constant for the purposes of finding the relative suitability of any

given grammar for that set of training data, and it can be set aside as a constant factor:

$$\begin{aligned}
\text{p(hypothesis|data)} &= \frac{\text{p(data|hypothesis)} \cdot \text{p(hypothesis)}}{\text{p(data)}} \\
\text{p(hypothesis|data)} &= k \cdot \text{p(data|hypothesis)} \cdot \text{p(hypothesis)} \\
\text{p(hypothesis|data)} &\propto \text{p(data|hypothesis)} \cdot \text{p(hypothesis)}.
\end{aligned}$$

In order to find the hypothesis that is most probable, given the data, one still finds the hypothesis that maximizes the product of p(data|hypothesis) and p(hypothesis). In other words, one tries to find the hypothesis for which the product of the probability of generating the observed data under the hypothesized grammar and the prior probability assigned to the hypothesized grammar is greatest.

At this point, a few words about the way in which such a term can be used to give preference to more linguistically apt grammars are in order, although a detailed description of the way in which the term favors certain grammars in the present system is reserved for section 2.4. For the moment, though, note that one can employ a model that uses a flat, or nearly flat, distribution to determine the value of certain key parameters of the grammar, such as the number of roots, the number of inflectional rules, the number of inflection classes into which roots and rules are organized, and the number of phonological rules. Each root and each rule must then be specified in terms of a fixed number of other parameters. These parameters specify, for example, the phone that appears in each position of a root, or the specific structural description or structural change of a phonological rule. Under this approach, the distribution over the number of roots does very little to favor grammars with fewer roots directly. Nevertheless, the generative model still tends to assign higher prior probabilities to grammars with fewer roots: after all, a grammar that has more roots will usually have to include more phones in order to specify the phonological forms of those roots.

Since the probability of any particular phone appearing in any particular position in a root can only fall between 0 and 1, and since each additional phone that is specified in a grammar must have its probability included as a term in the expression used to find p(hypothesis), each additional phone that is included in a grammar can only cause its probability to move closer and closer to zero. The key point here is that even though the generative model may choose many of its parameter values according to flat or nearly flat distributions, the model can still assign higher prior probabilities to simpler grammars on the basis of this fact that fewer terms tend to give a higher value for p(hypothesis). The fact that models with fewer parameters tend to have higher probabilities simply because they contain fewer degrees of freedom is related to the discussion of Bayes's theorem and Ockham's razor found in MacKay [44], but it is distinct: while MacKay's discussion centers on the posterior probability of the observed data, given some hypothesis, the point being made here is simply focused on the nearly inescapable tendency for the prior probability of a model with many terms to be lower than the prior probability of a model containing only a few.

Note also that there are obvious parallels between, on the one hand, this kind of preference for models containing fewer parameters in Bayesian approaches to machine learning, and evaluation metrics within traditional generative grammar on the other. (For a more detailed discussion of these evaluation metrics, see Chomsky and Halle [15], Anderson [5], [6], and their treatment in section 1.3.3.) Given the fact that linguists almost always value terseness in grammars, it seems reasonable to expect an objective function based on Bayes's theorem to be able to distinguish which of two grammars is more linguistically apt in a way that is similar to the evaluation metrics generative grammar. This depends, of course, on an appropriate regime for assigning prior probabilities to grammars—but, as discussed above, the fact that grammars that can be described briefly tend to be more probable than grammars that can only be described with more parameters as a natural way to quantify linguists' intuitions.

The next section describes in more detail the set of grammars of word formation that are considered, and the section that follows describes how prior probabilities are assigned to these grammars.

2.2 The form of hypothesized grammars

Each grammar consists of a set of roots, R, a set of morphological rules, M, and a set of phonological rules, P. Every root and every morphological rule must be associated with an inflection class; this set of inflection classes is denoted by IC. The surface form of a word is generated by choosing a root from the set R, and then choosing an appropriate set of morphosyntactic feature values, f, for it on the basis of its grammatical category, gc. Then the appropriate inflectional rules from M apply, on the basis of those feature values and the inflection class to which it belongs. Finally, the set of phonological rules, P, applies to give the surface form of the word. Each of these phonological rules has the opportunity to apply once, as the set applies simultaneously. One might imagine word formation as the application of functions, composed in a certain order, representing morphological and phonological rules, on some base that serves as the argument: $h \circ g \circ f(\text{base})$. When grammars are construed in this way, learning the morphology and phonology of a natural language can be thought of the process of finding the set of bases that serve as the domain for certain functions, as well as finding the ordered set of functions that apply to those bases.

The set of roots in a grammar is denoted by R; the expression |R| is used to indicate the number of roots in a particular grammar. Each root is associated with an underlying phonological form, pf. Each root is also associated with some particular inflection class, ic, which groups the root with other roots of the same grammatical category, gc, and which inflect according to the same set of morphological rules.

The set of morphological rules in a grammar, M, contains |M| inflectional rules.

Each morphological rule is associated with an inflection class that specifies the roots—all of the same grammatical category—on which it can potentially apply, and the other morphological rules with which it can potentially compete for application. In this way, inflection classes serve as conjugation or declension classes which associate the appropriate rules and roots with one another.

Every morphological rule specifies the morphosyntactic feature values that must be present on the words to which it applies. Every morphological rules is also associated with a mapping from the null string to some more elaborate string, indicating an insertion operation on the base upon which the rule applies. This insertion operation can take place as a prefix, suffix, or infix, depending on the conditions specified by the rule's structural description. One can think of this mapping from null to some more elaborate string as a structural change, and the context of this mapping as either a prefix, suffix, or infix as its structural description. This context, or structural description, for the rule may specify that the string is inserted at the extreme left or right edge of the form to which the rule applies, or it may indicate that the rule is a rule of infixation, with the site of infixation being some specified number of segments from either the left or right edge of the word. Note that this is a significant simplifying assumption: many theories of morphology assume that the application of morphological rules can be conditioned by the phonological context provided by the base form. These effects are well-attested in natural languages, but the present system assumes that the structural description of a morphological rule can only restrict its application in terms of the location of its application within a word, and that this conditioning is always in terms of the suffix, prefix, or infixes position relative to either the left or right edge. Note, though, that this assumption is adopted for the purposes of making the space of possible grammars easier to search, and not as a theoretically desirable restriction on the space of grammars.

There is, as well, another advantage to be had in stating the structural change in

a morphological rule as a mapping from the null string to a more elaborate string: it allows the probability of morphological rules in a grammar to be evaluated in a way that is easily comparable with the probability of the strings that are used to represent the underling forms of roots. See section 2.4 for more on the way in which the probability of strings in roots and morphological rules is found.

The fact that the structural change of morphological rules is stated in terms of a mapping from the null string to a more elaborate string has a consequence for some languages: under this formalism, it is not possible for a single rule to induce a change in the phonological form of a word at two or more discontinuous points. This means, for example, that certain English strong verbs that undergo both ablaut and suffixation in order to mark the past tense (such as *feel*∼*felt*, or [fil]∼[fɛlt], phonetically) must be analyzed as having the feature [+past] marked with two separate rules: one rule infixes [ɛ], and another that suffixes [t]. In the case that the word is marked with [−past], on the other hand, a rule must infix [i] at the same position so that the word can contain a vowel. This analysis of suffixation and ablaut in English helps give some idea of what kinds of phenomena the present system can capture in a way that a linguist would find reasonable, and at what point the formalisms allowed in the present system will fail to reasonably capture phenomena in natural languages.

Finally, each inflectional rule is also associated with a depth, d, which is some small, non-negative integer specifying the depth of the derivation at which that inflectional rule may apply. In principle, the depth of an inflectional rule is essentially the same as the rule block to which a rule belongs in Anderson [7] and Stump [61]: when a form is first passed to the system of inflectional morphology, only rules of depth d = 0 are allowed to apply, and in fact only the most specific rule of depth d = 0 may apply, if indeed any of the conditions have been met for any of the rules of depth d = 0. Once one of the rules of depth d = 0 have been allowed to apply, the derivation is passed on to the rules of depth d = 1. At this point, only rules of

depth d = 1 may apply, and again only the most specific rule applies, in the case that there are several whose conditions are met. Note that the depths associated with inflectional rules ensure that the rules apply in the right order, so that morphs are ordered appropriately in the word. They also allow more specific rules to block the application of more general rules, but this blocking only applies to rules that occupy the same depth, so that appropriate multiple exponence is not also blocked. The most specific rule at each depth is found strictly by counting the number of morphosyntactic features that each rule realizes—more specific rules realize more morphosyntactic features, but no further measures of specificity is employed.

Note that under these assumptions about the form of a grammar, a grammar can account for fully irregular forms by putting the irregular form in its own inflection class, taking the phonological form of the root to be null, and taking the full set of surface forms to be affixes onto this null root. Essentially, the kind of grammars used here assume that all unpredictable information about the surface form of a word is captured either in the underlying representation of its root, or in the set of inflectional rules that are associated with its inflection class.

At this point, the stage is set for a simple example of the application of several inflectional rules. Consider [evleri], the accusative plural form of the word "house" in Turkish. This word is formed from a root, [ev], which has had first a plural-marking suffix applied to it, [ler], followed by another suffix which marks the accusative, [i]. Assume for the moment that there are no other roots in this inflection class, and that there are also no other inflectional or phonological rules at play. The rules can be stated as follows:

$$\text{IC-1} : \{\langle \varnothing \rightarrow \text{ler}, \text{suffix}, \text{d} = 0, \langle \text{plural} \rangle \rangle,$$
$$\langle \varnothing \rightarrow \text{i}, \text{suffix}, \text{d} = 1, \langle \text{accusative} \rangle \rangle \}.$$

At the beginning of the derivation, the root [ev] is selected, and it is assigned a tag associating it with the accusative plural:

ev ⟨accusative, plural⟩

The derivation of the surface form now proceeds by walking through the various depths that are included in the set of inflectional rules, and applying those rules which are associated with a sub-set of the morphosyntactic feature values found on the form. The first depth to be considered is depth 0, at which point the plural marker is suffixed to the form:

evler ⟨accusative, plural⟩

Note that the string introduced by the plural suffix is inserted at the right edge of the base form, because the rule specifies that it applies as a suffix. The ⟨plural⟩ tag on the form is what triggers the application of the rule. Next, the derivation moves on to depth 1, and the rule marking the accusative applies in a similar way:

evleri ⟨accusative, plural⟩

At this point, there are no remaining rule depths for this particular inflection class. The form is now passed on to the system of phonological rules, which may operate in such a way as to change its form. Note that the notion of rule depths or rule blocks in this not just responsible for the rule blocking effects between less specific and more specific inflectional rules: it is also responsible for ensuring that morphological rules are applied in the right order during the derivation. These suffixes must be applied to the base form in the correct order if the correct surface form is to be derived.

The set of phonological rules, P, consists of |P| unordered phonological rules. Each

rule consists of a structural description (SD) and a particular structural change (SC), which can be written together as XAY $\rightarrow$ XBY or A $\rightarrow$ B/X__Y. In the present system, it is assumed that a structural change always operates on a string that is no more than one segment long, either modifying a single segment, inserting a single segment, or deleting a single segment. The structural description can be stated in terms of a single segment to the immediate left of the SC, in terms of the two segments to the immediate left of the SC, or in terms of a single segment to the immediate right of the SC, or in terms of two segments to the immediate right of the SC, or in terms of either one or two segments to the left and right of the SC. The structural description also contains information indicating whether the immediate left and immediate right context should be understood in terms of all segments to the left or right of the SC, or whether it should be restricted to either the vowel tier or the consonant tier. In this way, the system is able to account for a reasonably large set of potential phonological rules, although the space of possible phonological rules is nowhere near as complex as the space of possible finite-state transducers. Note also that this discussion is in terms of segments, as positions in the string, but that both structural changes and structural descriptions are in fact stated in terms of natural classes of phones rather than in terms of particular segments or phones.

The structural change of a phonological rule is given as one or zero segments, defined in terms of a partially- or fully-specified vector of phonological feature values, and one or zero output segments that record the feature values that change. Note that the phonological rules in this system resemble very closely the rules of generative phonology, although they carry the additional restriction of only performing edits on single segments. This simplifying assumption restricts the set of phonological rules that can be modeled—it means, among other things, that rules of metathesis and coalescence cannot be represented.

Phonological rules are assumed to refer only to phones and natural classes of

phones, but not to syllables or other prosodic structure, and not to any kind of morphological information. The entire set of phonological rules applies once, simultaneously, to every word form. It is also assumed that each phonological rule applies unfailingly, with p(application) = 1, when its structural description is met; furthermore, the assumption is that all phonological generalizations are surface true, and never rendered opaque by any other processes in the grammar. The treatment of phonological generalizations as a set of re-write rules that operate over matrices of phonological features is familiar from the literature on generative phonology—see, for example, Chomsky and Halle [15], Anderson [5], and Kenstowicz and Kisseberth [36]—although the present system makes many simplifying assumptions when compared with the literature on generative phonology.

See Anderson [5] for a discussion of simultaneous rule application. Although Anderson rejects simultaneous rule application as an appropriate way to describe the systems of phonological rules in natural languages, that work still contains several examples in which simultaneous rule application is in fact adequate for a particular set of data. For the purposes of the present system, simultaneous rule application is implemented as follows: each rule that is present in the grammar is given its own copy of the form that serves as the input to the phonological component. Each rule then attempts to operate on each segment of the form that was given to it as input. It is then possible to merge the outputs of these rules: for each position in the word, the outputs of all the rules are compared. Most of the time, all the rules will agree that no change should take place on the segment being considered. In some cases, though, a single rule will indicate that the segment in question should not have an output form identical to its input form, and in such a case the modified segment will be used. (The same holds for inserted and deleted segments.) In principle, it is in fact possible that several rules would all attempt to modify the same segment—for example, one might imagine a rule of assimilation and a rule of deletion that both try

to operate on the same segment, because the context is appropriate for both rules. In such instances, the precedence of phonological rules is determined by a count of the number of phonological features that are used to define their SD and SC. The basic idea is that more specific rules ought to displace the application of more general rules, so the rule that is defined with a larger set of phonological rules applies in place of the rule with a smaller set.

As a brief example of phonological rule application, consider a rule in Latin that devoices [g] to [k] in certain contexts:

$$[g] \rightarrow [k] \; / \; \_\_ \; [s\#] \text{ (on the general tier)}$$

Note that [g], [k], and [s] are taken as abbreviations for the phonological feature matrices spelled out in full in section 4.4.1. The symbol # is taken as the end-of-word marker; it is associated with a phonological feature matrix which contains only negative values, including a negative value for the feature termed "segmental". This sets up one natural class that contains all of the phones of the language, and another that contains only the # symbol. Note, however, that the # symbol is never deployed within representations—it is used only as a placeholder at the edge of a representation, indicating that the edge of the representation has been reached.

Consider the root [reg], meaning "king", when it appears in the nominative singular. It begins the derivation as a bare root, marked with the appropriate morphosyntactic feature values:

[reg] ⟨nominative, singular⟩

Now the nominative singular suffix is added to the representation:

[regs] ⟨nominative, singular⟩

This exhausts the inflectional rules that can apply to this form, so it is now dispatched to the phonological rule system. It is only at this point that # symbols are added to the representation, so that phonological rules that apply near word boundaries can determine where, exactly, they should apply—see Anderson [7] for a discussion of boundary symbols in various stages of morphological and phonological derivations:

[##regs##] ⟨nominative, singular⟩

This representation is now fed to all the phonological rules that exist in this language. The rule for [g]-devoicing finds that its context is met at one particular point in the representation, and so it produces this output:

[reks] ⟨nominative, singular⟩

This representation is now compared with the outputs of all the other phonological rules that exist in the language. Suppose, for the sake of argument, that Latin also contains a distinct rule of [b]-devoicing:

[b] → [p] / __ [s#] (on the general tier)

This rule also takes [##regs##] as its input representation, and it produces [regs] as its output representation. At this point, the two output representations are aligned with one another. The representations clearly agree on all segments, except for the [g]∼[k] alternation. For this particular segment, [k] is selected, because it can attributed to the application of a specific phonological rule. This determination can be made on the basis of the fact that the rule of [g]-devoicing can be seen to have made an edit, versus its input form, on the last segment of its output, whereas the rule of [b]-devoicing can be seen to have made no change in that location of this

input-to-output mapping.

Note that generation of the morphosyntactic feature values f associated with words is not part of the grammars that the present system must learn. The assumption is that a natural language allows, for any word of a particular grammatical category, certain morphosyntactic feature values to occupy certain positions in a vector associated with that word. For the most part, these morphosyntactic representations are similar to those discussed in Anderson [7], but it is also assumed that morphosyntactic representations can be represented with a flat structure, and that recourse to a nested or other more complex structure is not necessary. According to this model of morphosyntactic representations, the Turkish form [evlernin] might be associated with the vector ⟨plural number, null possessor, ablative case⟩. For the purposes of the present work, the probability that a particular feature value will be put in a particular position in this vector is taken to be independent of everything except for the grammatical category of the word. This probability can be taken directly from the training data, however, and does not need to be learned. In fact, the generation of f does not even need to be included as part of the grammar of word formation—the values in f are responsible only for triggering the application of inflectional rules, and their generation is taken to be external to the grammar.

The fact that the set of feature values that is associated with a word depends only on the grammatical category of the root has an important consequence for the treatment of defective and deponent forms. The present system treats tokens of a defective form as exemplars of a fully inflectable root since, to the system, they will be indistinguishable from regular words which happen to be attested in forms that carry only certain feature values. The present system is also not equipped to treat deponent forms appropriately. In any case, the present system works under the assumption is that the assignment of morphosyntactic feature values to forms is made strictly on the basis of grammatical category, and that this assignment is distinct from the grammars

that must be learned.

The next section explains how the probability of the data under a particular generative model can be found; the section after that explains how the probability of a particular hypothesized grammar can be found.

2.3 Likelihood of the training data

The treatment of the term p(data|hypothesis) is straightforward under the present assumptions, in which the grammatical category of each root is given in the training data, the morphosyntactic feature values that appear on a form depend on the grammatical category of the root, and morphological and phonological rules apply deterministically. Finding p(data|hypothesis) is more involved if the system attaches probabilities to the application of morphological and phonological rules, or if the base forms to which inflectional rules apply are built by a set of probabilistic derivational rules, but under the present assumptions, p(data|hypothesis) can simply be treated as a constant for all hypotheses that are indeed capable of generating at least the observed data. Consider this situation in which one has both a set of tagged training data and a grammar on hand: for each item in the training data, one can simply see what surface form the grammar returns for that particular root and configuration of morphosyntactic feature values. Since the grammar will always return the same surface form when given the same root and configuration of morphosyntactic feature values, the grammar is either capable of generating all the points in the training data, or it is not. This means that, in practice, one can maximize p(data|hypothesis) $\cdot$ p(hypothesis) by simply maximizing p(hypothesis), as long as one makes sure to consider only hypotheses that can indeed generate the points in the training data. The term p(data|hypothesis) is simply a constant for all hypotheses that are sufficient to generate the observed data.

The fact that p(data|hypothesis) can be treated as a constant for grammars that

can indeed generate the observed data relies upon on the assumption that the generation of the morphosyntactic features associated with each root during the process of word formation depends only on the roots grammatical category, and that this generation is separate from the grammar under evaluation. This approach can be compared with certain other approaches to inflectional morphology that do not allow p(data|hypothesis) to be taken as a constant.

First, consider the systems described in Goldsmith [25], [26], Snover [59], and Snover, Jarosz, and Brent [60]. In these systems, morphosyntactic features are not represented; instead, allowing for certain differences between the terminology used in that work and in the present work, roots and rules are simply associated with one another in inflection classes for the purposes of determining which rules apply to which roots in order to generate surface forms. Under this approach to the form of grammars, p(data|hypothesis) can be increased by avoiding hypotheses in which stems and suffixes are paired in ways that generate unattested forms—even if those unattested forms are just cells in a particular word's paradigm that happen not to be attested in the training data. The fact that the present system represents morphosyntactic features, but that the generation of these features is outside the province of the hypothesized grammars, means that p(data|hypothesis) cannot be increased inappropriately in this way.

Along similar lines, compare the present system with a system in which inflectional rules are incremental rather than realizational—where in an incremental system, the application of a phonological rule changes the morphosyntactic representation of a form by adding feature values to it, whereas in a realizational system, inflectional rules apply as a result of feature values that are already present in the representation. The term p(data|hypothesis) can be treated as a constant under an incremental approach to inflectional morphology as long as the following condition is met: probabilities must associated with the application of inflectional rules in such a way that the forms

given as outputs by a set of grammars associated with a set of training data always conform to a certain distribution of probabilities over the various configurations of morphosyntactic features.

The assumptions that inflectional rules are realizational and that determining the values associated with any given word is external to the grammar are both helpful for allowing p(data|hypothesis) to be treated as a constant term that can be factored out of the objective function. Instead of performing a lengthy calculation of p(data|hypothesis) for the grammar under consideration, one instead must verify that the grammar is indeed capable of generating the training data. As long as this condition is met, p(data|hypothesis) can be factored out a constant, and grammars can be compared with one another just on the basis of the term p(hypothesis).

2.4 The distribution of prior probabilities

Under the approach outlined here, a grammar consists of a set of roots and a set of inflectional rules organized into inflection classes, and a set of phonological rules that apply uniformly to all words, after the appropriate inflectional rules apply. Since the prior probabilities of the phonological component and the set of inflection classes are independent, the prior probability of a particular hypothesized grammar can be found as the product of the prior probability of the set of inflection classes and the prior probability of the set of phonological rules:

$$\text{p(hypothesis)} = \text{p(IC)} \cdot \text{p(P)}.$$

As a concrete example, take a fragment of the Turkish nominal system. Imagine the following training data, in which the first column gives the surface form of each word form, and the second column gives its grammatical category. The third column gives the lemma to which each word form belongs, and the last column gives its

morphosyntactic feature values:

adam,	NOUN,	man,	⟨nominative singular⟩
adam~~ı~~,	NOUN,	man,	⟨accusative singular⟩
adamlar,	NOUN,	man,	⟨nominative plural⟩
adamlar~~ı~~,	NOUN,	man,	⟨accusative plural⟩
en,	NOUN,	width,	⟨nominative singular⟩
eni,	NOUN,	width,	⟨accusative singular⟩
enler,	NOUN,	width,	⟨nominative plural⟩
enleri,	NOUN,	width,	⟨accusative plural⟩
ev,	NOUN,	house,	⟨nominative singular⟩
evi,	NOUN,	house,	⟨accusative singular⟩
evler,	NOUN,	house,	⟨nominative plural⟩
evleri,	NOUN,	house,	⟨accusative plural⟩
kar,	NOUN,	snow,	⟨nominative singular⟩
kar~~ı~~,	NOUN,	snow,	⟨accusative singular⟩
karlar,	NOUN,	snow,	⟨nominative plural⟩
karlar~~ı~~,	NOUN,	snow,	⟨accusative plural⟩

Now consider an analysis of this data in which the full set of morphological rules and the full set of roots belong to a single inflection class, the class of regular nouns:

$$\text{IC-1} : \textsc{nouns}, \{\langle\text{adam}\rangle, \langle\text{en}\rangle, \langle\text{ev}\rangle, \langle\text{kar}\rangle\}.$$

In this example, one can reasonably suppose that all roots belong to the same inflection class, but if one were considering a reasonable subset of Latin nouns, a wider set of inflectional classes might be employed to account for the various declensions and sub-regularities within them.

The set of morphological rules for this fragment of Turkish are also associated with this same class of regular nouns:

$$\begin{aligned}\text{IC-1} : \{&\langle \varnothing \rightarrow \text{ler}, \text{suffix}, \text{d} = 0, \langle \text{plural} \rangle \rangle, \\ &\langle \varnothing \rightarrow \text{i}, \text{suffix}, \text{d} = 1, \langle \text{accusative} \rangle \rangle \}.\end{aligned}$$

Finally, the set of phonological rules consists of just one rule, a rule of progressive vowel harmony according to the feature [+back]:

$$\begin{bmatrix} +\text{vocalic} \\ -\text{consonantal} \end{bmatrix} \rightarrow [+\text{back}] / \begin{bmatrix} +\text{vocalic} \\ -\text{consonantal} \\ +\text{back} \end{bmatrix} \_\_ \text{ (on the vowel tier).}$$

The exact value for the particular p(hypothesis) associated with this grammar will be worked out as each term of p(hypothesis) is described in the rest of this section.

Returning to the question of how to find the value for p(hypothesis) = p(IC)·p(P), one can expand each of the terms thus:

$$\begin{aligned}\text{p(IC)} &= \text{p}(|\text{IC}|) \cdot \prod_{i=1}^{|\text{IC}|} \text{p}(\text{ic}_i); \\ \text{p(P)} &= \text{p}(|\text{P}|) \cdot \prod_{i=1}^{|\text{P}|} \text{p}(\text{rule}_i).\end{aligned}$$

In order to account for p(IC), one must first account for the probability that the grammar has a particular number of inflection classes; one must then account for the probability associated with each one of those inflection classes. Along similar lines, in order to account for p(P), one first finds the probability that the grammar indeed has a particular number of phonological rules; one then finds the probability associated with each one of those rules.

This value for p(|IC|) is found according to the probability distribution associated

with the series of inverse squares:

$$\mathrm{p}(|\mathrm{IC}|) = \frac{6}{\pi^2} \cdot \frac{1}{|\mathrm{IC}|^2}.$$

This term $\frac{1}{|\mathrm{IC}|^2}$ favors small values of |IC| to high values |IC|, but this becomes less and less pronounced as higher and higher values are compared against one another. After all, for $|\mathrm{IC}| = 1$, $\frac{1}{1^2} = \frac{1}{1}$, which can be compared with the situation for for $|\mathrm{IC}| = 2$, where $\frac{1}{2^2} = \frac{1}{4}$. The difference between 1 and $\frac{1}{4}$ is dramatic. At higher values, however, the penalty for each additional |IC| becomes less pronounced: while $|\mathrm{IC}| = 10$ returns $\frac{1}{10^2} = \frac{1}{100}$, $|\mathrm{IC}| = 11$ returns only $\frac{1}{11^2} = \frac{1}{121}$.

Note that it is in this regard that the inverse squares series differentiates itself from several other distributions which also assign probabilities to the whole numbers from 1 or 0 to infinity, such as the Poisson distribution, the series of inverse cubes, or the series based on the inverse exponential function, $\frac{1}{2^n}$: the series of inverse squares devotes more of its probability mass to high numbers rather than low numbers, as compared with these other distributions. Although all of these distributions tend to assign higher probabilities to lower inputs than higher inputs, the inverse squares distribution has a much thicker tail than even the Poisson distribution in the limit. In short, the inverse square series always prefers smaller values to larger values; this preference is most dramatic for very small values, whereas it is less pronounced for larger values; the distribution also falls off much slower than several other distributions which also cover the same set of inputs. At the same time, of course, each inflection class is associated with a particular probability which depends on the roots and rules that it contains. These probabilities also work to favor grammars containing fewer, rather than more, inflection classes.

In this way, the terms used to find p(|IC|) serves to give a slight preference for grammars with fewer inflection classes. It is in the other term, $\prod_{i=1}^{|\mathrm{IC}|} \mathrm{p}(\mathrm{ic}_i)$, that the

probabilities associated with the class's roots and rules reside, and it is that term that does the most to prefer short, simple grammars over more complex grammars. At higher values of |IC|, there are more and more inflectional rules, and more and more terms that must be specified. It is these terms that give preference to terse grammars over more detailed grammars with more inflection classes. For more on the inverse-squares distribution, see Aigner and Ziegler [2].

In order to find the value for this other term in p(IC), one simply needs to take the product of the probabilities associated with each of the inflection classes, ic_i, found in the grammar. The probability of any $\text{p}(\text{ic}_i)$ is given by the following formula:

$$\text{p}(\text{ic}_i) = \text{p}(\text{gc}) \cdot \text{p}(\text{z}) \cdot \text{p}(|\text{roots}|) \cdot \left(\prod_{j=1}^{|\text{roots}|} \text{p}(\text{root}_j) \right) \cdot \text{p}(|\text{rules}|) \cdot \left(\prod_{k=1}^{|\text{rules}|} \text{p}(\text{rule}_k) \right).$$

This formula accounts for the probability of the grammatical class, gc, associated with the inflection class. It also accounts for the maximum depth, or number of disjunctive rule blocks, denoted by z, that the inflection class permits. The remaining terms treat the probability of the number of roots that it contains, for the probability of each of those roots, for the probability of the number of inflectional rules that it contains, and for the probability of each of those rules.

In the formula above, it is the term p(gc) that represents the probability that the inflection class is associated with a particular grammatical category, such as NOUN, VERB, or some other category recognized by the language being examined. This probability is drawn from the flat distribution over the set of grammatical categories, GC, recognized in the language, so it can be stated in the following way:

$$\text{p}(\text{gc}) = \frac{1}{|\text{GC}|}.$$

Note that this term is actually a constant for all grammars that might possibly be

associated with some particular set of training data, since that set of training data will always contain the same set of grammatical categories. This term is necessary to make the formula for p(hypothesis) probabilistically sound, but its value does not vary for a set of grammars associated with a particular set of data.

The value of this term can be found quite simply in the Turkish example, where GC = {NOUN}:

$$\begin{aligned} \mathrm{p(gc)} &= \frac{1}{|\mathrm{GC}|} \\ &= \frac{1}{1} \\ &= 1. \end{aligned}$$

Turning now to the value for p(z), the value for this term is found according to the now-familiar series of inverse squares:

$$\mathrm{p(z)} = \frac{6}{\pi^2} \cdot \frac{1}{(\mathrm{z}+1)^2}.$$

The purpose of this term is to provide the set of values that can be assigned to the depth, d, of a morphological rule belonging to the ic. The value of d for a morphological rule belonging to some inflectional class is drawn from the interval [0, z], where z is specific to that particular inflection class. In other words, the term z simply provides the maximum depth permitted by the inflection class.

In the current example, there are morphological rules that apply at two different

depths, d = 0 and d = 1. The value for p(z) is found accordingly:

$$\begin{aligned} \mathrm{p(z)} &= \frac{6}{\pi^2} \cdot \frac{1}{(\mathrm{z}+1)^2} \\ &= \frac{6}{\pi^2} \cdot \frac{1}{(1+1)^2} \\ &= \frac{6}{\pi^2} \cdot \frac{1}{2^2} \\ &= 0.15198178. \end{aligned}$$

The value for p(|roots|), which represents the number of roots associated with a given inflection class, is found with the distribution over inverse squares:

$$\mathrm{p}(|\mathrm{roots}|) = \frac{6}{\pi^2} \cdot \frac{1}{|\mathrm{roots}|^2}.$$

Again, this distribution tends to favor smaller values of |roots| to larger values, but it behaves similar to a flat distribution when comparing the probabilities associated with large values for |roots| against one another.

In the current example, this term can be worked out quite straightforwardly:

$$\begin{aligned} \mathrm{p}(|\mathrm{roots}|) &= \frac{6}{\pi^2} \cdot \frac{1}{|\mathrm{roots}|^2} \\ &= \frac{6}{\pi^2} \cdot \frac{1}{4^2} \\ &= \frac{6}{\pi^2} \cdot \frac{1}{16} \\ &\approx 0.037995508. \end{aligned}$$

The value for any $\mathrm{p}(\mathrm{root}_j)$ can be found with the expression:

$$\mathrm{p}(\mathrm{root}_j) = \frac{6}{\pi^2} \cdot \frac{1}{(|\mathrm{pf}|+1)^2} \cdot \prod_{l=1}^{|\mathrm{pf}|} \mathrm{p}(\mathrm{phone}_l).$$

In this term, pf is the phonological form of the root, consisting of a string of phones, and |pf| is the length of the phonological representation, in phones.

Both of these terms serve to make short roots more probable than long roots. The length of each root is associated with the same inverse squares probability distribution used earlier—but notice that the denominator is $(|\text{pf}|+1)^2$, which allows for null roots, with length zero. Also note that $\text{p}(\text{phone}_l)$ is simply calculated as the raw frequency of the phone in the training data.

As a brief example of the $\text{p}(\text{root}_j)$ term at work, consider the value for p(root) for the root [kar] in the current example:

$$\begin{aligned}
\text{p}(\text{root}_{[\text{kar}]}) &= \frac{6}{\pi^2} \cdot \frac{1}{(|\text{pf}|+1)^2} \cdot \prod_{j=1}^{|\text{pf}|} \text{p}(\text{phone}_j) \\
&= \frac{6}{\pi^2} \cdot \frac{1}{(3+1)^2} \cdot \text{p}([\text{k}]) \cdot \text{p}([\text{a}]) \cdot \text{p}([\text{r}]) \\
&= \frac{6}{\pi^2} \cdot \frac{1}{16} \cdot \frac{4}{76} \cdot \frac{16}{76} \cdot \frac{4}{76} \\
&\approx 8.8632 \times 10^{-5}.
\end{aligned}$$

The probability associated with each phone in this example is taken as the phone's relative frequency in the sample subset of Turkish given earlier. In order to find the probabilitiy of each phone, one simply tallies up the total number of segments that appear in the training data, and then finds the fraction of those segments that are instances of each phone.

Turning now to the morphological rules associated with an inflection class, the value for p(|rules|) can be found straightforwardly, again with reference to the distribution of associated with the series of inverse squares:

$$\text{p}(|\text{rules}|) = \frac{6}{\pi^2} \cdot \frac{1}{(|\text{rules}|+1)^2}.$$

As is quite familiar at this point, this term uses the inverse squares series to prefer

grammars that have fewer morphological rules to grammars that have more, but without making this preference insurmountable for high values of |rules|. It also allows for inflection classes that contain no inflectional rules at all with the term (|rules| + 1) in the denominator.

The probability of any particular morphological rule i can be given by

$$\text{p}(\text{rule}_i) = \text{p}(\text{SD}) \cdot \text{p}(\text{SC}) \cdot \text{p}(\text{other parameters}).$$

The terms p(SD) and p(SC) express the probability of the rule's structural description and structural change, while the term p(other parameters) expresses the probability of the depth at which it applies and the set of feature values it realizes.

First consider the term p(SD). The structural description of a morphological rule can specify that the rule is a prefix, a suffix, or an infix. In the event that the rule is a prefix or suffix, no other parameters must be specified, but in the event that it is an infix, the SD must specify whether the infix is placed relative to the left or right edge of the word, and the number of segments from the edge that it is placed. The generative model assumes the following assignment of probabilities to SDs:

$$\text{p}(\text{SD}) = \begin{cases} \frac{1}{3} & \text{(prefix);} \\ \frac{1}{3} & \text{(suffix);} \\ \frac{1}{3} \cdot \frac{1}{2} \cdot \frac{1}{f} \cdot \frac{6}{\pi^2} & \text{(infix).} \end{cases}$$

This distribution assigns a probability of $\frac{1}{3}$ to prefixes and $\frac{1}{3}$ to suffixes. The remaining $\frac{1}{3}$ is distributed over all the possible specifications for infixes, in such a way that an infix specified from the left edge of a word is just as probable as an infix specified from the right edge of a word, with a probability of $\frac{1}{2}$ being assigned to both possibilities. Furthermore, infixes with a small offset (indicated with f) are more probable than infixes with a large offset. The probability associated with each of these suffixation

rules in Turkish is thus $\frac{1}{3}$.

Turning now to the formula used to find p(SC) for a morphological rule, the system uses the following expression:

$$\text{p(SC)} = \text{p(output)},$$

where the output is simply the string that is introduced, either as a prefix, suffix, or infix. The expansion of p(output) is

$$\text{p(output)} = \frac{6}{\pi^2} \cdot \frac{1}{(|\text{output}| + 1)^2} \cdot \prod_{n=1}^{|\text{output}|} \text{p(phone}_n\text{)}.$$

The term $|\text{output}_j|$ is the number of phones in the output to that mapping. The value for $\text{p(phone}_n\text{)}$ is found according to the same principles as it is for $\text{p(phone}_j\text{)}$ in the expansion of p(R): that is, for each phone, it is simply the raw probability of that phone in the training data.

Notice in particular that probabilities are associated with the number of phones in the SC of a morphological rule in a way that is very similar to the way in which probabilities are associated with the number of phones in a root. The same holds true for the way in which probabilities are associated with the identity of the phones appearing in each position. This means that in the search for the most probable grammar for some set of data, one might try to move from a less probable grammar to a more probable grammar by adjusting which phones are taken to be part of the root, and which are taken to be added by morphological rules. Depending on how these adjustments are made, it may be possible to find sets of roots that can be appropriately grouped into a single inflection class. This parity between the phonological material that appears in roots and in morphological rules is a key feature of the term p(grammar), and the search procedure described in chapter 3 relies upon it.

The term referring to "other parameters" deals with the probabilities associated

with the formal properties of the inflectional rule, rather than with its context or effect. It is expanded thus:

$$\mathrm{p}(\text{other parameters}) = \mathrm{p}(\mathrm{d}|\mathrm{ic}) \cdot \mathrm{p}(\mathrm{f}|\mathrm{gc}),$$

where f is the bundle of feature values present on the forms to which the rule applies, and d is the depth at which it applies. The value for d is assumed to be drawn from a flat distribution over the interval [0, z], where z is used to specify the maximum depth allowed by the inflection class:

$$\mathrm{p}(\mathrm{d}) = \frac{1}{(\mathrm{z}+1)}.$$

Recall that z is a property of the inflection class, and that it is used to specify the number of rule blocks that exist in the inflection class. The depth, d, of any given inflection rule within an inflection class can be any value in the interval [0, z]. Because $\mathrm{z} = 1$ for this Turkish example, the value of $\mathrm{p}(\mathrm{d}) = \frac{1}{2}$.

Turning now to the probabilities associated with the particular morphosyntactic features associated with inflectional rules, one can assume that the probability for any particular set of morphosyntactic feature values f drawn from a flat distribution over all the possible combinations of morphosyntactic feature values that can appear on the grammatical category associated with the inflection class. Suppose that H represents the set of morphosyntactic features that can appear on a surface form of a particular grammatical category, and that values(H(i)) represents the number of distinct values that can be assigned to that feature as it appears on a word. It is then the case that values(H(i)) + 1 represents the number of distinct values that the morphosyntactic feature might take on when it appears in an inflectional rule, as well as accounting (with the +1 element) for the possibility that no feature value is specified for this particular feature in this particular rule. The probability for any

particular set of morphosyntactic feature values appearing in f is then given by

$$\mathrm{p(f|gc)} = \prod_{i=1}^{|\mathrm{H}|} \frac{1}{(\mathrm{values(H}(i)) + 1)}.$$

This formula allows for morphological rules that have a value specified for just one morphosyntactic feature, or for several, or for none at all, in the case that all morphosyntactic features happen to have no particular value specified, as is the case with thematic rules. A morphological rule can have any available value specified for any of the morphosyntactic features in the language; alternatively, it may leave any number of morphosyntactic features unspecified. The assumption is that a morphological rule may be specified for no feature values, or for one, or for several, and that those specified values are what must be present on the form in order for the rule to apply. Note that the value of p(f|gc) is in fact a constant for all hypotheses consistent with a particular set of training data: the value of values(H(i)) depends not on the hypothesis, but rather on the training data, so the value of p(f|gc) cannot vary from hypothesis to hypothesis.

Note that this convention does not penalize a language for making use of fusional rather than agglutinating morphology—it simply charges a flat price per morphological rule, although this price does vary from language to language on the basis of the number of morphosyntactic features that a language employs.

Consider briefly the probability associated with the rule marking the plural in Turkish, which may be spelled out as

$$\langle \varnothing \rightarrow \mathrm{ler}, \mathrm{suffix}, \mathrm{d} = 0, \langle \mathrm{plural} \rangle \rangle.$$

The value for the term p(other parameters) associated with this rule can be found quite easily. Remember that d is found according to a flat distribution over the integers in the interval [0, z], where the value of z is specific to the ic; remember also

that p(f|gc) is a constant for any gc in a particular language. Working this term out explicitly, one finds the following:

$$\begin{aligned} \text{p(other parameters)} &= \text{p(d|ic)} \cdot \text{p(f|gc)} \\ &= \frac{1}{2} \cdot \frac{1}{9} \\ &\approx 0.055555556. \end{aligned}$$

Finding the probability for the structural description is also quite simple: because this is a rule of suffixation, the SD is associated with probability $\frac{1}{3}$.

Finding the probability of the structural change associated with the rule is more involved. In order to do this, one must find the probability of the string that the rule introduces, [ler]:

$$\begin{aligned} \text{p(output)} &= \frac{6}{\pi^2} \cdot \frac{1}{(3+1)^2} \cdot \prod_{n=1}^{3} \text{p(phone}_n\text{)} \\ &= \frac{6}{\pi^2} \cdot \frac{1}{(3+1)^2} \cdot \text{p([l])} \cdot \text{p([e])} \cdot \text{p([r])} \\ &= \frac{6}{\pi^2} \cdot \frac{1}{4^2} \cdot \frac{8}{76} \cdot \frac{12}{76} \cdot \frac{12}{76} \\ &\approx 9.97112035208 \times 10^{-5}. \end{aligned}$$

This means that the value for p(rule) can now be calculated:

$$\begin{aligned} \text{p(rule)} &= \text{p(SD)} \cdot \text{p(SC)} \cdot \text{p(other parameters)} \\ &\approx 1.8465037689 \times 10^{-6}. \end{aligned}$$

One work can out the values for p(root) and p(rule) for the other roots and rules that are part of this inflection class. Once this step is completed, it is possible to find

the value for p(IC) in this grammar:

$$\text{p(IC)} \approx 5.40801197024 \times 10^{-30}.$$

The next step is to find the probability associated with the set of phonological rules in the grammar. The value for p(P) can be determined by finding the value for the following two terms, and taking their product:

$$\text{p}(|\text{P}|) = \frac{6}{\pi^2} \cdot \frac{1}{(|\text{P}|+1)^2};$$
$$\prod_{i=1}^{|\text{P}|} \text{p}(\text{rule}_i).$$

The term

$$\text{p}(|\text{P}|) = \frac{6}{\pi^2} \cdot \frac{1}{(|\text{P}|+1)^2}$$

straightforwardly prefers grammars with fewer phonological rules to grammar with more phonological rules. Since p(|P|) uses the same distribution based on the inverse squares series as, for example, the term p(|pf|) in p(phonological form$_i$), the model prefers smaller numbers of phonological rules to larger numbers of phonological rules, although this is most pronounced for very small values of |P|.

Finding the probability of a particular phonological rule is a bit more involved. The initial formula is given by

$$\text{p}(\text{rule}_i) = \text{p(SD)} \cdot \text{p(SC)},$$

but these terms each expand to

$$\text{p(SC)} = \text{p(c)} \cdot \text{p(a)};$$
$$\text{p(SD)} = \text{p(t)} \cdot \text{p(s)}.$$

The term p(t) refers to the general facts about the triggering environment for the rule: is the rule environment one segment long, or two? Does it fall on the left side of the segment that undergoes the change, or on the right, or on both sides? Is the context to be read as belonging to the vocalic tier, the consonantal tier, or to the all-purpose, simple segmental tier? The term p(s) refers to the probability associated with the feature matrices that are used to define the segments appearing in the triggering environment. The term p(c) refers to the probability of the feature matrix defining the segment that serves as the input to the phonological rule, and the term p(a) refers to the probability of the feature matrix that defines the output of the rule.

The value for p(t) is simply taken from a flat distribution over all the available triggering environments that allowed in this system, so it consistently evaluates to $\frac{1}{18}$, as there are in fact 18 possible ways of defining the context of a phonological rule, independent of the feature matrices that are included.

The other terms that are used to describe the properties of phonological rules all assume that the probability of a possibly under-specified phonological feature matrix that is one segment long can be given by the following formula:

$$\text{p(phonological matrix)} = \frac{6}{\pi^2} \cdot \frac{1}{(|\text{f}|)^2} \cdot \left(\prod_{i=1}^{|\text{f}|} \frac{1}{|\text{F}|} \cdot \frac{1}{2} \right)$$

The term |f| is used to represent the number of features that are specified in the matrix, and |F| is used to represents the number of features that exist in the language. The first term, $\frac{6}{\pi^2} \cdot \frac{1}{(|\text{f}|)^2}$ accounts for the number of features that appear in the rule; the second term, $\prod_{i=1}^{|\text{f}|} \frac{1}{|\text{F}|} \cdot \frac{1}{2}$ is used to account for the probability of the feature values that actually appear in those slots. This distribution allows a phonological matrix to throw probability away onto phonologically unrealistic, or even phonologically impossible, matrices. It has the advantage, however, that it prefers simpler rules to more complex rules, without completely eliminating the less probable rules from consider-

ation. Notice too that this distribution does not prefer "natural" phonological rules to "unnatural" rules—for more on this issue, see Chomsky and Halle [15], Anderson [6]—but it does prefer shorter, more general rules to longer rules.

With these formulae in place, it is now possible to find the probability of the phonological rule in the Turkish example at the beginning of this section. Recall that the rule gives a simple statement of vowel harmony according to the feature [+back]:

$$\begin{bmatrix} +\text{vocalic} \\ -\text{consonantal} \end{bmatrix} \rightarrow [+\text{back}] / \begin{bmatrix} +\text{vocalic} \\ -\text{consonantal} \\ +\text{back} \end{bmatrix} \_\_ \text{ (on the vowel tier).}$$

The value for p(t) is always $\frac{1}{18}$. The value for the probability of the matrix that serves as the context can be found thus, based on the fact that it contains one segment, with three phonological features specified:

$$\begin{aligned} \text{p(s)} &= \frac{6}{\pi^2} \cdot \frac{1}{(|\text{f}|)^2} \cdot \left(\prod_{i=1}^{|\text{f}|} \frac{1}{|\text{F}|} \cdot \frac{1}{2} \right) \\ &= \frac{6}{\pi^2} \cdot \frac{1}{(3)^2} \cdot \frac{1}{36} \cdot \frac{1}{36} \cdot \frac{1}{36} \\ &= 1.4477788467 \times 10^{-6} \end{aligned}$$

The value for the probability of the input to the structural change can be found thus, again based on the number of segments specified, and the number of features specified per segment:

$$\begin{aligned} \text{p(c)} &= \frac{6}{\pi^2} \cdot \frac{1}{(|\text{f}|)^2} \cdot \left(\prod_{i=1}^{|\text{f}|} \frac{1}{|\text{F}|} \cdot \frac{1}{2} \right) \\ &= \frac{6}{\pi^2} \cdot \frac{1}{(3)^2} \cdot \frac{1}{36} \cdot \frac{1}{36} \\ &= 0.000117270086582 \end{aligned}$$

The value for the probability of the output to the structural change is based on the same principle:

$$\begin{aligned} \mathrm{p(a)} &= \frac{6}{\pi^2} \cdot \frac{1}{(|\mathrm{f}|)^2} \cdot \left(\prod_{i=1}^{|\mathrm{f}|} \frac{1}{|\mathrm{F}|} \cdot \frac{1}{2} \right) \\ &= \frac{6}{\pi^2} \cdot \frac{1}{(3)^2} \cdot \frac{1}{36} \\ &= 0.0168868924679 \end{aligned}$$

At this point, all the terms necessary to calculate p(SD) have been found:

$$\begin{aligned} \mathrm{p(SD)} &= \mathrm{p(t)} \cdot \mathrm{p(s)} \\ &\approx 8.04321581498 \times 10^{-8}. \end{aligned}$$

Furthermore, the terms necessary to calculate p(SC) have been found:

$$\begin{aligned} \mathrm{p(SC)} &= \mathrm{p(c)} \cdot \mathrm{p(a)} \\ &\approx 1.98032734182 \times 10^{-6}. \end{aligned}$$

One can now find the value for p(rule), since all of its constituents have been found:

$$\begin{aligned} \mathrm{p(rule)} &= \mathrm{p(SD)} \cdot \mathrm{p(SC)} \\ &\approx 1.59282001945 \times 10^{-13}. \end{aligned}$$

This leads to the value for p(P):

$$\begin{aligned} \mathrm{p(P)} &= \mathrm{p}(|\mathrm{P}|) \cdot \prod_{i=1}^{|\mathrm{P}|} \mathrm{p}(\mathrm{rule}_i) \\ &\approx 2.42080023503 \times 10^{-14}. \end{aligned}$$

One can also work out the probabilities for the other roots and morphological rules in this toy grammar; one can multiply these out to find the probability of the entire grammar. Under the assumptions laid out here, one assigns a probability of about $1.30917166486 \times 10^{-43}$ to this particular grammar. Although this is indeed an extremely small figure, it is important to remember that it represents the prior probability of this particular grammar in the space of all possible grammars. This figure of $1.30917166486 \times 10^{-43}$ can be compared with the probability associated with another grammar that generates exactly the same surface forms, but that has a different set of morphological and phonological rules.

2.5 Comparing the prior probabilities of grammars

The grammar described in the previous section can be compared with a grammar that generates the same data, but with a somewhat different set of morphological and phonological rules. One might ask which of the two grammars is more apt, in light of linguistic theory, and whether the objective function described in the previous section in fact prefers the more linguistically appropriate grammar.

The following grammar generates exactly the same fragment of the noun system of Turkish as the grammar given in the previous section. Rather than recognizing a phonological rule of vowel harmony, however, this alternative grammar accounts for the allomorphy present in the surface forms by supposing that nouns fall into two distinct inflection classes, and that each inflection class has its own rules for marking

the plural and the accusative:

- Roots:

$$\begin{aligned} &\text{IC-1:} \quad \textsc{nouns}, \{\langle \text{adam} \rangle, \langle \text{kar} \rangle\}; \\ &\text{IC-2:} \quad \textsc{nouns}, \{\langle \text{ev} \rangle, \langle \text{er} \rangle\}. \end{aligned}$$

- Morphological rules:

$$\begin{aligned} \text{IC-1:} \quad \{ \quad & \langle \varnothing \rightarrow \text{ler}, \text{suffix}, \text{d} = 0, \langle \text{plural} \rangle \rangle, \\ & \langle \varnothing \rightarrow \text{i}, \text{suffix}, \text{d} = 1, \langle \text{accusative} \rangle \rangle \}; \end{aligned}$$

$$\begin{aligned} \text{IC-2:} \quad \{ \quad & \langle \varnothing \rightarrow \text{lar}, \text{suffix}, \text{d} = 0, \langle \text{plural} \rangle \rangle, \\ & \langle \varnothing \rightarrow \text{i}, \text{suffix}, \text{d} = 1, \langle \text{accusative} \rangle \rangle \}. \end{aligned}$$

- Phonological rules: (none).

Jumping to the punchline, one can work out the probability of this grammar according the procedure given in the previous section; the result is p(hypothesis) $\approx 2.36934448746 \times 10^{-44}$. This alternative grammar has a significantly lower value for p(hypothesis) than the grammar given in section 2.4, for which p(hypothesis) $\approx 1.30917166486 \times 10^{-43}$. The more linguistically informed grammar, which recognizes Turkish allomorphy as a fact to be treated with a phonological process rather than with inflection classes, is associated with a probability about five times greater than that associated with the alternative. Which aspects of the grammars are responsible for this difference?

First, consider the set of phonological rules associated with each grammar. The prior probability of the first grammar incurs a certain penalty with the term p(P), representing the probability of the set of phonological rules, since stating the pro-

cess of vowel harmony reduces the overall value for p(hypothesis) somewhat. This can be compared with the situation in the second grammar, in which there are no phonological rules at all. In the first grammar, the probability of having a single phonological rule in the grammar in the first place is found to be 0.151982032211, while the parameters associated with the particular rule of vowel harmony result in an overall probability of p(P) = $2.42080023503 \times 10^{-14}$. In the second grammar, the probability of having no phonological rules at all is found to be 0.607928128843, and there are no other parameters to adjust p(hypothesis). Thus, the probability of the set of phonological rules in the second grammar is found to be about 10^{13} times greater than the probability of the set of phonological rules in the first.

The first grammar makes up for this penalty, however, because the phonological rule allows the allomorphy in Turkish to be stated without recourse to multiple inflection classes. This results in a higher overall value for p(hypothesis). The fact that the first grammar recognizes only one inflection class means that the p(ic) term in each p(root) term is always evaluated as 1.0, whereas it is evaluated as 0.5 in the second grammar that recognizes two inflection classes. Since there are four roots in both grammars, the value for p(R)—the probability associated with the set of roots—is eight times more higher in the first grammar than in the second.

The biggest gain for the first grammar, however, comes from the fact that recognizing vowel harmony as a phonological rule allows the set of morphological rules to be much smaller. Simply having fewer morphological rules is something that the term p(|M|) within p(M) prefers: this term evaluates as 0.0675 for the first grammar with two rules, and as 0.0243 for the second grammar with four rules. This results in a slight preference for the first grammar. The real issue, however, is the probability associated with each individual morphological rule. Because these rules are relatively complex, and each contains many parameters, the probability with any one rule is always fairly small. Stating just two morphological rules results in a higher value for

p(IC) than stating four—and as more inflection classes that are used to account for the same data, more inflectional rules must be deployed as well. For example, the probability associated with the rule marking the plural in the first grammar is just $1.8465037689 \times 10^{-6}$; the probability associated with the rule marking the accusative is $9.97112035208 \times 10^{-5}$. In the case of the second grammar, however, there are twice as many morphological rules to be stated, all with probabilities on the order of 10^{-5} to 10^{-7}. It is these differences in the set of morphological rules that result in the dramatically higher prior probability that is associated with the first grammar rather than the second.

The next chapter describes the way in which the present system seeds the search for the optimal grammar of word formation with an initial hypothesis, and how it moves from the initial state to nearby states by generating new, but related, hypotheses. These hypotheses are evaluated according to the measure of p(hypothesis) described in this chapter, and the system moves to successively more probable grammars, each of which is capable of generating the observed data.

Chapter 3

The search problem

The function described in the previous chapter assigns prior probabilities to grammars under certain assumptions about the kinds of grammars that are allowed. The objective function can be used to distinguish which grammar, out of several, is the most probable. So long as the objective function is appropriate, the grammar that it identifies will also be the most linguistically apt. However, it is not immediately clear how one can find, just on the basis of some set of training data, a grammar that is consistent with that training data, as well as reasonably probable. After all, the space of all possible grammars included in the probability distribution described by the objective function is not just vast: although the prior probability function assigns smaller and smaller probabilities to more and more complex grammars, the set of grammars that it allows is infinite. This chapter addresses the question of how one can search this space of grammars for a grammar which is consistent with some set of training data and which is also reasonably probable. The grammar that the learner arrives at after completing such a search will, ideally, be an appropriate grammar from the point of view of a linguist, insofar as it captures much of the knowledge that a native human speaker of that language would have of the same training data.

The approach taken in the present work is that one can construct an initial hy-

pothesis about the morphological grammar that corresponds with the training data according to certain principles, and then improve upon it gradually. This chapter describes a search strategy whereby one forms an initial hypothesis in which each lemma is assigned to its own inflection class; one can then make pairwise comparisons between that initial hypothesis and the hypotheses that can be built by modifying the initial hypothesis in certain limited ways. Whichever newly generated hypothesis gives the greatest increase in probability is adopted, and then used as the base from which further possible changes are made. The process of building frontier hypotheses by modifying the current hypothesis and greedily selecting whichever frontier hypothesis gives the biggest increase in probability continues until one arrives at a grammar from which no modification can result in a grammar with a higher probability.

This top-level control of the search is described more fully in section 3.2. For the moment, however, realize that the initial state is one in which every lemma is assigned to its own inflection class. The search proceeds by considering pairs of inflection classes that might be merged, and then selecting whichever potential merge results in the biggest improvement to the grammar's probability. At no point are inflection classes ever split up: the search simply marches forward by merging a pair of inflection classes at every step of the process. Eventually, the grammar arrives at a point at which no more merges can be made, or at least no merges that improve the overall probability of the grammar—this is the point at which the search terminates.

Of crucial importance to the success of such a search are the specific procedures that are used to generate frontier grammars on the basis of an input grammar, as well as the procedure that is used to formulate the initial grammar, and the contour of the search space. Note that if the contour of the search space contains any local maxima, certain search procedures and initial hypotheses will adopt these states as their final state, rather than the global maximum—the overall success of a search strategy requires that all three components interact appropriately. The selection of

the initial grammar is described in full in section 3.1. Put briefly, this stage of the search works by assigning each lemma to its own inflection class, and then finding a reasonable disposition of the material found in those forms between roots and inflectional rules. The set of operations that are used to generate frontier hypotheses is more involved, although they all work in essentially the same way—the procedures consider pairs of inflection classes and determine whether the two classes can be merged in a way that results in a net improvement to the probability of the grammar.

The most basic procedure for merging inflection classes—described in section 3.3—simply tries to merge two existing inflectional classes with a new, more probable analysis of the roots and rules that build surface forms, with the proviso that only morphological processes that make use of prefixation, suffixation, and infixation are considered. Finally, section 3.4 describes a procedure in which two existing inflection classes are merged by introducing a phonological alternation that applies over the entire grammar.

3.1 Finding an initial grammar

The search is initialized with a state in which every lemma is assigned to its own inflection class, and in which the assignment of material found in surface forms to roots and rules have been arranged in such a way that the result is a valid grammar, and one that makes reasonable assumptions about which segments belong to roots and which segments belong to affixes and infixes. Simply assigning each lemma to its own inflection class is straightforward; finding the appropriate analysis in terms of roots and rules within that inflection is slightly more involved.

One particular issue is that at the outset one does not know, for an inflection class, which sets of feature values are marked together with a single rule and which are marked separately, each with its own independent rule. Adding to the difficulty is the fact that any one feature value might participate in multiple exponence, perhaps

with other features, and perhaps with a different set of other feature values in each separate rule of exponence. Furthermore, some feature values may not be marked explicitly on the surface form of words at all. It is not practical, at least in the general case, to suppose that every possible association of feature values might participate in rules together, and to actually search for strings of segments that forms carrying each possible association of features show.

In order to determine the appropriate root and rule set for the inflection classes in the initial grammar, then, it is immensely helpful to have some idea of what sets of feature values are likely to be associated with one another in morphological rules. The sets of features whose values participate together in morphological rules are called "feature syndromes". This section begins with a description of the procedure that is used to search for the syndromes on each lemma; it goes on to lay out the way in which the analysis, in terms of roots and rules, for each inflection class is determined.

3.1.1 The syndrome search

Suppose that each word in a data set is associated with a tag in which a certain set of features are associated with certain feature values. At the outset, one does not know which features are marked together in the inflectional morphology of the language, and which are marked separately. One might use a procedure to determine which features are marked together in a syndrome—to begin with, one might suppose that every feature is stands as its own syndrome, by itself. If there is evidence that a particular feature constitutes a feature by itself, one can add it to the list of syndromes in the language. If, on the other hand, there is not evidence that a particular feature belongs to a syndrome by itself, but there is in fact enough data available to determine that this is a fact about the language, and not about the data that is available, then that feature can be combined with other features, and two-feature syndromes can be tested. Once again, in the event that no evidence for a syndrome can be found, but

enough data to find a syndrome is present, once can continue the search with more and more features. Finally, in the case that no evidence for a particular syndrome can be found because there are not enough surface forms available to either confirm or deny that such a syndrome exists, then that path of the search is terminated—no further potential syndromes will be created by adding features to the set of features for which no suitable evidence can be found to confirm or deny its status as a syndrome.

The results of the syndrome search are then used during the search for inflectional rules. It is automatically assumed that every inflection class may contain rules associated with no morphological features at all ("thematic" rules). It is also assumed that every inflection class may contain rules associated with the full set of morphological feature values on each surface form—that is, with the set of morphosyntactic feature values that fully specify the morphosyntactic features on each surface form. The search for syndromes is intended to determine what other, more restricted, inflectional rules might exist in a language—for example, it determines whether there is a rule to mark the plural as distinct from any rules that mark plural nominative or plural accusative. The output of the search for syndromes is loaded, along with these default syndromes, into the set of syndromes that serve as the basis for the search for inflectional rules in the later components of the search, which identify morphological rules, either as part of a singleton inflection class, or as inflection classes are merged.

This is essentially the procedure that is used to search for syndromes in the present system. The key is being able to know what constitutes evidence for a syndrome, and what constitutes having enough data to know that the lack of evidence is not simply problem of low data. The present system addresses these issues by searching for syndromes by looking at word forms associated with each lemma, and noting which pairs of words are related by an edit that is either the same or different from the edit that separates some other pair.

More concretely, suppose that the question at hand is whether some feature a

constitutes a syndrome. The word forms are divided into as many sets as there are feature values for a. For this example, suppose that one set tagged $a+$ is created, while another set tagged $a-$ is created. Pairs of words are now created, in such a way that one member is drawn from the $a+$ set, while the second member is drawn from the $a-$ set, and in such a way that the pairs match in terms of their feature values for features b, c, and d. If one wants to know whether or not there exists evidence for feature a as a syndrome, one simply needs to ask if the edits separating the $a+$ member of each pair from the $a-$ member of the pair are identical over all the pairs. If one wants to know whether or not there is enough evidence to be able to determine whether or not feature a constitutes a syndrome, one simply needs to ask whether two or more such pairs exist.

These edits are obtained by finding the set of the cheapest Levenshtein edits that separate the two strings. See Gusfield [30] for a description of the procedure used to the find the Levenshtein edit distance that separates two strings. For the present purposes, though, note that the Levenshtein edit distance is usually computed such that an insertion or a deletion edit has a cost of one, and a substitution edit has a cost of two. It is possible to construct a matrix that contains all of the cheapest edits that separate the two strings. In most instances where the Levenshtein edit distance is deployed, it is usually enough to simply find the cost of the cheapest edit separating the two strings. In the present context, however, it is necessary to have access to the full set of these cheapest edits. These can be read from the matrix, and they are given in the form of particular insertion, deletion, or substitution edits that can be used to map one string to the other.

For example, thinking back to the Turkish example, consider the edits that separate [ev] from [evler] and [evi] from [evleri]. The edits that separate [ev] from [evler] might be described as "insert l, insert e, insert r", while the edits that separate [evi] from [evleri] might be described in the same way: "insert l, insert e, insert r". Note

that the position of these edits in the word is not relevant for determining whether number is really an independent syndrome in Turkish—but the relative ordering of each edit, left to right, is important. In other words, the syndrome search does not care that the sequence "insert l, insert e, insert r" occurs at the end of the word in the [ev]∼[evler] case, but that it occurs in the middle of the word in the [evi]∼[evleri] case. The syndrome search does care, however, that both pairs show the same edits in the same sequence: in this case, the insertion of e follows the insertion of l, and the insertion of r follows the insertion of r for both sets of edits. If this left-to-right pattern were broken in some way, there would not be evidence that number is indeed a syndrome in Turkish, because the number feature alone would not determine whether a particular set of edits applies.

Procedure 1 Find syndromes associated with a lemma

given an IC containing a single lemma, l
initialize Q as the set of all the features can be associated with l
initialize S as the empty set, to be used to collect feature syndromes as they are found
for all elements R in Q **do**
    if R is a syndrome for l **then**
        remove R from Q and put it into S
    else
        remove R from Q, create a set M of sets in which R is paired with each of the features appearing on l, and put the elements of M into Q
    if there is not enough evidence to determine whether R is a syndrome **then**
        move to next available R
return S

Bear in mind that this syndrome search procedure can fail to find syndromes that really are necessary later on during the search. This can happen, for example, when several lemmas are each associated with only a single surface form in the training data. When these lemmas are eventually merged, it would helpful to know that several of the morphological features belong to distinct syndromes, but this fact cannot be uncovered when looking at solitary surface forms. In the current system, the only defense against this low-data issue is the hope that such low-data forms will in fact

be merged with high-data forms, in which the facts about feature syndromes can be determined more readily—but for a more detailed discussion of the performance of this syndrome search when confronted with realistic training data, however, see chapter 5.

Note, though, that the current system deliberately makes use of a syndrome search procedure that considers only one lemma at a time, because it does carry with it several important benefits. First and foremost, using this kind of search procedure means that this computationally costly search for syndromes, in which potentially any possible combination of morphological features might be identified as a syndrome, is undertaken only once per lemma, at the very beginning of the top-level search. Some other system, in which the syndrome search is revisited throughout the search for inflection classes, might be able to make use of information present in those inflection classes that it hypothesizes later on, but revisiting such a complex procedure constantly is likely to have very detrimental effects to the time complexity of the overall search, unless a good heuristic can be deployed. Additionally, looking at only one lemma at a time can help mitigate against the impact of phonological alternations which might happen to apply only to certain forms of certain lemmas, and which might obscure the fact that certain sets of morphological features really are marked together with the same edits, at least when one looks at the morphological rules that apply.

Additionally, several extra syndromes are included in the syndrome set before it is returned for the purposes of searching for morphological rules. Every set of syndromes is assumed to contain the empty set, $\varnothing$, to allow for the possibility of thematic rules. Every set of syndromes is also assumed to contain the full set of surface tags found on the set of input data that was used to form it. This precaution is taken for those cases in which the syndrome search must be abandoned, for some or all of the syndromes that actually exist in the language, because of a lack of evidence.

As a final note, consider the complexity of this search. This kind of complete breadth-first search for syndromes is, in principle, of exponential complexity with respect to the number of features that are found on the lemma. However, there are some very tight bounds on the complexity of the search in practice. As a result, the $O(a^n)$ behavior of a complete breadth-first search is not really an issue: it is important to realize that the search only progresses as long as there is enough data to form correspondence sets. In other words, the search procedure looks for increasingly more complex syndromes only if it is the case that the data cannot be accounted for with simpler syndromes and that the data necessary to put together correspondence sets actually exists. The result is that, in practice, the search terminates after a small number of steps and after finding all the syndromes for which there is evidence in the set of word forms.

3.1.2 Improving a singleton inflection class

As mentioned earlier, the initial grammar state is one in which each lemma is assigned to its own inflection class. This section describes the way in which the initial hypothesis for each of these singleton inflection classes is found. The procedure used here resembles the procedure used to merge two inflection classes from scratch, described in section 3.3, but there are certain aspects in which the search for roots and rules over one lemma and over several must differ.

The procedure begins with a set of word forms and a set of syndromes that are both associated with some lemma. The longest substring that is common to all the surface forms is found, and it it taken to be the underlying form of the root. In order to find this longest common substring, a generalized suffix tree is built out of the set of surface forms, using a naive algorithm that requires $O(n^3)$ time, with respect to the total number phones n in the input set. It is then possible to search this suffix tree for the longest common substring in time that is linear with respect to the number of

words in the set. (For a description of suffix trees, and several algorithms that can be used to build them in $O(n^3)$ and $O(n)$ time, see Gusfield [30]. Gusfield also describes the way in which such trees can be searched for longest common substrings.)

This string is marked in each of the surface forms so that it cannot be included in the rules that will be discovered later. This step is crucial: it is only by marking the root in this way that it can be preserved from being devoured by the greedy search for prefixes and suffixes that is about to grab unassigned segments and attribute them to particular morphological rules.

Recall that the syndromes, identified with the procedure just described, are stated in terms of morphological features rather than feature values, even though morphological rules are always stated in terms of a particular configuration of feature values within a syndrome. This means that the form of the syndromes discovered using the syndrome search must be changed in order for them to be useful during the search for actual morphological rules—after all, these rules must find correspondences between, on the one hand, particular prefixes and suffixes and, on the other, particular configurations of values such as *nominative*, *accusative*, *singular*, *plural*, *perfect*, and *progressive*. It is a straight-forward matter to find, for each syndrome passed to the function, the set of feature value configurations that those morphological features can take, but this step cannot be overlooked. After all, the syndrome {case, number} cannot be used to find a morphological rule directly: it must be converted into {nominative, singular}, {nominative, plural}, {accusative, singular}, {accusative, singular}, {oblique, singular}, and {oblique, plural} to be useful for finding actual morphological rules.

The search for morphological rules proceeds by looping over this set of feature value syndromes as long as new rules can be found. This search for rules works by considering just the set of word forms whose morphological tags make them consistent with the set of feature values in the current feature value syndrome. One can then determine the longest common substring that can be built from the left and right

edges on this set of surface strings—when such a string is found, it is identified as either a prefix or a suffix, and removed from further consideration.

Note that the procedure for finding such a substring does not require recourse to suffix trees. Suffix trees are deployed in order to find the longest common substring that is lurking somewhere within a set of surface forms so that it can serve as the initial hypothesis for that set's root. In this case, however, the start position of any potential longest common substring is known in all the strings that might contain it: for a prefix, it is always the leftmost phone in a word, and for a suffix, it is always the rightmost. This means that finding such substrings only requires linear time with respect to the number of forms that are found within the inflection class, and with respect to length of the shortest word in the set.

Whenever such a non-null longest common substring is found, a rule that relates that string of segments to the current configuration of feature values is identified. Determining whether the rule is a rule of prefixation or suffixation is straight-forward; determining the depth of the rule, relative to the other rules that have been found, simply involves counting the number of prefixes and suffixes that have been found on each form.

When a rule is found, the prefix or suffix that the rule was responsible for adding to a set of surface forms is removed from those surface forms. Once this prefix or suffix has been clipped off, new material is now available at the edges of these forms, so that correspondences between feature value syndromes and affixes can continue to be found on each pass through the loop of feature value syndromes. The search is complete when all material, except for the previously marked roots, have been removed from the surface forms of the words, and no more rules can be discovered. An outline of the algorithm is given in procedure 2.

At this point, it is possible to pause for a moment in order to reflect on how this procedure creates initial hypotheses about inflection classes, and how it is indeed

Procedure 2 Optimize a singleton IC

given a lemma l, and the set of surface forms W associated with it
let S = expand-features-to-values(syndromes(l))
let rules = the empty set
let root = longest-common-substring(surface forms in W)
mark root in each surface form in W
while elements can be added to rules **do**
    for each syndrome s in S **do**
        let H = the set of forms in W that contain s as a subset of the morphological feature values they carry
        let k = longest-common-substring(surface forms in H), such that k is found at either the left or the right edge of the string
        let r = a rule that marks s with the string k, as either a prefix or suffix, as appropriate, at a certain depth in the derivation
        push r onto rules
        remove k from the appropriate point from the surface forms in H
        revise W to include the clipped forms from H
return IC = ⟨root, rules⟩

guaranteed to find a solution. First, note that it is alway guaranteed to find an appropriate root, because it picks the longest substring common to all surface forms for this purpose. (In a case where no such substring exists, the procedure may in fact posit a null root.) The rest of the problem is merely one of assigning the leftover material to the set of syndromes in some plausible way to form inflectional rules. The procedure does this by cycling through the set of syndromes and assigning material to each syndrome—recall that this cycle continues until all the remaining phonological material has been exhausted, and that every syndrome set contains a set of "garbage bin" syndromes that give the full morphosyntactic representations of each of the surface forms associated with that lemma. As a result, the search always begins by finding a consistent root; it then proceeds to soak up any remaining phonological material by assigning it to an inflectional rule.

Notice that this procedure searches only for prefixes and suffixes, not infixes. In the case that a form does, indeed, contain infixed material, and this infixed material causes the form to differ from some other form belonging to the same lemma, the

learner will simply identify the infix and the material between it and the nearest edge of the word as either a prefix or suffix. This mistaken analysis of an infix and part of the root as either a prefix or suffix can only be resolved during later stages of the top-level search—described below.

3.2 Top-level control of the search

At the highest level, the search for the most apt grammar is controlled by a loop that takes the current state of the grammar as its starting point, and then generates a set of frontier grammars that can be made from the current grammar according to a local successor function. This successor function operates by taking the set of inflection classes that are present in the current grammar, and attempting to merge those inflection classes in certain ways, while ensuring that the frontier grammars are also consistent with the training data.

The probabilities of these frontier grammars are compared against the probability of the current grammar, and whichever frontier grammar is associated with the greatest improvement is selected as the new current grammar. In the event that no new frontier grammars can be generated from the current grammar, or in the event that none of the frontier grammars actually represent an improvement in terms of p(grammar), the search terminates and the current grammar state is returned as the most apt hypothesis. The outline given in procedure 3 simply formalizes this description of the greedy local search.

Procedure 3 Top-level control of search

given an initial grammar, g
while improvements to g can be discovered **do**
    let $F = \text{successor}(g)$
    let $g = \max_f(\text{p}(f))$ where $f \in F$
return g

This form of top-level control of the search can be described as a "greedy local

search" or "hill-climbing search" (see, for example, Russell and Norvig [56]). The search procedure is greedy in the sense that it always selects the frontier grammar that results in the biggest improvement in p(grammar), without any kind of look-ahead or heuristic for guessing whether that selection will indeed lead to the global optimum later on. (Note that this search also qualifies as an "agglomerative search", in the taxonomy of search types, because it operates by merging smaller sets into larger sets, without ever breaking up members of a set once they have been placed in a set together.)

Searches of this kind are known for the weakness that they do not guard against the possibility that a local maximum might trap the search when a better, global maximum exists elsewhere in the search space. In this case, the global maximum is only accessible by backing off from the local maximum, and finding a state for which the objective function gives a lower score, but which will eventually allow access to the global maximum. Such a situation, in which the search state is trapped at a local maximum, is in principle possible in the current system.

Another potential problem with this kind of hill-climbing search—or really, any search in which the search states are individual grammars, and the search for the optimal grammar proceeds by moving from grammar to grammar to grammar on the basis of certain modifications that are available according to a successor function—is the possibility that the global maximum exists in a part of the space of grammars that simply cannot be reached by following available steps from the initial state. In other words, there is no guarantee that the space of grammars provided by the probability distribution over grammars and the space of grammars provided by the search procedure are congruent. There could, for example, be large sectors of the space of grammars provided by the probability distribution that simply can never be reached by following the search procedure's successor function. This is an issue which will be addressed in sections 3.3 and 3.4, where the specific components of the

successor function are discussed, and in chapter 4, in which more complete examples are considered. The bottom line is that these two spaces of grammars need not be perfectly congruent for the system to be successful, so long as the search space that the successor function can access includes grammars that are in fact linguistically appropriate, in terms of the kind of knowledge that a human linguist would attribute to a native speaker of a language.

The specifics of the successor function are described in the next two sections of this chapter. The successor function always works by attempting to merge pairs of inflection classes. One kind of merge that is available is a plain merge, described in section 3.3, in which inflectional classes are essentially put together from scratch. Under this procedure, the lemmas from two existing inflection classes are put together into one and a search for an assignment of roots and rules that return a high value for p(IC) begins with no knowledge of the previous morphological analysis. In contrast, section 3.4 describes a procedure that finds merges that rely on the discovery of phonological alternations. This final kind of merge is the most complex, since it has consequences for the forms that are posited in the other inflection classes that have already been established.

A few words are in order about the time complexity of the top-level search. The time complexity of the overall search depends on the number of word forms associated with each lemma, and the length of each of these word forms, but from the point of view of the top-level search, the number of inflection class merges that need to be considered is strictly a function of the number of inflection classes in the search's initial state. Note that for a search involving n inflection classes there are $\frac{n\cdot(n-1)}{2}$ unordered pairs of inflection classes that exist. This means that there are exactly $\frac{n\cdot(n-1)}{2}$ unordered pairs of inflection classes that can be tried in the merge from scratch procedure at any stage in the search, and $n \cdot (n-1)$ ordered pairs of inflection classes that can be tried in the merge and find phonological rules procedure. Also note

that each merge takes two inflection classes and turns them into one, so each merge reduces the number of inflection classes in the grammar by one. This means that the search can proceed for at most n rounds before it ends, if there are n inflection classes in the initial state. If, at each stage i, there are $\frac{3}{2} \cdot (i^2 - i)$ merges that must be considered, then there are no more than $\sum_{i=1}^{\mathrm{n}} \frac{3}{2} \cdot (i^2 - i)$ merges that can possibly be considered in the course of an entire run. This summation reduces to $\frac{\mathrm{n}^3 - \mathrm{n}}{2}$. (For information on reducing polynomial summations, see Graham, Knuth, and Patashnik [29].) Thus, the time complexity of the search is bounded by $k \cdot \mathrm{n}^3$, where n is the number of inflection classes in the initial grammar. Note, however, that this figure only addresses the time complexity in terms of the number of input lemmas—the number of word forms associated with each lemma impacts the time required to form hypotheses about merging pairs of inflection classes.

There are, however, certain ways in which better performance can be obtained. In particular, note that merging any given pair of inflection classes leaves all the other pairs untouched, at least in the merge from scratch case. This is not quite true of the merge and find phonological rules case: identifying a new phonological rule changes the landscape of the search in the sense that it may make certain other merges less costly than they would have been otherwise, or it may make certain other merges impossible, given the phonological rules that it introduces to the grammar. Suppose, then, that the search is being conducted with only the merge from scratch procedure in place. The improvement in the probability of the grammar that is gained by merging any particular pair of inflection classes is the same, no matter what other merges have taken place. This means that memoization can be used to speed a search that makes use only of the merge from scratch procedure. Suppose that there are n inflection classes in the initial state. In the first round, all $\frac{\mathrm{n} \cdot (\mathrm{n}-1)}{2}$ pairs must be considered in the merge from scratch procedure. In subsequent round i, however, there are not $\frac{\mathrm{n} \cdot (\mathrm{n}-1)}{2}$ pairs to consider, since most of those pairs were already considered during the

previous round. Instead, there are only $\mathrm{n} - i$ pairs to consider at each following round i, since there are only that many pairs involving an inflection class that did not exist previously. Using this form of memoization, then, it is possible to test $\frac{\mathrm{n}^2}{2}$ pairs of inflection classes in the first round, followed by $\mathrm{n} - i$ pairs of inflection classes in each of the next $\mathrm{n} - 1$ rounds. In other words, the maximum number of merges that can be considered under this search strategy can be given by

$$\frac{\mathrm{n} \cdot (\mathrm{n} - 1)}{2} + \sum_{i=1}^{\mathrm{n}-1} i$$

with the first term representing the number of merges that must be considered in the initial state, and the second term representing the number of merges that must be considered in all following states. This expression reduces to $\mathrm{n}^2 - \mathrm{n}$, so the time complexity of the search is bounded by $k \cdot \mathrm{n}^2$ when this kind of memoization is employed.

3.3 Merging a pair of inflection classes from scratch

The most basic procedure employed within the successor function is to attempt to merge two inflection classes essentially from scratch. The idea is to find the best assignment of roots and rules—or at least something close to it—for the set of lemmas in the two input inflection classes without relying on the assignment of roots and rules found in the input inflection classes. In order to do this, one takes the set of lemmas, the set of surface forms, and the set of syndromes from the two input inflection classes, and searches for a hypothesis for the roots and rules so that the resulting inflection class can account for the observed surface forms while representing some kind of improvement in terms of p(grammar). Of course, some inflection classes simply cannot be merged with one another, and the procedure must return null in such a case.

The procedure begins by taking the intersection of the feature value syndromes associated with the input inflection classes. This is based on the assumption—which is hardly true—that any syndromes present in the intersection of two inflection classes can be found on those individual inflection classes. (See sections 3.1 and 5.4 on this point.)

The search for rules proceeds from the outside of words to the inside of words, identifying prefixes and suffixes as they are discovered, and clipping them from the set of forms being considered. Note that this search is greedy, in the sense that every syndrome, when it is given the opportunity to identify a prefix or suffix with which it can be associated in an inflectional rule, will grab the longest string that it can possibly grab at the either the left or right edge of the string. The key condition that must be met is that this string must be found at that same location in all word forms associated with that syndrome—which is indeed critical, because that is the string that the newly-identified inflectional rule is meant to introduce.

The time complexity of this search is in fact linear with respect to the number of segments in the surface forms being considered, and the number of syndromes in the inflection class. (Notice that finding such longest strings is nowhere near as complex as finding the longest common substring in the midst of a surface form, since its beginning point is known—only the endpoint must be determined.) This portion of the search is almost exactly the same as this portion of the procedure used to improve singleton inflection classes. Marking a rule as a prefix or suffix at some particular depth is performed in the same way, however. One important difference between the procedures, however, is that this merge procedure does not mark the roots ahead of time using a longest common substring algorithm—instead, the roots are simply taken to be whatever segments are left over at the point at which no more morphological rules can be found. This fact about the way in which roots are identified requires that one check, for each lemma, whether a consistent root has been identified for it.

If not, the search state includes some mistake about the assignment of segments to rules, and the procedure simply returns null, indicating a failure to merge the inputs.

Another difference between the procedure that is used to optimize over singleton inflection classes and the procedure that is used to merge inflection classes is the fact that it is possible to search for infixes when merging inflection classes. The search for infixes relies on the fact that looking at a set that contains several roots makes it much easier to discover broken roots than does looking at a set that contains just one. The search for infixes waits until the search for prefixes and suffixes has progressed to the point at which the right and left edges of surface forms can be taken as either prefixes or suffixes. At this point, the procedure that searches for infixes begins, working in the same way as the procedure that search for prefixes and suffixes. The only difference is that the search for infixes does not start at and offset of zero from the left or right edge of the word, as it does during the search for prefixes and suffixes. Instead, this search starts at offset i, where i iterates from 1 up to the length of the shortest word in the set.

Note, though, that the search procedure always attempts to find prefixes and suffixes, if any are available, before it turns to looking for infixes. In other words, it first attempts to look for prefixes or suffixes, and when no viable prefixes or suffixes can be found, the search for infixes begins. Once every syndrome has progressed through the search for infixes, the search resumes with a search for prefixes and suffixes again—this repeats until no more rules can be discovered. Notice that it really is necessary to exhaust the search for prefixes and suffixes before turning to the search for infixes: the reasoning here is that parts of prefixes and suffixes might be incorrectly identified as infixes if one allows the search for infixes to proceed before all the prefixes and suffixes that can currently be discovered have been exhausted.

Note that this offset is stated in terms of segments from the right or left edge of the word. A more sophisticated approach might be to count the offset in terms

of vowel segments, consonant segments, or positions within syllables, but this one catches basic instances of infixation. The system can be modified to account for more complex instances of infixation as long as one is willing to represent a more complex set of points at which infixes may be placed.

The search is set up in a kind of a loop: it begins by attempting to pass through the set of syndromes; for each syndrome it first looks for a prefix or suffix that would be consistent with that syndrome. Then it loops over those syndromes looking for infixes. If, at any point, any prefix, suffix, or infix is found, the search is allowed to continue again with the full set of syndromes once the set has been exhausted. Once an entire pass through the set of syndromes has been completed without finding any rules, the search is considered to be complete. At this point the remaining surface forms can be compared, with the identified prefixes, suffixes, and infixes clipped out. If, for each lemma, the forms are identical, the search state has arrived at an analysis in which those forms can be taken as the roots for each lemma. If, on the other hand, the forms are not identical for each lemma, the search state has crashed without finding a workable analysis—because no consistent roots can be found. The procedure then either returns null, to indicate that no workable analysis can be found, or the new, merged inflection class. The procedure for merging inflection classes from scratch is laid out in outline form in procedure 4.

A small example that make use of the Turkish data will, perhaps, make certain aspects of this procedure more clear. Although this data is exceptionally simple, it should make the order in which these searches take place more clear. Suppose that

Procedure 4 Merge two inflection classes from scratch

given two inflection classes, $\mathrm{IC}_a, \mathrm{IC}_b$
let S = intersection of feature value syndromes found on $\mathrm{IC}_a, \mathrm{IC}_b$
while rules can be found **do**
    for all elements s in S **do**
        try to find the longest string k present on the left or right edge of all the forms consistent with s
        create a new rule associating s with k; revise the set of word forms so that the surface forms no longer include k
    if no prefix/suffix rule can be found at offset = 0 from the edge **then**
        look for infix rules at offset = 1..length of shortest string from the edge, using the same procedure used to find prefixes and suffixes
if the remaining forms are consistent over each lemma **then**
    initialize IC = ⟨ roots, rules ⟩
return IC, or null if no IC can be found

one is dealing only with the following forms:

en,	width,	⟨singular⟩
enler,	width,	⟨plural⟩
ev,	house,	⟨singular⟩
evler,	house,	⟨plural⟩

In order to simplify the situation, case has been set aside. The set of syndromes, therefore, contains { ∅ , ⟨ plural ⟩ , ⟨ singular ⟩ }. It is important to note that this search is, in fact, sensitive to the order in which syndromes are addressed—for this reason, the search procedure always sorts them alphabetically. This ensures that the same order is used whenever two inflection classes are merged. Assuming that the syndromes have been sorted in this order, the search begins with ∅, which, during its search for prefixes and suffixes, is capable of identifying [e] as a thematic prefix at the left edge of the forms. As a result, the initial segments have been removed from

consideration:

n, width, ⟨singular⟩

nler, width, ⟨plural⟩

v, house, ⟨singular⟩

vler, house, ⟨plural⟩

The search continues with ⟨ plural ⟩. The search procedure cannot, therefore, even attempt to associate ⟨ plural ⟩ with the prefix [e]; instead, the search procedure identifies [ler] as a suffix associated with ⟨ plural ⟩. Identifying this rule results in the following situation:

n, width, ⟨singular⟩

n, width, ⟨plural⟩

v, house, ⟨singular⟩

v, house, ⟨plural⟩

Now the search procedure attempts to find a rule for ⟨ singular ⟩, but no such rule can be found. This means that the search for infixes begins—the procedure attempts to find infixes the whole set of syndromes, { ∅ , ⟨ plural ⟩ , ⟨ singular ⟩ }, in these pared-down forms, but of course this is not successful. The search procedure cycles around one more time to look for prefixes and suffixes, in case removing the plural suffix on the last cycle had exposed a thematic element that the ∅ syndrome had not been able to capture before, because this element might still have been trapped by a suffix. Therefore, the search for prefixes and suffixes continues with { ∅ , ⟨ plural ⟩ , ⟨ singular ⟩ } on the cut-down forms:

n, width, ⟨singular⟩

n, width, ⟨plural⟩

v, house, ⟨singular⟩

v, house, ⟨plural⟩

This is not successful, in the sense that no prefixes or suffixes can be found, so the procedure passes through its search for infixes for a final time. This too is not successful, and as a result, the search is now in a state in which every syndrome has attempted to find a prefix, suffix, or infix at least once, and failed. Now, and only now, is it safe to assume that the procedure will not be able find any more inflectional rules. The search procedure now determines that [n] can stand as a consistent root for "width", and [v] can stand as a consistent root for "house". These are returned as the new hypothesized roots, and the inflection class is completed with the rules that have been identified. (The assignment of each rule to a particular depth is based on the order in which it was discovered.)

Generally speaking, most sets of forms do not contain a common string at the extreme left edge like this. If this example made use of [kar] and [adam], for example, no prefix like [e] would have been identified—although a spurious infix, [a], might well have been identified, as standing one segment from the right edge of the word. In some cases, however, the null syndrome can end up capturing several segments at the left edge, right edge, or middle of a word. The fact that this search function is greedy in its assignment of phonological material to rules means that this kind of error is certain to occur, whether or not the spurious inflectional rule is actually doing anything to improve the probability associated with the newly created inflection class. One key question, however, is whether or not the resulting inflection class will later be merged with some other, more diverse inflection class whose members can iron out these idiosyncracies. For example, later on the search, [ev] and [er] might have the opportunity to merge with a word like [kemik], "bone". Depending on the probability of that new inflection class as compared with the two input inflection classes, the merge might or might not be accepted. This kind of diversity, though, is the key to avoiding these kinds of spurious analyses—the merge procedure cannot defend against them.

Note that the successor function is free to attempt to merge all pairs of inflection classes each time it is called, at least given the restriction that only inflection classes of the same grammatical category can be merged. As mentioned earlier, this means that there are (at most) $\frac{n^2-n}{2}$ possible pairs to consider, but it is possible to use memoization to keep track of just how much improvement most of these merges would lend to p(grammar). As a result, it is not necessary to call this procedure every time—instead, it only needs to be called once for each pair of inflection classes, and the result can be looked up again later, as needed. Also, it should be mentioned that the top level search selects best merges first which in turn lead to better merges being available later on.

3.4 Searching for phonological alternations

The technique used to search for phonological alternations is rather different from the technique used to merge inflection classes from scratch, even though both procedures operate by attempting to merge a pair of existing inflection classes in such a way as to improve p(grammar). The key difference can be summed up thus: the merge from scratch procedure takes two existing inflection classes and erases all preconceived notions about the representations associated with roots and rules, and it starts by looking for viable rules of inflectional morphology with the set of syndromes that have been passed to it from the input classes. It successfully returns an output when it has identified a set of inflectional rules and root representations that can correctly account for the surface forms in the data, and it returns null if no such solution can be found.

The procedure that searches for phonological alternations, on the other hand, takes two inflection classes and marks one as the host and the other as the guest. The search procedure then attempts to make use of the host's inflection rules in order to account for the surface forms associated with both the host and the guest inflection

classes. Sometimes this can be done successfully without positing any phonological alternations, but sometimes the procedure finds that there are certain segments whose identity is not predicted accurately according the hypothesis provided by the host inflection class. In these instances, the system considers the single-character edits that might be employed in order to account for the surface forms in the guest inflection class with the inflection rules found in the host class.

This is done by generating surface forms on the basis of the rules that exist in the host inflection class, and the roots that are posited by both the host and the guest inflection classes. The resulting forms are then aligned with the actual surface forms. For each such pair of forms, the full set of the cheapest Levenshtein edits that separate the true surface form from the surface forms that are generated using the host's rules are examined (recall the discussion of the sets of cheapest Levenshein edits in section 3.1.1). In some cases, the two strings align perfectly and no new phonological rules need to be posited. In other cases, the strings are separated by certain edits that are not more than a single character long. In these cases, the search procedure considers whether or not it is possible to include phonological rules in the grammar in order to effect those edits.

Once the procedure has identified the possible alternation, the next step is to consider all the ways in which that alternation might be codified in a phonological rule, given the space of phonological rules that the present system allows, and given the facts about the surface-true phonotactics that can be found in the training data. First, notice that it is not initially clear whether the alternation is $a \rightarrow b$ or $b \rightarrow a$, so these two possibilities (along with the appropriate revisions to the roots' URs or the rules representations) must be considered.

This means that, for both $a \rightarrow b$ and $b \rightarrow a$ a full set of contexts must be checked. One might have an idea of what is meant by "full set of contexts" from the description of the context of phonological rules in chapter 2. Essentially, one

must look at the possible triggering environments that are present in the forms in the inflection class, consisting of the forms X__Y, WX__YZ, X__, __Y, XY__, and __XY on the full segmental representation, the vocalic tier, and the consonantal tier. Then one must check to see whether a ever appears in any of these environments—if it does, then the $a \rightarrow b$ alternation cannot be triggered by the appearance of a in that environment. The same holds true for b: if b ever appears in any of the posited triggering environments, then the alternation cannot be of the form $b \rightarrow a$ in that context. The result is that many, if not all, of the posited alternations will be discarded. The basic idea here is to use the facts about the surface distribution of segments in the training data in order to identify which phonological alternations might be true, and which cannot be. Needless to say, this procedure can only find alternations that are surface true; it can also make the mistake of attempting to reduce a chain of phonological rules into a single alternation.

Consider, very briefly, an example drawn from Turkish. Suppose that one is trying to merge an inflection class containing [kar] with an inflection class containing [er], and that each inflection class contains a rule used to mark the accusative. If [er] is the host class, its rule will be deployed—so the search procedure attempts to mark the accusative form of [kar] with the suffix [i]:

[kari]

This is inconsistent, however, with the actual observed forms:

[karɨ]

Therefore, the procedure must consider the [ɨ] → [i] rule, as well as the [i] → [ɨ] rule. For each rule, there are a wide variety of possible triggers. For instance, if the rule is [i] → [ɨ], is it triggered by the segment to the immediate left of the segment

that changes? In other words, could the rule be [i] → [ɨ] when it follows [r]? Or is the rule [i] → [ɨ] when it precedes the end-of-word symbol, #? These possibilities, and all the other possibilities that [kari] → [karɨ] presents, must be considered. This means that rules on the general tier, the vocal tier, and the consonantal tier must be considered, and that rules in all possible configuration of left-side and right-side triggering environments, must also be considered. For example, the [i] → [ɨ] might have any of the following triggering enviornments on the general tier:

	_	#
	_	##
r	_	
ar	_	
r	_	#
ar	_	##

The triggering environments for the vocalic tier and the consonantal tier must be constructed in a similar way.

In a realistic setting, most of these possibilities are discarded, because they fail to be surface true—for example, in the present example, it is clear that [i] → [ɨ] when it follows [r] is not a valid rule, because [eri] stands as a clear counter-example. In cases where several triggering environements are possible, given the training data, however, preference is given to triggering environments that are associated with higher probabilities, unless there are already phonological rules present in the phonological system. In that case, preference is given to whichever form of stating the rule results in the most probable phonological system, according to the objective function—for more on this aspect of the search, see below.

Note that the time complexity of this final check is not nearly as large as it might appear at first glance. The goal is to see if there are any instances of some segment

a that are adjacent, in the appropriate way, to some context c, which is thought to trigger the mapping of $a \rightarrow b$. This does not actually require that one scan the entire training data looking for instances of a in the key context with c. Instead, it is possible to create a hash table in which the keys are the various segments of the language in all the possible configurations that matter from the point of view of triggering these phonological rules. It is then possible to look up, in essentially constant time, for each possible rule, whether it is indeed a possible phonological rule in the language.

Procedure 5 Merge inflection classes while finding phonological rules

given two inflection classes, $\mathrm{IC}_a, \mathrm{IC}_b$, a set of existing phonological rules P, and a set of training data D
generate S surface forms using the roots in $\mathrm{IC}_a, \mathrm{IC}_b$, and the rules in IC_a
for all elements s in S, compare s with the corresponding true surface form d **do**
    for all edits e separating s from d **do**
        determine whether e could exist as a phonological rule, given the distribution of surface phones in the training data, and some possible context found within s
if the edits e, and contexts, c required to map s to each t exist as phonological rules in P, or can be added to P **then**
    return revised P, new inflection class IC_n
else
    return null

This procedure, as described so far, is able to identify alternations between segments and the context for these alternations; it also revises the underlying representations associated with either roots or rule's suffixal material in the appropriate way. Turning the alternation stated in terms of segments into an alternation stated in terms of phonological features is a separate step. This separate step is crucial from the point of view of representing the grammar in a way that parallels the linguists' grammar, and in a way that can have its probability assessed appropriately.

At a high level, the conversion can be described thus: the learner maintains a collection of all the phonological alternations that have been discovered by previous applications of this merge procedure. These previously collected alternations are stated in terms of phonological features, not in terms of segments. When a new possi-

ble alternation is discovered, the procedure attempts to see whether it is a special case of an already identified alternation, or whether it is matches an existing alternation only in part. In the first case, where the new alternation is not a new alternation at all, but rather a special case of an existing alternation, no modification to the collection of phonological alternations is made. In the second case, where some features of the new alternation match an already existing alternation, the procedure will attempt to expand the coverage of the existing rule to include the new alternation—but if this is not successful, the new alternation will be treated as a completely new rule. Bear in mind that there may be several different ways in which the new alternation can be stated. In such cases, the learner maintains all of these statments of the rule in memory, and it attempts to match the new rule with the existing rules in a way that results in the highest probability for the phonological grammar.

It should also be pointed out that there is not a perfect congruence between the space of phonological grammars allowed by the generative model and the space of phonological grammars allowed by this search procedure. There is, however, significant overlap between the two spaces, and this is the area in which the phonological grammars discovered by this system exist. Recall from the previous chapter no phonological rule ever feeds itself, or any other rule: this is a property of the fact that rules are said to apply simultaneously. At the same time, however, it is a condition of phonological rules that are discovered by this system that every phonological rule that is discovered must be surface true. These phonological rules must be surface true in the sense that there can be no situations in the training data where the conditions for a phonological rule is met, but it appears not to have applied. This is just the situation that exists when rule 1 would provide a context for rule 2 to apply, but in which rule 1 and rule 2 have been applied simultaneously: the context for rule 2 is present in the output form, because rule 1 formed it, but rule 2 does not apply, because the conditions for its application were not met in the input form. The result

is that the search function allows sets of rules that feed one another in order to apply maximally, whereas the objective function allows sets of rules that do not interact—and which potentially counter-feed one another. Although the search function and the generative system are not perfectly congruent in this regard, the resulting system is still one in which many phonological systems of the natural languages of the world can be described.

As a final note, it should be mentioned that this procedure relies on having both a guest and a host class that are set up correctly: the host class must have the correct rules and roots in place, while the guest must at least have the correct roots in place. In this regard, this search procedure is much less forgiving than the merge from scratch procedure, which can successfully merge two inflection classes, even when the input inflection classes do not analyze their forms correctly, or even reasonably.

This search for phonological alternations is the last procedure that stands as a component of the search system. The next chapter presents three case studies in which the system is tested on tagged data, and its results are evaluated both quantitatively and qualitatively.

Chapter 4

Three case studies

This chapter presents the results of several experiments in which the machine learner was presented with input data from various natural languages, and the results evaluated. The languages Classical Arabic, Classical Latin, and Modern Turkish serve as the subjects for these experiments, because they represent a large cross-section of the kinds of inflectional morphologies found. This allows the results to be assessed in terms of three case studies, in which the performance of the learner can be assessed in terms of how it performs on a particular kind of natural language.

For each of the three languages, the learner is tested under several different configurations. In the first configuration, the learner is tested with a small data set with both the "merge from scratch" and the "merge while finding phonological rules" are both available as possible functions during the "move" portion of the evaluate-and-move search cycle. In the second configuration, the same data set is tested, but with only the merge from scratch variety of the move function available. In the third configuration, a much larger data set is tested, again with only the merge from scratch variety of the move function in place. The results for these three configurations are presented for Classical Arabic in section 4.3, Classical Latin in section 4.4, and Modern Turkish in section 4.5.

Before diving into the specifics of the languages that are evaluated, however, a few words are in order on the kinds of evaluation metrics that are used to assess the learner's performance.

4.1 Evaluation metrics

In one sense, a very clear standard for evaluating the grammars output by the learner exists: the kind of grammars of morphology and phonology that commentators have produced for centuries on languages. These grammars wrap up a description of the morphological principles and phonological alternations that are present in a given language, and they reveal a great deal about the principles of morphology and phonology that are admitted by their authors. Of course, there are certain drawbacks to using grammars written by human linguists. For one thing, it is not necessarily the case that the grammars assembled by human linguists are necessarily the optimal grammars, under the objective function described in the present work, for generating the training data. It could well be that some other grammar, as yet undiscovered by both the machine learner and by human linguists, in fact captures the facts about the training data in a way that is assigned a higher probability by the objective function. If such a grammar could be discovered, one might wish to compare the properties of this grammar with the grammar actually identified by the machine learner. At the moment, however, it is impossible to identify such a globally optimal grammar, so in the mean time there is still good reason to use the linguists' grammars as a tool for evaluating the outputs of the present system.

One reason that linguist-generated grammars are still one of the most appropriate measure of the success or failure of the present system is that they are written in a form very similar to that of the output of the learner described here. After all, comparing a grammar with a grammar at least allows one to compare objects of the same fundamental type—and one can compare this with the difficulties or short-

comings that are associated with making a comparison between a human linguist's segmentation of a set of words and a machine learner's segmentation of the same set of words. The learner presented in this work, however, assumes that the facts about the inflectional morphology and phonology of a language cannot be captured just in an object such as a segmentation of a set of words, and that a more elaborate grammar is necessary to capture the knowledge that associated with a particular natural language's inflectional morphology and phonology. Therefore, the comparisons that are used for evaluation in this work are, fundamentally, comparisons that are made of one grammar against another.

Additionally, linguists tend, at least impressionistically, to favor grammars that score well in terms of the objective function described in chapter 2. Even though linguists' grammars may not be truly optimal, in the sense of this objective function, they still capture many properties that are thought to be essential facts about knowledge of the morphology and phonology of the languages that they describe, and they often seem to do this in a way that performs well in terms of the objective function presented here.

Note, though, that this kind of comparison between the learner's grammars and the linguists' grammars requires a somewhat novel approach to evaluating the "alikeness" of two grammars. Even if two grammars both conform to certain standards, finding a way to compare the two is not necessarily trivial. This section describes the measures that are used to evaluate one grammar against another.

The basic principle is this: if two grammars assign the same lemma to the same inflection class, they are in agreement; if two grammars assign the same lemma to distinct inflection classes, they are in disagreement. However, it is not obvious how to define a particular inflection class as being the "same" inflection class across two different grammars. After all, if two different grammars make use of exactly the same set of inflectional rules in two different inflection classes, and they assign exactly the

same lemmas to that class, the inflection classes are clearly equivalent. This metric seems overly strict, however—it does not account for the fact that different grammars can account for the same facts in the same set of words in potentially different ways. Clearly, exact identity between the rules in one inflection class and the rules in another inflection class is too strict a standard.

Instead, for the purposes of this evaluation, we treat inflection classes as partitions of sets. If two grammars partition the lemmas in the training data into the same set partition, they are perfectly equivalent; the extent to which two set partitions resemble one another can be determined quantitatively in a reasonably straight-forward way. Of course, it does not make sense to refer to the names of inflection classes, since names like "first inflection" and "second inflection" are clearly arbitrary. Instead, the measure depends on whether pairs of lemmas are assigned to the same or different inflection classes in each of the two set partitions. (This notion of "pairwise" comparisons being used to evaluate set partitions is called the Rand index; more information on it can be found in Rand [54].)

A further wrinkle develops from the fact that, for any grammar, the training data can under-determine the assignment of lemmas to inflection classes: when one examines a particular set of training data, and one compares it with a particular grammar, it is not always the case that the grammar will assign that lemma uniquely to a particular inflection class. This is because the assignment of a particular lemma to a particular inflection class may be fully determined in principle, if perfect knowledge of all the word's forms are available; in practice, however, only a few forms may be available in the training data, so the assignment is under-determined by the available data. The evaluation metrics that are used to assess the learner's grammars against the human linguists' grammars are formulated in terms of whether the same lemmas are assigned to the same inflection classes in the machine learner's grammar are assigned to the same inflection class in the human linguists' grammar.

These scores refer to pairs of lemmas being assigned to the same inflection class by both the learner's grammar and the linguists' grammar as true positives, while pairs that are put together under the learner's grammar but not the linguists' grammar are referred to as false positives, and those pairs that are put together under the linguists' grammar but not the learner's grammar are called false negatives. Note carefully, however, the way in which these scores account for the fact that the training data and the set of inflection classes in the linguists' grammar can under-determine the assignment of lemmas to inflection classes in that grammar—if the linguists' grammar can find some way of assigning two lemmas to the same inflection class, then the assumption is that those two lemmas do indeed belong in the same partition for the purposes of the comparison with the learner's grammar. If the linguists' grammar cannot find some way of assigning two lemmas to the same inflection class, however, the assumption is that those two lemmas do not belong in the same partition for the purposes of the comparison with the learner's grammar.

With the values for true positives, false positives, and false negatives in place, the score for precision, recall, and f-score can be found in the standard way:

$$\text{precision} = \frac{\text{true positives}}{\text{true positives} + \text{false positives}}$$

$$\text{recall} = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}}$$

$$\text{f-score} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

For more on the values of precision, recall, and f-score, see van Rijsbergen [63].

In order to perform an evaluation of a particular grammar output by the machine learner, then, it is possible to use these precision and recall scores to compare the extent to which it matches a hand-built grammar. It is important to account for the

fact that both the machine learner and the human linguists' grammar may underdetermine the assignment of any given lemma to a particular inflection class, but the key fact remains that it is still possible to draw comparisons for two grammars of inflectional morphology on this basis.

Bear in mind that there are several other ways in which one might want to evaluate grammars. After all, the assignment of lemmas to inflection classes is certainly important, and it is a good sign if the assignment made by the learner's grammar resembles the assignment made by the linguists' grammar, but there are other aspects of these grammars that must be evaluated as well.

First, consider the standards used to evaluate phonological rules. For the set of rules present in the final form of the machine learner's grammar, the extension of those rules can be found. In other words, while the learner attempts to find general rules, the evaluation stage can make use of phonological rules that are fully explicit. Every general rule can be expanded to the full set of particular rules, in particular contexts, that it includes. In order to make this transformation, a general rule can be expanded to the full set of fully-specified rules that is it might possibly represent, given the segments that exist in the language. For example, suppose that a grammar contains the following phonological rule:

$$\begin{bmatrix} +\text{vocalic} \\ -\text{consonantal} \end{bmatrix} \rightarrow [-\text{voice}]/\_\_\# \ .$$

This would be expanded in the appropriate way, given the segments that appear in

the language in question—perhaps:

$$\text{p} \rightarrow \text{p} \; / \; \_\_ \; \#$$

$$\text{b} \rightarrow \text{p} \; / \; \_\_ \; \#$$

$$\text{t} \rightarrow \text{t} \; / \; \_\_ \; \#$$

$$\text{d} \rightarrow \text{t} \; / \; \_\_ \; \#$$

$$\text{k} \rightarrow \text{k} \; / \; \_\_ \; \#$$

$$\text{g} \rightarrow \text{k} \; / \; \_\_ \; \#$$

$$\text{f} \rightarrow \text{f} \; / \; \_\_ \; \#$$

$$\text{v} \rightarrow \text{f} \; / \; \_\_ \; \#$$

$$\text{s} \rightarrow \text{s} \; / \; \_\_ \; \#$$

$$\text{z} \rightarrow \text{s} \; / \; \_\_ \; \#$$

$$\text{x} \rightarrow \text{x} \; / \; \_\_ \; \#$$

$$\text{ɣ} \rightarrow \text{x} \; / \; \_\_ \; \#$$

It is then possible to compare each fully-specified rule in the learners grammar against each of the fully-specified rule in the human linguists' grammar—one can even obtain counts for true positives, false positives, and false negatives on a rule-by-rule basis. Although this method provides a convenient way to compare the alikeness of to phonological grammars, it is not employed here: in the end, the learner does indeed identify phonological rules in the case studies presented here, but the sets of phonological rules that it posits are not so vast as to resist simple and straightforward qualitative evaluation.

Finally, consider the standards used to evaluate the hypotheses about specific roots and rules that are hypothesized for a particular word form. It is not practical to compare rules against rules, since the comparison of two rules requires some way of

evaluating their morphosyntactic features, the depth at which they apply, as well as the strings that they introduce into the word form. In the end, it does not make sense to try to compare two rules against one another, when one really wants to compare two morphological analyses against one another. Morphological rules certainly form part of that analysis, but evaluating rules against rules is not the most natural way to make the comparison. The clustering analysis mentioned above goes part way to providing a metric that gives a sense of the alikeness of two morphological grammars. This clustering analysis, however, misses a great deal of the analysis when it comes to determining whether the analysis, for a particular word, is anything alike in two separate grammars. For this reason, a metric of the alikeness of roots' URs is introduced.

In order to measure the alikeness of a hypothesis about the form of a root under the linguists' grammar and the learner's grammar, an alignment is formed between the UR of the root as it is posited by the learner, and the UR of the root as it is posited by the linguist. At this point it is possible to describe the alikeness of the two strings in terms of precision, recall, and f-score: one can simply walk through the aligned strings, and for every segment that agrees in the two strings, a true positive is counted. For every segment that is present in the learner's grammar, but not the linguists' grammar, a false positive is counted, and for every segment that is present in the linguists' grammar but not the learner's grammar, a false negative is counted. Notice that for these purposes, a substitution edit is entirely equivalent to an insertion and a deletion—this is significant, because it means that any of the cheapest alignments returned by a Levenshtein edit algorithm will be equivalent in terms of their contribution to the precision, recall, and f-score. This fact in particular makes it easy to calculate the alikeness of two hypotheses about URs in a language. In the summary information about each experiment that is presented below, this measure for the alikeness of hyptheses about URs is presented along with the measure for the

alikeness of the clustering in the two grammars.

In the next three sections, the results from three studies conducted on Classical Arabic, Classical Latin, and Modern Turkish are presented. The evaluation metrics described here are presented for each experiment as a summary of the results from each experiment, but more information is also provided in the form of specific inflection classes, as well as morphological and phonological rules.

4.2 Data selection

A brief word is also in order about the way in which the data was selected for these experiments. For all three languages in question, the corpora of tagged morphological data is larger than what can be processed by the learner within a reasonable period of time. For example, the corpus of tagged works in Classical Latin numbers in the millions of words. While the top-level search procedure described in this work runs in polynomial time with respect to the number of lemmas given as input, as described in section 3.2, such a corpus is far too large to run in full, because there is considerable processing associated with each hypothesized merger. Therefore, some decisions must be made about just what data ought to be used for experimentation, because the full set of available data is too much to digest.

The first cut in the data is by part of speech. The learner will only merge inflection classes that belong to the same part of speech. Therefore, each experiment can be restricted to lemmas of a particular part of speech: although it is possible that there are phonological generalizations that might be missed by looking only at a particular subset of a language, the fact is that such a restriction allows each experiment to focus on the morphological facts that concern a particular verbal or nominal system. The picture that can be obtained of a particular language's verbal or nominal system is more complete than would be obtained if the available computational resources were split across several parts of speech.

The procedure for data selection also makes use of the ranked frequency of each lemma in languages. Notice that, according to Zipf's law, the frequency of a word of frequency rank k in a corpus of N words will have its frequency given by the following formula, where s is determined empirically on a text-by-text basis:

$$\text{frequency} = \frac{k^{-s}}{\sum_{n=1}^{N} n^{-s}}.$$

For a more complete outline of the facts surrounding this formula, see Zipf [67], [68] and Li [43]. The significance for the present work is this, however: for typical values of s observed in corpora of human language data, the distribution of a word of rank k usually works out to roughly $\frac{1}{k}$. In order to select a rich set of data for the learner to work with, one does not want to select data that is too "thin"—that is, a set of data in which there is only one or two forms available for each lemma is likely to lead to a data set that is seriously underdetermined in terms of the inflection classes to which each lemma can be assigned, because with so few forms per lemma, many lemmas are ambiguous in terms of their assignment. At the same time, one cannot use the entire set of available tagged data, since the experiment must run within a reasonable period of time. The solution, therefore, is to make use of the facts about Zipf's law in order to construct a data set that represents a reasonable selection of word forms compactly. For each set of tagged data, the available surface forms are assembled into sets based on the lemma to which each one belongs. These lemmas are then ranked by the frequency with which they appear in the full corpus. One can then select a given number of lemmas from this list—and if one selects from the top of the list, one is likely to obtain forms for which there are a very large number of forms available, and if one selects from farther down the list, one is likely to obtain forms for which several forms are available for each lemma. Finally, the long, thin tail of the distribution contains a very large number of lemmas for which every lemma is associated with at most

one or two distinct surface forms; these lemmas are avoided for these experiments, since they are unlikely to be forms that can be assigned unambiguously to a given inflection class, even by a perfect linguists' grammar. For some experiments, words near the top of the ranked list are used, and for other experiments, words slightly farther down the list are used.

With the facts about the experimental design and its evaluation in place, it is possible turn now to the particular languages of Classical Arabic, Classical Latin, and Modern Turkish.

4.3 Classical Arabic

Classical Arabic is, essentially by definition, the language of the Quran and related documents from the same historical context; these documents were standardized by about the year 700. At the highest level, it is numbered among the Afro-Asiatic languages, which also include the Berber, Chadic, Cushitic, and Egyptian languages; more narrowly, it belongs to the Semitic family, which includes all forms of Modern Arabic, Amharic, Modern Hebrew, and Maltese, as well as number of languages that are no longer spoken—among them, Akkadian, Phoenician, and Akkadian (see Crystal [16] and Paul [51]). The verbal morphology of the Semitic languages, and the Arabic languages in particular, have stood as an intriguing puzzle to linguists for centuries, in large part because of their non-concatenative nature—see Kiparsky [40], McCarthy [46], [47], [48], and the works cited therein for a taste of the formal linguistics literature dealing with the problems posed by the morphological and phonological systems of Arabic. For more facts on the inflectional morphology of Arabic, see Abu-Chacra [1] and Wightwick and Gaafar [65]. Because it forms such a convenient a well-studied corpus, the tagged Quran itself, obtained from the Quranic Arabic Corpus at the University of Leeds, was used as the corpus for this portion of the study.

4.3.1 Overview of the verbal system of Classical Arabic

Regular verbs in Classical Arabic can be assigned to inflectional categories based on the consonants which can be found in their stems. Recall the discussion of Albright [3] and Albright and Hayes [4] in section 1.3.4: it is certainly the case that humans have knowledge about the phonological regularities that sometimes govern the assignment of words to inflection classes. In a world in which the assignment of words to inflection classes is sometimes arbitrary and sometimes governed by phonological principles, the present system makes the explicit choice to treat all such assignments as arbitrary. This means that the learner may well be capable of learning the same categories that a human linguist identifies, but it cannot truly capture this aspect of a human learner's knowledge.

The key fact about the verbal system of Classical Arabic is that its system of conjugation is essentially regular, with the exception of several irregular verbs, and four classes of verbs that fall outside of the standard conjugation class. (See Abu-Chacra [1] and Wightwick and Gaafar [65] for two descriptions of the verbal system in Arabic which serve as the basis for the description of Arabic given here.) Knowing whether or not any of the glides ([j] and [w]) or the glottal stop ([ʔ]) appear in the root of the verb allow one to determine whether it belongs to the regular class or to one of the four irregular classes. When one of these so-called "weak" consonants appears as the first consonant of a triconsonantal root, the verb belongs to the first irregular class; the second irregular class captures verbs in which a weak consonant appears in the second position of its triconsonantal root; the third class encompasses those verbs in which a weak consonant appears in the third position of its root. The fourth class is made up of verbs in which a weak consonant appears in the first and last positions of the triconsonantal root. As mentioned above, this is not a fact that can be captured in the kind of grammars that are being described in the kind of grammatical formalism employed by the present learner. The resulting inflection

classes, however, can be described without difficulty, even if the assignment of words to inflection classes must be treated as an arbitrary fact rather than something that is predictable.

The training corpus was first obtained in the Buckwalter Arabic transliteration system. The symbols of this transliteration system were converted to their phonological equivalents, which were represented in the following way for the purposes of the search for phonological rules:

	vow	cons	oral	ant	lab	cor	dist	dors	high	back	low	son	cont	nas	lat	d.rel	phar	voiced
b	−	+	+	−	+	−	−	−	−	−	−	−	−	−	−	−	−	+
f	−	+	+	−	+	−	−	−	−	−	−	−	+	−	−	−	−	−
t	−	+	+	+	−	+	−	−	−	−	−	−	−	−	−	−	−	−
tˤ	−	+	+	+	−	+	−	−	−	−	−	−	−	−	−	−	+	−
θ	−	+	+	+	−	+	+	−	−	−	−	−	+	−	−	−	−	−
d	−	+	+	+	−	+	−	−	−	−	−	−	−	−	−	−	−	+
dˤ	−	+	+	+	−	+	−	−	−	−	−	−	−	−	−	−	+	+
ð	−	+	+	+	−	+	+	−	−	−	−	−	+	−	−	−	−	+
s	−	+	+	+	−	+	−	−	−	−	−	−	+	−	−	−	−	−
sˤ	−	+	+	+	−	+	−	−	−	−	−	−	+	−	−	−	+	−
z	−	+	+	+	−	+	−	−	−	−	−	−	+	−	−	−	−	+
ʃ	−	+	+	−	−	+	−	−	−	−	−	−	+	−	−	−	−	−
ʤ	−	+	+	−	−	+	−	−	−	−	−	−	−	−	−	+	−	+
ʒˤ	−	+	+	−	−	+	+	−	−	−	−	−	+	−	−	−	+	+
k	−	+	+	−	−	−	−	+	−	−	−	−	−	−	−	−	−	−
x	−	+	+	−	−	−	−	+	−	−	−	−	+	−	−	−	−	−
ɣ	−	+	+	−	−	−	−	+	−	−	−	−	+	−	−	−	−	+
q	−	+	+	−	−	−	−	+	−	+	−	−	−	−	−	−	−	−
l	−	+	+	+	−	+	−	−	−	−	−	+	+	−	+	−	−	+
ʁ	−	+	+	−	−	−	−	+	−	+	−	+	+	−	−	−	−	+
m	−	+	+	−	+	−	−	−	−	−	−	−	−	+	−	−	−	+
n	−	+	+	+	+	−	−	−	−	−	−	−	−	+	−	−	−	+
j	+	+	+	−	−	−	−	+	+	−	−	+	+	−	−	−	−	+
w	+	+	+	−	+	−	−	+	−	+	−	+	+	−	−	−	−	+
i	+	−	+	−	−	−	−	+	+	−	−	+	+	−	−	−	−	+
a	+	−	+	−	−	−	−	+	−	−	+	+	+	−	−	−	−	+
u	+	−	+	−	+	−	−	+	−	+	−	+	+	−	−	−	−	+
ħ	−	+	−	−	−	−	−	−	−	−	−	−	+	−	−	−	+	−
ʕ	−	+	−	−	−	−	−	−	−	−	−	−	+	−	−	−	+	+
ʔ	−	+	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−
h	−	+	−	−	−	−	−	−	−	−	−	−	+	−	−	−	−	−

Although the facts about triconsonantal roots and verb classes is not something that the present system can capture as a generalization about the phonological forms of words, one might note that the glides [w] and [j] do not form a natural class with the glottal stop [ʔ]. Capturing the special behavior of [w], [j], and [ʔ] while excluding the true vowels and the other non-oral consonants (such as [h],[ħ], and [ʕ]) is a challenge for most theories of phonology.

4.3.2 Experiments and results

Word forms were obtained from the set of inflected verbs that appear in the Quran. This corpus was obtained from the Quranic Arabic Corpus at the University of Leeds,

and it includes, for each word, information about its morphosyntactic features, the identity of its root, and a transliteration in the Buckwalter system. These word forms were grouped by lemma and sorted by the number of word forms available for each lemma. Different portions of this sorted list of verb forms were used for the three experiments conducted on Arabic. The set of 51 lemmas of rank 250 through 300 were used for two experiments, one in which only the merge from scratch procedure was deployed, and the other in which the merge from scratch procedure and the merge and find phonological procedure were both used. The set of lemmas from rank 501 through 1499 were used for a final experiment, in which the merge from scratch procedure was the only procedure used. The results are presented below.

Arabic 51 lemmas with phonological rules:

Overview:	Distinct surface forms:	755
	Lemmas:	51
	ICs:	50
Cluster analysis:	true positives:	1
	false positives:	0
	false negatives:	1274
	precision:	1
	recall:	0.000784313725490196
	f-score:	0.00156739811912226
URs:	true positives:	86
	false positives:	15
	false negatives:	88
	precision:	0.851485148514851
	recall:	0.494252873563218
	f-score:	0.625454545454545
phonological rules:	(none)	

Even though the set of inflection classes in this output grammar is relatively small, the set of output inflection classes given below has been abbreviated. See figure 4.1 for a graphical representation of the distribution of lemmas to inflection classes. Within each inflection class, items listed as lemmas are just that: the form of the word as it would appear as a dictionary key word, or a gloss of it, which in either case can serve as a unique identifier. The UR is the underlying representation that the learner has posited for the corresponding lemma. Each inflection class is also associated with a set of morphological rules. These sets of rules can be fairly involved, but see the discussion that follows for some examples.

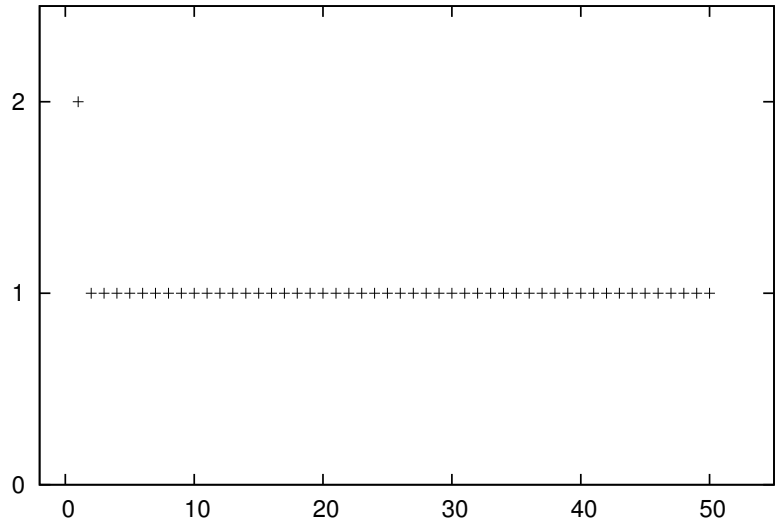

Figure 4.1: Lemmas vs. inflection class by rank, Arabic (small data set).

Output inflection classes:

inflection class 1:

lemma: ʔawʁai, UR: uuʁ

lemma: ʔafʁaħ, UR: afʁaħ

inflection class 2:

lemma: tawakal, UR: tawak

inflection class 3:

lemma: taʕalam, UR: t

inflection class 4:

lemma: tabajan, UR: baj

inflection class 5:

lemma: ʔasʔal, UR: asʔal

inflection class 6:

lemma: ɣaʃiia, UR: a

inflection class 7:

lemma: ʔaʒˤhaʁ, UR: ʒˤh

inflection class 8:

lemma: ʔaftan, UR: ft

inflection class 9:

lemma: ʔawkal, UR: tawakal

inflection class 10:

lemma: sˤadiia, UR: asˤ

inflection class 11:

lemma: ʔaħsˤan, UR: na

inflection class 12:

lemma: ʔaɣʁaq, UR: ɣʁ

The remaining inflection classes are not shown.

In this experiment, the learner performs very poorly. This is true both in comparison with the linguists' grammar, in which the entire set of verbs is captured in five inflection classes, but also in comparison with other experiments performed with this very same learner, on similar tagged training data. In this particular experiment, the learner is successful only in identifying a single merge between inflection classes, and that this merge is a product of the merge from scratch process, rather than as the product of a merge that is associated with a phonological alternation. In the experiments on Latin and Turkish, a much larger set of merges is discovered, and some of these merges make use of phonological rules. These other experiments also perform much better in terms of the precision and recall associated with their identification of URs. The simple fact is that for this small data set, the learner is trapped in a position that is hardly different at all from the learner's initial state, and this state is hardly a good representation of a human speaker's knowledge of Arabic.

In an experiment related to this experiment on Arabic in which phonological rules are allowed, the same data is tested on the learner with the merge while finding phonological rules procedure turned off. The results are, unsurprisingly, identical:

Arabic 51 lemmas without phonological rules:

Overview:	Distinct surface forms:	755
	Lemmas:	51
	ICs:	50
Cluster analysis:	true positives:	1
	false positives:	0
	false negatives:	1274
	precision:	1
	recall:	0.000784313725490196
	f-score:	0.00156739811912226
URs:	true positives:	86
	false positives:	15
	false negatives:	88
	precision:	0.851485148514851
	recall:	0.494252873563218
	f-score:	0.625454545454545
phonological rules:	(none)	

The output inflection classes for both experiments are identical, and they are not repeated here.

In the final experiment involving Arabic, a set of word forms associated with nearly 1000 lemmas is tested, with the search for phonological rules turned off, in consideration of the very large amount of time that is required for searches for phonological rules, even over a modest set of surface forms. In this experiment, the set of merged inflection classes is much richer than it was in the previous two experiments, although the recall score for inflection classes is still low:

Arabic 999 without p:

Overview:	distinct word forms:	3952
	lemmas:	999
	ICs:	640
Cluster analysis:	true positives:	738
	false positives:	17
	false negatives:	483922
	precision:	0.977483443708609
	recall:	0.00152271695621673
	f-score:	0.00304069713544081
URs:	true positives:	2041
	false positives:	1495
	false negatives:	481
	precision:	0.577205882352941
	recall:	0.809278350515464
	f-score:	0.67381974248927
phonological rules:	(none)	

The performance of this learner is far from stellar from a quantitative point of view. On the one hand, the learner is able to posit several hundred merges between inflection classes, and almost all of these merges are reasonable—notice that there are far more true positives than false positives to be found in the cluster analysis, and this is reflected in a very high precision score for the learner's clustering. At the same time, though, the learner leaves a great number of potential merges on the table: the recall value for the learner's clustering is very low, although it is considerably better than the recall value for the learner's clustering in the previous experiments. The key fact here is that the learner was able to identify a large number of merges correctly,

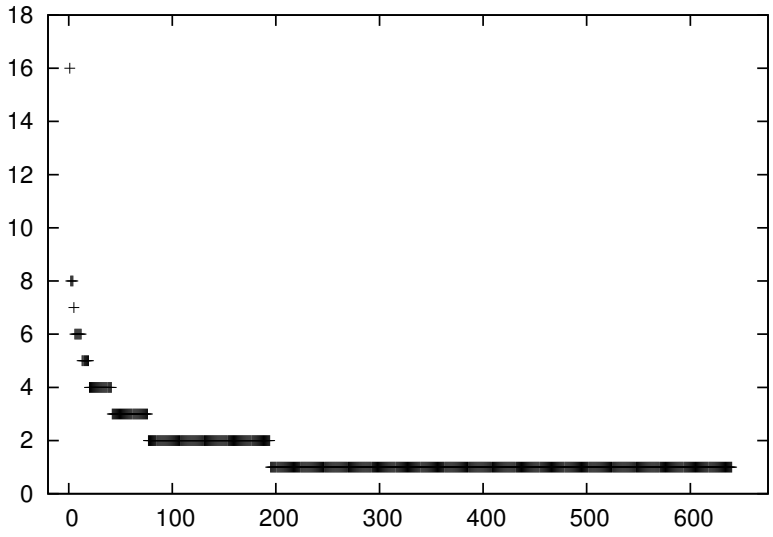

Figure 4.2: Lemmas vs. inflection class by rank, Arabic (large data set).

but many merges that were necessary from the point of view of the linguists' grammar were not implemented. Furthermore, to judge by the precision and recall associated with the URs that the learner has posited, the analyses that the learner provides about the disposition of surface forms into stem material and suffixed material is far from ideal. A sample of the most populous inflection classes from the learner's final output are provided:

Output inflection classes:

inflection class 1:

lemma: sawii, UR: aqtaʁaba

lemma: miθl, UR: astawqada

lemma: ʔiħsaan2, UR: aʕtadai

lemma: daaʁak, UR: adaaʁaka

lemma: miman2, UR: aftaʁai

lemma: aimʁuʔ2, UR: aktasaba

lemma: maluum2, UR: abtaɣai

lemma: ʤabal2, UR: astaqaʁa

lemma: tuqai, UR: altaqai

lemma: ʁiʤiim, UR: astaʁaqa

lemma: jaʕquub, UR: asˤtˤafai

lemma: tabaaʁ, UR: astamaʕa

lemma: ʔalɣam, UR: alɣami

lemma: sˤawt2, UR: amtaħana

lemma: man2, UR: antasˤaʁa

lemma: ʔabħaʁ, UR: ʔbħuʁi

inflection class 2:

lemma: satˤaħ, UR: sutˤħ

lemma: ħasˤiʁa, UR: ħa sˤʁ

lemma: ʕatˤila, UR: ʕutˤl

lemma: satˤaħa, UR: sutˤħ

lemma: ʁabaħ, UR: ʁabħ

lemma: ʁabiħa, UR: ʁabħ

lemma: ʕatˤal, UR: ʕutˤl

lemma: nadˤiʤa, UR: nadˤʤ

inflection class 3:

lemma: ʕaʁaʤ, UR: ʕʁuʤ

lemma: ʔaqnatˤ, UR: qnatˤ

lemma: ʃaqaq, UR: ʃaqaq

lemma: naħb, UR: ntaʒˤiʁ

lemma: ʕaʁiʤa, UR: ʕʁuʤ

lemma: ʔaʕʁaʤ, UR: ʕʁuʤ

lemma: ʔahab, UR: hab

lemma: muqniʕ, UR: ʁtad

inflection class 4:

lemma: ʔaʠnaħ, UR: fʠnaħ

lemma: nasj, UR: wsˤtˤabiʁ

lemma: ʕuʒˤm, UR: wʃtaʕala

lemma: muqtadiʁ, UR: fstamsik

lemma: ʔuðun, UR: wstaɣfiʁ

lemma: ʕuʠaab, UR: wntˤalaqa

lemma: ʔaɣlaʒˤ, UR: wɣluʒˤ

lemma: ʕusʁ, UR: wzduʠiʁa

inflection class 5:

lemma: nafaʃ, UR: nfaʃat

lemma: ʁaħiba, UR: ʁħubat

lemma: taʕamad, UR: tʕamadat

lemma: ʁaħab, UR: ʁħubat

lemma: θaqal, UR: θqulat

lemma: jaqina, UR: jqnut

lemma: kafaf, UR: kfaftu

inflection class 6:

lemma: daawud, UR: astaɣfaʁ

lemma: bahita, UR: buhit

lemma: taʃaabah, UR: taʃaabah

lemma: mawʕiʒˤah, UR: antahai

lemma: ʕazaz, UR: ʕazazna

lemma: ʕaqaʁ, UR: ʕaqaʁ

inflection class 7:

lemma: ʕawað, UR: ʕuuð

lemma: ʔaʁʃad, UR: ʁʃud

lemma: ʔaksab, UR: ksib

lemma: ʔalwai, UR: lw

lemma: ʔaɣdˤai, UR: ɣudˤ

lemma: tanaazaʕ, UR: tanaazaʕ

inflection class 8:

lemma: taʕaawan, UR: taʕaawan

lemma: lawam, UR: luum

lemma: quuah, UR: astaħj

lemma: zalzal, UR: zulzil

lemma: saaʁaʕ, UR: saaʁiʕ

lemma: sˤaabaʁ, UR: sˤaabiʁ

Smaller inflection classes not shown.

Because the inflection classes identified in this experiment are, in general, larger than the inflection classes identified in the first two experiments, the set of inflectional rules that are posited are much more in line with those posited in the linguists' grammar. The reasoning here is that a singleton inflection class that goes unmerged will not have the opportunity to improve the analysis of its roots and rules from the initial state. When inflection classes are merged, however, they are able to make use of the advantages that the merge from scratch procedure has over the procedure that sets up singleton inflection classes in the initial state. In particular, the merge from scratch procedure requires that every lemma be associated with a single, consistent root; when two words with very different roots are merged, a clear distinction between material belonging to the root and material belonging to affixes and infixes can be made. Consider the following set of inflection rules, which apply to the set of forms that fall within inflection class 1.

Sample rules:

inflection class 1:

rule: a l1 ⟨ moodindic person1st voiceact gendernull

prefectivityimperf resultativen posverb numbersg precliticn ⟩ 0

rule: a l1 ⟨ prefectivityperf moodindic voiceact gendermasc
resultativen posverb numbersg precliticn person3rd ⟩ 1

rule: hu r ⟨ moodindic voiceact prefectivityimperf
gendermasc resultativen posverb numbersg precliticn person2nd ⟩ 10

rule: hu r ⟨ moodindic genderfem voiceact prefectivityimperf
resultativen posverb numbersg precliticn person3rd ⟩ 11

When examining these sample inflection rules, note that the first piece of information indicates what string of phonological material is introduced by the rule. The second piece of information tells whether the rule introduces that material at the extreme left edge of the word, as prefix, with the symbol "l", or at the extreme right edge of the word, as a suffix, with the symbol "r". A notation like "r1" indicates that the rule is an infix place one character from the right edge of the root, while "l3" would indicate an infix placed three characters from the left edge of the root. The object enclosed in brackets is the set of morphosyntactic features that the rule marks. Finally, the number at the end of each line indicates the relative depth at which the rule applies. For the purposes of the representation of rules, these sequences of numbers do not need to be continuous: they simply indicate the relative order in which the morphological rules in the inflection class apply.

This sample set of inflection rules is fairly typical of the rules that are discovered for the inflection classes in Arabic: the search for syndromes has not yielded any results, beyond the default assumption that the form may contain thematic material and that each distinct surface form represented in the data may use that full set of morphosyntactic features to condition the application of rules. The fact that the set of morphosyntactic features is so rich as compared with the number of surface forms that are available for each lemma makes it difficult to perform the syndrome search successfully. The inflection class also makes use of infixes as well as prefixes

and suffixes. As one can see when one examines the rules that are discovered for Latin and Turkish, however, this use of prefixation, suffixation, and infixation is by no means unique to Arabic—although it is used more heavily in Arabic, the other languages which are traditionally thought of as strictly suffixing languages also have infixes and prefixes as part of their analyses under the present system.

As already noted, the output is exactly the same for the two smaller experiments, regardless of whether or not the search for phonological rules is allowed. Clearly, in a case where the search for phonological rules cannot discover any useful merges, the output grammar will be identical to an output grammar that is based solely on the merge from scratch procedure: only the merge from scratch procedure will be active. It is also clear that the amount of data is of great significance to the outcome of the search. In fact, the size of the set of training data and its distribution are essentially the only properties that differ across these experiments; the nature of the natural language that underlies the data sets is the same for all experiments. In the first two experiments, the number of lemmas available is very small, although the number of word forms for each lemma is relatively high. As a result of the facts about this particular distribution of forms, there are virtually no merges that can be made in order to increase the probability of the grammar. This is in contrast with the situation in the second experiment, where the number of lemmas is greater, but each lemma is associated with fewer word forms. More merges are found in this case—presumably, this can be attributed to the fact that there is a wider variety of lemmas to be merged into inflection classes, as well as the fact that each lemma is associated with fewer word forms in the training data. With each lemma being associated with fewer word forms, there are fewer forms that must be reconciled with one another when roots and rules are identified. Additionally, with a wider variety of lemmas on hand, it is easier to find sets of lemmas that can be profitably combined into an inflection class together, while in the case of a small training set, such sets of lemmas are less likely

to exist. The key fact here, though, is that it is strictly a fact about the distribution of the training data that gives rise to the dramatic difference in the performance of the learner in the first two experiments, as compared with the last experiment.

4.4 Classical Latin

Classical Latin was the language of members of the wealthy and educated classes during the time of the late Roman Republic and Roman Empire. Genetically, Classical Latin is numbered among the Indo-European languages, and it is closely related to the modern Romance languages—although these languages are in fact derived from Vulgar or Colloquial Latin. Although Vulgar Latin was spoken by a large number of people concurrently with Classical Latin, Classical Latin was recorded and preserved, and a large corpus of Classical Latin is available today. For a more detailed discussion of the history and context of Classical Latin, see Pulgram [53], Vineis [64], and Silvestri [58].

Classical Latin is notable for its highly inflectional morphology, in which a single suffix will mark several distinct morphological feature values. A complete description of the morphology of Latin can be found in Moreland and Fleischer [49] and in Bennet [10]. The corpus used for the present study is the morphologically tagged Latin corpus available as part of the Perseus Digital Library Project, a compendium of tagged classical texts made available by Tufts University.

4.4.1 Overview of the nominal system of Classical Latin

The traditional description of Classical Latin nouns makes use of five distinct inflection classes, or declensions. In the terms of the kinds of grammars described in the present work, however, it is necessary to make use of a much larger set of distinct inflection classes in order to describe the nominal system. The reason for this is that some of the declensions, as they are traditionally defined, actually include several sim-

ilar inflection classes, according to the definition of inflection classes that is used here. See Moreland and Fleischer [49] and Bennet [10] for the facts about Latin that serve as the basis for this discussion of the linguists' grammar of the inflectional system of Classical Latin.

For example, the first and fifth declensions essentially stand up as their own inflection class according to the abbreviatory conventions being employed here. The second declension, however, must be divided into one inflection class for second declension masculine nouns and another for second declension neuter nouns, because gender plays a role in the way that certain morhposyntactic syndromes are realized. Nevertheless, even this division of the second declension into two inflection classes is not, by itself, enough to account for the entire set of nouns that are traditionally assigned to this declension: there are certain second declension nouns whose stems end in r and that do not take on the typical suffix in the nominative singular. Certain other nouns differ very slightly from the typical second declension pattern in terms of the presence or absence of a particular vowel in their suffixes. Although the paradigms for the second declension nouns are very similar at first inspection, the fact is that they cannot be reduced to a single inflection class, according the definition of an inflection class as a set of words that are all marked with the same set of inflectional rules.

The situation with the third and fourth declensions is similar. The fourth declension must be divided into one inflection class for fourth declension masculine nouns, and another for fourth declension neuter nouns; this leaves a handful of irregular nouns that resemble the other fourth declension nouns closely, but not entirely. The disposition of the third declension nouns is particularly complex: this declension must be broken down into separate classes for third declension masculine nouns, third declension feminine nouns, and third declension neuter nouns in addition to separate classes for i-stem masculine, i-stem feminine, and i-stem neuter nouns, as well as

large number of nouns that have irregular forms in the nominative singular. In total, the linguists' grammar used to evaluate the learner's grammar in Classical Latin makes use of fourteen distinct inflection classes.

The morphological system of Latin shows a number of segmental alternations. While these alternations may have some clear phonological basis, most of them appear to be frozen, and codified in the morphology of the language, by the time of Classical Latin, because they appear only in certain declensions. There is, however, unambiguous evidence for an active phonological rule that devoices certain obstruents when they appear immediately to the left of the context [s#]. The rule applies without exception to [g], as seen in the nominative singular form of [reg], "king", when it takes on the nominative singular ending [s]: [regs] → [reks]. The rule applies with more variability to other forms, so the the nominative singular form of [pleb], "plebe", appears as both *plebs* and *pleps* in writing. For this reason, the linguists' grammar includes the rule

$$[g] \rightarrow [k] / \_\_ [s\#] \text{ (in the general context)}$$

but there is no reason to think that the grammar must necessarily contain more rules of devoicing.

The phonological inventory of Latin is significantly less exhuberant than that of Arabic. The experiments involving Latin made use of the following mapping between phones and phonological features. Note that vowel length is not represented:

	vow	cons	lab	cor	dors	high	back	low	son	cont	nas	lat	voiced
i	+	−	−	−	+	+	−	−	+	+	−	−	+
e	+	−	−	−	+	−	−	−	+	+	−	−	+
a	+	−	−	−	+	−	+	+	+	+	−	−	+
o	+	−	+	−	+	−	+	−	+	+	−	−	+
u	+	−	+	−	+	+	+	−	+	+	−	−	+
j	+	+	−	−	+	+	−	−	+	+	−	−	+
w	+	+	+	−	+	+	+	−	+	+	−	−	+
r	−	+	−	+	−	−	−	−	+	+	−	−	+
l	−	+	−	+	−	−	−	−	+	+	−	+	+
p	−	+	+	−	−	−	−	−	−	−	−	−	−
b	−	+	+	−	−	−	−	−	−	−	−	−	+
m	−	+	+	−	−	−	−	−	+	−	+	−	+
f	−	+	+	−	−	−	−	−	−	+	−	−	−
t	−	+	−	+	−	−	−	−	−	−	−	−	−
d	−	+	−	+	−	−	−	−	−	−	−	−	+
n	−	+	−	+	−	−	−	−	+	−	+	−	+
s	−	+	−	+	−	−	−	−	−	+	−	−	−
k	−	+	−	−	+	−	−	−	−	−	−	−	−
g	−	+	−	−	+	−	−	−	−	−	−	−	+
n	−	+	−	−	+	−	−	−	−	−	−	−	+
ŋ	−	+	−	−	+	−	−	−	+	−	+	−	+
h	−	+	−	−	−	−	−	−	−	+	−	−	−

As with the Arabic experiments, the original corpus was obtained in orthographic, rather than phonological, form. In the case of the Latin language, however, it is relatively easy to map between orthographic forms of words and the appropriate phonological form.

4.4.2 Experiments and results

For this work, the set of nouns from all Classical Latin works found in Tufts University's Perseus Project corpus was grouped by lemma, and then sorted according the number of word forms available for each lemma. In the first two experiments, the set of 51 lemmas of rank 50 through 100 were employed. One experiment was conducted in which both merge from scratch and merge and find phonological rules were deployed; in the other experiment, only merge from scratch was deployed. This allows the grammar found without the benefit of phonological rules to be compared

with the grammar found with their use. Another experiment was conducted on a larger data set was conducted using just the merge from scratch form of the search. This data set was constructed by taking the 999 lemmas of rank 501 through 1499.

The training data for Classical Latin was obtained in orthographic form, and was converted into a phonological representation using several simple finite-state mappings (such as orthographic $x \rightarrow$ phonological [ks]). This allows the learner to work on phonological forms, and to search for rules of phonological alternation, rather than for rules of spelling alternation.

The results from these experiments are given below.

Latin 51 lemmas with phonological rules:

Overview:	Distinct surface forms:	1169
	Lemmas:	51
	ICs:	28
Cluster analysis:	true positives:	58
	false positives:	0
	false negatives:	1217
	precision:	1
	recall:	0.0454901960784314
	f-score:	0.0870217554388597
URs:	true positives:	223
	false positives:	6
	false negatives:	99
	precision:	0.973799126637555
	recall:	0.692546583850932
	f-score:	0.809437386569873
phonological rules:	[d] → [s] / __ # (general context)	
	[i] → [e] / [l] __ [s] (general context)	

There are several key facts to notice about this summary. First, the clustering is far more successful with this small set of Latin data than it was with the small set of Arabic data. This aspect of the experiment can be seen quantitatively in terms of the recall and f-score associated with the clustering that the learner gives as output. One can also simply notice that the 51 lemmas in this experiment were reduced to 28 total inflection classes. This is in contrast with the situation in Arabic, where 51 lemmas were reduced to merely 50 total inflection classes.

The results of this experiment are also intriguing because they show the learner

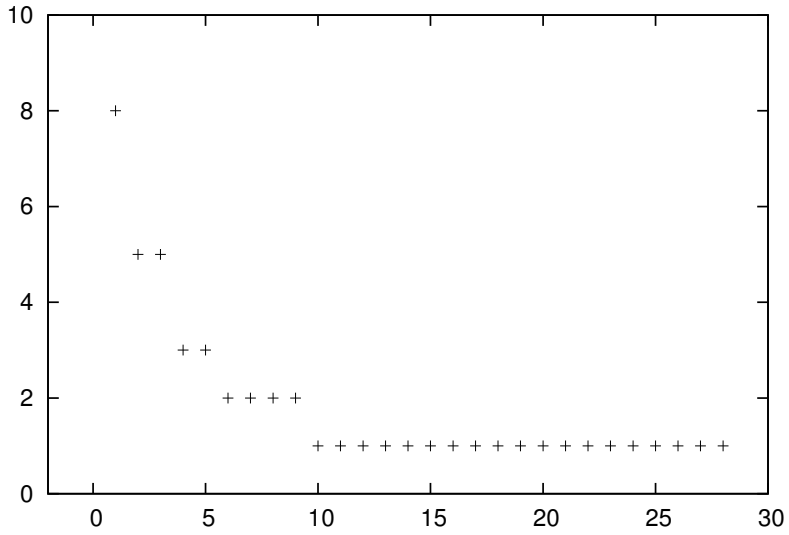

Figure 4.3: Lemmas vs. inflection class by rank, Latin (small data set, phonological rules included).

making use of a phonological rule when merging categories, and because the precision and recall scores associated with the identification of roots' URs is so much better.

The sets of inflection classes in the output grammar of these smaller experiments on Classical Latin are small enough to present in full; they are particularly intersting because of the phonological alternations that are included. In this corpus, the lemmas listed for each inflection class represent the dictionary key words associated with each word, regularized according to the same set of orthography to phonology mappings. Sample sets of inflectional rules are given in the discussion that follows.

Ouput inflection classes:

inflection class 1:

lemma: Skapula, UR: skapul

lemma: athleta, UR: athlet

lemma: inkola, UR: inkol

lemma: parrikida, UR: parrikid

lemma: adwena, UR: adwen

lemma: werna, UR: wern

lemma: transfuga, UR: transfug

lemma: konwiwa, UR: konwiw

inflection class 2:

lemma: diwus, UR: diw

lemma: alwus, UR: alw

lemma: antidotum, UR: antidot

lemma: raphanus, UR: raphan

lemma: balanus, UR: balan

inflection class 3:

lemma: owis, UR: ow

lemma: finis, UR: fin

lemma: kiwis, UR: kiw

lemma: kanalis, UR: kanal

lemma: popularis, UR: popular

inflection class 4:

lemma: serpens, UR: serpen

lemma: interpres, UR: interpre

lemma: sakerdos, UR: sakerdo

inflection class 5:

lemma: heres, UR: hered

lemma: satelles, UR: satelles

lemma: fur, UR: fur

inflection class 6:

lemma: praekeps, UR: pre

lemma: ankeps, UR: n

inflection class 7:

lemma: defensor, UR: defens

lemma: auktor, UR: aukt

inflection class 8:

lemma: parens2, UR: (null)

lemma: parens, UR: (null)

inflection class 9:

lemma: miles, UR: milit

lemma: arbor, UR: arbor

inflection class 10:

lemma: deskendens, UR: deskenden

inflection class 11:

lemma: kontinens, UR: kontinen

inflection class 12:

lemma: konjunks, UR: kon

inflection class 13:

lemma: korteks, UR: kort

inflection class 14:

lemma: kustos, UR: sto

inflection class 15:

lemma: hostis, UR: host

inflection class 16:

lemma: seneks, UR: sen

inflection class 17:

lemma: testis, UR: test

inflection class 18:

lemma: judeks, UR: iud

inflection class 19:

lemma: tigris, UR: tigr

inflection class 20:

lemma: Arabs, UR: arab

inflection class 21:

lemma: testa, UR: test

inflection class 22:

lemma: simia, UR: simi

inflection class 23:

lemma: kanis, UR: kan

inflection class 24:

lemma: kardo, UR: kard

inflection class 25:

lemma: nepos, UR: nepo

inflection class 26:

lemma: homo, UR: hom

inflection class 27:

lemma: luks, UR: luk

inflection class 28:

lemma: mus, UR: mu

Sample rules:

inflection class 1:

rule: ae r ⟨ casenom genderfem numberpl ⟩ 6

rule: ae r ⟨ casevoc genderfem numberpl ⟩ 13

rule: ae r ⟨ casenom gendermasc numberpl ⟩ 20

rule: ae r ⟨ gendermasc casevoc numberpl ⟩ 27

rule: ae r ⟨ casedat genderfem numbersg ⟩ 34

rule: ae r ⟨ genderfem numbersg casegen ⟩ 41

rule: ae r ⟨ gendermasc casedat numbersg ⟩ 48

rule: ae r ⟨ gendermasc numbersg casegen ⟩ 55

rule: am r ⟨ genderfem numbersg caseacc ⟩ 62

rule: am r ⟨ gendermasc numbersg caseacc ⟩ 69

rule: a r ⟨ genderfem numbersg caseabl ⟩ 70

rule: a r ⟨ casenom genderfem numbersg ⟩ 71

rule: a r ⟨ casevoc genderfem numbersg ⟩ 72

rule: a r ⟨ gendermasc numbersg caseabl ⟩ 73

rule: a r ⟨ casenom gendermasc numbersg ⟩ 74

rule: a r ⟨ gendermasc casevoc numbersg ⟩ 75

rule: as r ⟨ genderfem numberpl caseacc ⟩ 76

rule: is r ⟨ genderfem numberpl caseabl ⟩ 77

rule: is r ⟨ casedat genderfem numberpl ⟩ 78

rule: is r ⟨ gendermasc numberpl caseabl ⟩ 79

rule: is r ⟨ gendermasc casedat numberpl ⟩ 80

rule: as r ⟨ gendermasc numberpl caseacc ⟩ 81

rule: arum r ⟨ genderfem numberpl casegen ⟩ 82

rule: arum r ⟨ gendermasc numberpl casegen ⟩ 83

rule: arum r ⟨ gendermasc numberpl casegen ⟩ 85

rule: a r ⟨ numberpl caseacc genderneut ⟩ 86

rule: a r ⟨ casenom numberpl genderneut ⟩ 87

rule: a r ⟨ casevoc numberpl genderneut ⟩ 88

rule: is r ⟨ numberpl genderneut caseabl ⟩ 90

rule: is r ⟨ casedat numberpl genderneut ⟩ 91

This inflection class corresponds roughly with the first declension in traditional grammars, but it admits some interlopers who would have been excluded if they had

come furnished with a more complete set of surface forms—for certain lemmas, the forms that would give them away as necessarily belonging to some other inflection class simply were not part of the training data, and for this reason they are allowed into the inflection class.

A word or two is clearly in order about the phonological rules, and the impact that they have on the analysis that the learner produces. The first phonological rule, which maps [d] → [s] when the segment appears in word final position, is being deployed to bring *heres* into inflection class 5:

lemma: heres, UR: hered

lemma: satelles, UR: satelles

lemma: fur, UR: fur

The word *heres* is essentially irregular, in the sense that it appears as *heres* in the nominative singular, but as *heredis*, *heredi*, etc., in other morphosyntactic configurations. This fact about [d] → [s] in word final position is not really a fact about the phonology of Classical Latin. Rather, it is a fact about the morphology of the language, but the learner finds that it is in fact profitable to take this regularity in the training data as a fact of the phonology. The same is true of the other phonological rule, which maps [i] → [e] in certain contexts. The learner deploys this phonological rule in order to collapse *arbor* and *miles* into the same inflection class; the fact that the nominative singular form of *miles* is *miles* and not *milis* is the fact—clearly morphological, rather than phonological, from the point of view of a human linguist—that the rule is being employed to capture.

Next, consider the results obtained when a leaner making use only of the merge from scratch procedure is trained on this same small set of data.

Latin 51 lemmas without phonological rules:

Overview:	Distinct surface forms:	1169
	Lemmas:	51
	ICs:	29
Cluster analysis:	true positives:	57
	false positives:	0
	false negatives:	1218
	precision:	1
	recall:	0.0447058823529412
	f-score:	0.0855855855855856
URs:	true positives:	218
	false positives:	3
	false negatives:	104
	precision:	0.986425339366516
	recall:	0.677018633540373
	f-score:	0.802946593001842
phonological rules:	(none)	

The learner fares slightly poorer in several regards: the output grammar is not quite as successful as the previous grammar at either the clustering task or at identifying roots' URs appropriately. That being said, the two grammars are very close to one another in terms of the analyses that they posit. For comparison, the set of inflection classes is given below:

Ouput inflection classes:

inflection class 1:

lemma: Skapula, UR: skapul

lemma: athleta, UR: athlet

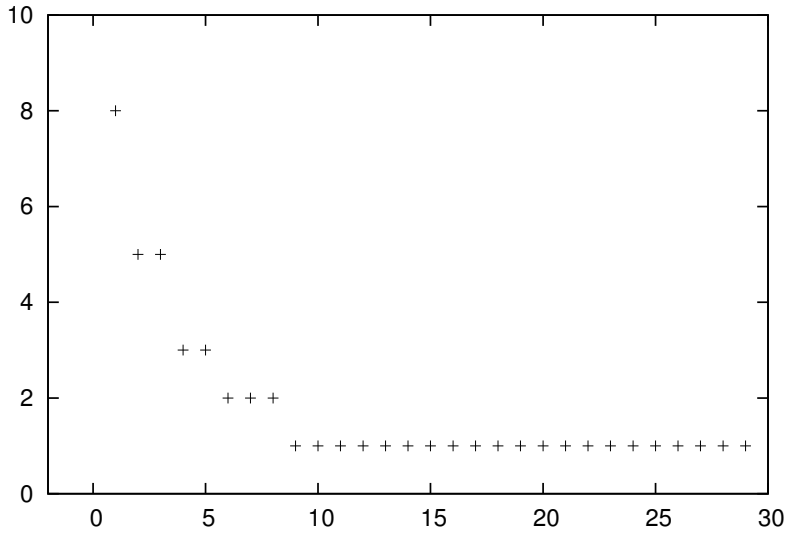

Figure 4.4: Lemmas vs. inflection class by rank, Latin (small data set, no phonological rules).

lemma: inkola, UR: inkol

lemma: parrikida, UR: parrikid

lemma: adwena, UR: adwen

lemma: transfuga, UR: transfug

lemma: werna, UR: wern

lemma: konwiwa, UR: konwiw

inflection class 2:

lemma: alwus, UR: alw

lemma: diwus, UR: diw

lemma: antidotum, UR: antidot

lemma: raphanus, UR: raphan

lemma: balanus, UR: balan

inflection class 3:

lemma: owis, UR: ow

lemma: kanalis, UR: kanal

lemma: finis, UR: fin

lemma: kiwis, UR: kiw

lemma: popularis, UR: popular

inflection class 4:

lemma: serpens, UR: serpen

lemma: interpres, UR: interpre

lemma: sakerdos, UR: sakerdo

inflection class 5:

lemma: kontinens, UR: kontin

lemma: deskendens, UR: deskend

inflection class 6:

lemma: fur, UR: fu

lemma: defensor, UR: defenso

lemma: auktor, UR: aukto

inflection class 7:

lemma: praekeps, UR: pre

lemma: ankeps, UR: n

inflection class 8:

lemma: parens2, UR: (null)

lemma: parens, UR: (null)

inflection class 9:

lemma: konjunks, UR: kon

inflection class 10:

lemma: satelles, UR: satell

inflection class 11:

lemma: korteks, UR: kort

inflection class 12:

lemma: kustos, UR: sto

inflection class 13:

lemma: hostis, UR: host

inflection class 14:

lemma: seneks, UR: sen

inflection class 15:

lemma: testis, UR: test

inflection class 16:

lemma: judeks, UR: iud

inflection class 17:

lemma: tigris, UR: tigr

inflection class 18:

lemma: Arabs, UR: arab

inflection class 19:

lemma: testa, UR: test

inflection class 20:

lemma: simia, UR: simi

inflection class 21:

lemma: kanis, UR: kan

inflection class 22:

lemma: miles, UR: mil

inflection class 23:

lemma: kardo, UR: kard

inflection class 24:

lemma: heres, UR: here

inflection class 25:

lemma: arbor, UR: arbor

inflection class 26:

lemma: nepos, UR: nepo

inflection class 27:

lemma: homo, UR: hom

inflection class 28:

lemma: luks, UR: luk

inflection class 29:

lemma: mus, UR: mu

Because the phonological rules described above cannot exist in this system, the lemmas for *heres* and *miles* cannot be merged with other lemmas in the way that they were in the previous experiment. Notice, though, that this grammar is not simply two merges short of being identical to the grammar that makes use of phonological rules. Instead, whether or not phonological rules can be discovered early on can perturb the grammar in such a way that different inflection classes are discovered. In other words, the difference between the two clusters is not merely the difference between a particular merge being made or not; rather, the difference is between two different paths through the search space, with the final arrangement of clusters differing slightly because of these paths.

In the final experiment on Classical Latin, a large set of data is run without phonological rules being employed. This experiment, and the experiments that follow it on Turkish, are especially illuminating in terms of what they reveal about the attributes of the learner's objective function and search procedures.

Latin 999 lemmas without phonological rules:

Overview:	Distinct surface forms:	11879
	Lemmas:	999
	ICs:	923
Cluster analysis:	true positives:	366
	false positives:	0
	false negatives:	4785
	precision:	1
	recall:	0.0710541642399534
	f-score:	0.132680804785209
URs:	true positives:	4987
	false positives:	107
	false negatives:	1434
	precision:	0.978994895956027
	recall:	0.776670300576234
	f-score:	0.866174554928354
phonological rules:	(none)	

This grammar does quite well in terms of the precision and recall associated with its hypotheses about the URs associated with roots. The recall and f-score associated with its clustering performance are also better than those obtained with smaller data sets, by a factor of about 1.5, in each case. Although large, sparse data sets are problematic from many points of view—including the syndrome search, in particular—they apparently tend to perform better than smaller data sets in terms of the clusters that can be formed over them. This is certainly the case for both Latin and Arabic. Consider some of these clusters from this larger data set:

Ouput inflection classes:

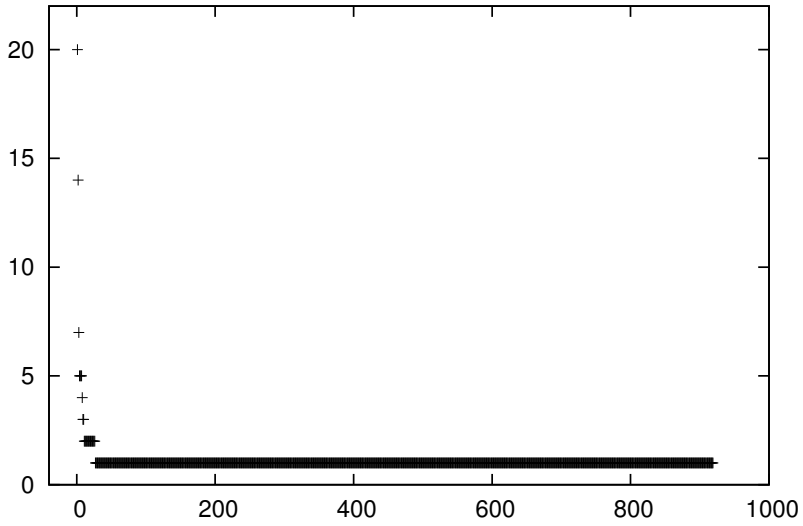

Figure 4.5: Lemmas vs. inflection class by rank, Latin (large data set).

inflection class 1:

lemma: deliberatio, UR: deliberat

lemma: purgatio, UR: purgat

lemma: kogitatio, UR: kogitat

lemma: sukkessio, UR: sukkess

lemma: kontio, UR: kont

lemma: aktio, UR: akt

lemma: postulatio, UR: postulat

lemma: kuaestio, UR: kuaest

lemma: superstitio, UR: superstit

lemma: dominatio, UR: dominat

lemma: eruptio, UR: erupt

lemma: dimikatio, UR: dimikat

lemma: kontignatio, UR: kontignat

lemma: ambulatio, UR: ambulat

lemma: kognitio, UR: kognit

lemma: populatio2, UR: populat

lemma: rogatio, UR: rogat

lemma: interrogatio, UR: interrogat

lemma: seditio, UR: sedit

lemma: dispositio, UR: disposit

inflection class 2:

lemma: gratia, UR: grati

lemma: fabula, UR: fabul

lemma: marita, UR: marit

lemma: diskiplina, UR: diskiplin

lemma: nata, UR: nat

lemma: plaga2, UR: plag

lemma: plaga3, UR: plag

lemma: Graeke, UR: graek

lemma: miseria, UR: miseri

lemma: praeda, UR: praed

lemma: nupta, UR: nupt

lemma: porta, UR: port

lemma: palaestra, UR: palaestr

lemma: bestia, UR: besti

inflection class 3:

lemma: flamen2, UR: fla

lemma: kakumen, UR: kaku

lemma: legumen, UR: legu

lemma: flumen, UR: flu

lemma: medikamen, UR: medika

lemma: diskrimen, UR: diskri

lemma: kertamen, UR: kerta

inflection class 4:

lemma: kompos, UR: kompo

lemma: videns, UR: viden

lemma: aduleskens, UR: adulesken

lemma: Maekenas, UR: maekena

lemma: innokens, UR: innoken

inflection class 5:

lemma: faktio, UR: fakt

lemma: kuratio, UR: kurat

lemma: oratio, UR: orat

lemma: ratio, UR: rat

lemma: opinio, UR: opin

inflection class 6:

lemma: serta, UR: sert

lemma: spuma, UR: spum

lemma: makula, UR: makul

lemma: statua, UR: statu

lemma: physika, UR: physik

inflection class 7:

lemma: munitio, UR: unit

lemma: mansio, UR: ans

lemma: modulatio, UR: odulat

lemma: missio, UR: iss

inflection class 8:

lemma: vimen, UR: vi

lemma: agmen, UR: ag

lemma: semen, UR: se

lemma: lumen, UR: lu

lemma: limen, UR: li

inflection class 9:

lemma: hirundo, UR: hirun

lemma: testudo, UR: testu

lemma: solitudo, UR: solitu

inflection class 10:

lemma: leno2, UR: len

lemma: sermo, UR: serm

lemma: akuilo, UR: akuil

inflection class 11:

lemma: nobilitas, UR: nobil

lemma: dignitas, UR: dign

inflection class 12:

lemma: impense, UR: (null)

lemma: impensa, UR: (null)

Remaining inflection classes are not shown. Consider inflection class 7 in particular:

lemma: munitio, UR: unit

lemma: mansio, UR: ans

lemma: modulatio, UR: odulat

lemma: missio, UR: iss

This inflection class is associated with an unremarkable set of suffixes that mark each word in the appropriate way. Notice, however, that these are all words that start with m, and that in the final analysis this m has been factored out as a thematic prefix:

rule: m l ⟨ ⟩ 0

rule: io r ⟨ casenom genderfem numbersg ⟩ 1

rule: io r ⟨ genderfem numbersg casevoc ⟩ 2

rule: ione r ⟨ genderfem numbersg caseabl ⟩ 3

rule: ionem r ⟨ genderfem numbersg caseacc ⟩ 4

etc.

The key thing to realize about this analysis is that both the search function and the objective function are deliberately designed to allow for both prefixes and suffixes, in order to make their use as general as possbile among the languages of the word. Note, however, that this can sometimes have unintended consequences: in this case, a thematic prefix has been identified. Why has this thematic prefix been identified in this way? It is possible to explain the identification of this prefix from two different points of view: that of the objective function, and that of the search function.

From the point of view of the search function, the merge from scratch procedure cannot help but identify a spurious prefix like *m* in this context—it is simply a property of the merge from scratch procedure that it allows for thematic prefixes and suffixes when it conducts a search for inflectional rules; another property of the search, adopted in order to make it run efficiently and to avoid certain pitfalls, is that it is greedy, in the sense that it always takes the longest available string as the exponent of the feature syndrome under consideration. In the case of this instance of merge from scratch, the left-most segment *m* was identified as a prefix on the basis of these properties of that algorithm.

This fact about the merge from scratch procedure explains why the merge function returned an inflection class in which *m* was factored out in this way. But why does the objective function allow this inflection class to be created? The answer is simple: even if factoring the *m* out as a prefix is not optimal for the probability of this particular inflection class, there was still some stage of the search at which adopting

this inflection class was superior to the others that were available at the time. In many cases, these kinds of spurious analyses can eventually be corrected, as would happen if a word that does not begin with m were to be introduced into this inflection class. In this case, however, the spurious prefix remains even in the output grammar.

Putting this another way, the the system presented here does a reasonably good job of capturing the facts about inflectional morphology of Latin, as intended. After all, the precision and recall associated with the URs that it identifies are quite strong. There are times, though, when it captures facts about the Latin in a way that is unintended—this can be attributed, first and foremost, to the manner in which the merge from scratch procedure is conducted, but for details on this point see section 5.7.

4.5 Modern Turkish

Modern Turkish is the language of the Turkish Republic and of ethnic Turks worldwide; it is the most widely spoken of the modern Turkish languages. It is found at the extreme western end of the geographic distribution of Turkic languages, which today extend from Asia Minor and parts of Southern and Eastern Europe in the west, across vast areas of Central Asia, to areas which today are western China, Mongolia, and Russian Siberia in the east, and among Turkish communities the world over.

A large portion of the Turkish vocabulary is derived from proto-Turkic, but the language has also acquired a large number of loans from Persian and Arabic. Various reformers, particularly in the early twentieth century, have tried, never with complete success, to replace these loan words with native Turkic forms. See Landau [41] and Lewis [42] for information on the origins of Modern Turkish. This fact about vocabulary items that originated as loan words is significant from the point of view of the phonological generalizations that apply within the language: the language displays a strong tendency towards progressive vowel harmony. In both native and foreign

roots, the suffixes that apply to a root acquire certain vowel features that are found on the root. Roots of native Turkic origin are, generally, vowel harmonic, whereas those roots of Arabic, Persian, or other origin do not necessarily display root-internal harmony, however. This means that vowel harmony applies virtually without exception in the suffixes, but it is only a surface-true fact about the surface forms found in the language when one considers words of Turkic origin—the generalizations about vowel harmony are most certainly not respected within roots of foreign origin.

Modern Turkish is known for its extensive use of agglutination to mark both inflectional and derivational morphology, and vowel harmony. For a detailed description of Turkish morphology, see Göksel and Kerslake [23] and Thomas and Itzkowitz [62]. Because Turkish is used so extensively in the modern world, vast amounts of untagged Turkish can be scraped from sources on the internet. In the present study, a tagged corpus was obtained by automatically obtaining surface forms from the Turkish language section of Wikipedia (WikiMedia's Türkçe Vikipedi project), parsing the forms with a hand-built parser of the author's design, and checking the resulting roots against those that can be found in the Turkish language section of WikiMedia's Wiktionary project (the Türkçe Vikisözlük project).

4.5.1 Overview of the nominal system of Modern Turkish

The description of Turkish in chapter 2 has already touched on the way in which the nominal system of Modern Turkish can be described in terms of the kinds of grammars being treated here. To review, the nominal system of Turkish is extremely regular: with the exception of a tiny handfull of irregular nouns, all Turkish nouns belong to a single inflection class in which number, possession, and case are all marked with agglutinative suffixes. See Göksel and Kerslake [23] and Thomas and Itzkowitz [62] for two very thorough treatments of the inflectional morphology of Modern Turkish, which serve as the basis for the linguists' grammar described here.

One key thing to recall is that this description of Modern Turkish grammar—in which vowel harmony allows front-harmonic and back-harmonic lemmas to be included in the same inflection class—depends upon certain rules of vowel harmony. These are rules of vowel harmony that the learner would, in one sense, ought to be able to learn. The learner does, after all, search for phonological alternations on the vocalic tier, so something like backness harmony among vowels in a word is the kind of generalization that the learner could detect and codify in a phonological rule.

In reality, though, there is no chance for the learner to discover a rule of vowel harmony in any kind of realistic data. The reason for this is that vowel harmony in Turkish is scarcely a surface-true phenomenon. The language is rife with words which break vowel harmony at some point in their stems, and there are even certain vowel harmony surprises within the suffix system. Recall too that the search for phonological alternations demands that the alternation be entirely surface true: there can be no cases in the training data in which the rule appears to have been given the chance to apply, but passed up that chance. This means that the linguists' grammar has one key advantage over the learner's grammar. While the linguist can recognize exceptions to a phonological rule, and even recognize the fact that the phonological rule applies quite productively in derived environments if not within stems, the machine learner does not have this luxury. The present system has no notion of derived or non-derived environments, and it has no notion of phonological rules that might fail to apply to certain roots. The linguists' grammar can capture the facts about the vowel harmony very easily, but the machine learner cannot.

The orthography of Turkish—like the orthography of Latin—maps closely to the phonological forms. This fact about the orthography of Turkish makes it possible to use written texts as the basis for this kind of experimental work, as long as they can be appropriately tagged, and as long as the phonological mapping for each orthographic form can be obtained. For the purposes of this experiment, the following set of

phonological features were associated with the phonolgical segments of Turkish:

	vow	cons	lab	cor	ant	dors	high	low	back	son	cont	nas	lat	d.rel	vd
i	+	−	−	−	−	+	+	−	−	+	+	−	−	−	+
e	+	−	−	−	−	+	−	−	−	+	+	−	−	−	+
a	+	−	−	−	−	+	−	+	+	+	+	−	−	−	+
o	+	−	+	−	−	+	−	−	+	+	+	−	−	−	+
u	+	−	+	−	−	+	+	−	+	+	+	−	−	−	+
ɯ	+	−	+	−	−	+	+	−	−	+	+	−	−	−	+
ɤ	+	−	+	−	−	+	−	−	−	+	+	−	−	−	+
ɨ	+	−	−	−	−	+	+	−	+	+	+	−	−	−	+
j	+	+	−	−	−	+	+	−	−	+	+	−	−	−	+
w	+	+	+	−	−	+	+	−	+	+	+	−	−	−	+
r	−	+	−	+	+	−	−	−	−	+	+	−	−	−	+
l	−	+	−	+	+	−	−	−	−	+	+	−	+	−	+
p	−	+	+	−	−	−	−	−	−	−	−	−	−	−	−
b	−	+	+	−	−	−	−	−	−	−	−	−	−	−	+
m	−	+	+	−	−	−	−	−	−	+	−	+	−	−	+
f	−	+	+	−	−	−	−	−	−	−	+	−	−	−	−
v	−	+	+	−	−	−	−	−	−	−	+	−	−	−	+
t	−	+	−	+	+	−	−	−	−	−	−	−	−	−	−
d	−	+	−	+	+	−	−	−	−	−	−	−	−	−	+
n	−	+	−	+	+	−	−	−	−	+	−	+	−	−	+
s	−	+	−	+	+	−	−	−	−	−	+	−	−	−	−
z	−	+	−	+	+	−	−	−	−	−	+	−	−	−	+
ʧ	−	+	−	+	−	−	−	−	−	−	+	−	−	+	−
ʤ	−	+	−	+	−	−	−	−	−	−	+	−	−	+	+
ʃ	−	+	−	+	−	−	−	−	−	−	+	−	−	−	−
ʒ	−	+	−	+	−	−	−	−	−	−	+	−	−	−	+
c	−	+	−	−	−	+	−	−	−	−	−	−	−	−	−
ɟ	−	+	−	−	−	+	−	−	−	−	−	−	−	−	+
k	−	+	−	−	−	+	−	−	+	−	−	−	−	−	−
g	−	+	−	−	−	+	−	−	+	−	−	−	−	−	+
ɣ	−	+	−	−	−	+	−	−	+	−	+	−	−	−	+
h	−	+	−	−	−	−	−	−	−	−	−	−	−	−	−

4.5.2 Experiments and results

The training data for Modern Turkish was obtained in a somewhat different manner from the Classic Arabic and the Classical Latin training data. For this language, the author employed a script to crawl through the WikiMedia Foundation's Türkçe Vikipedi project, with an appropriate delay between requests, and to collect the surface forms found in the articles there. A morphological parser implementing the

grammar described in section 4.5.1 was then used to parse each word, and to check to see whether the resulting stem in fact exists as a true Turkish noun in the Wikimedia Fournation's Türkçe Vikisözlük. This procedure was stopped after an appropriate number of lemmas were obtained. For this reason, the set of words used in this experiment are clustered near the beginning of the alphabet.

As with the previous case studies, three experiments were obtained. In the first two experiments, a set of word forms associated with 149 lemmas was fed to the learner under two configurations. In one configuration, the learner made use of both the merge from scratch and the merge while finding phonological rules procedures. In the other configuration, the learner made use of the merge from scratch procedure only. In the third experiment, the learner was trained on a set of word forms associated with 400 lemmas, with only the merge from scratch procedure in place. The results from the smaller experiments are presented first.

Turkish 149 lemmas with phonological rules:

Overview:	Distinct surface forms:	929
	Lemmas:	149
	ICs:	132
Cluster analysis:	true positives:	32
	false positives:	0
	false negatives:	10994
	precision:	1
	recall:	0.00290223109015055
	f-score:	0.00578766503888587
URs:	true positives:	675
	false positives:	1
	false negatives:	29
	precision:	0.998520710059172
	recall:	0.958806818181818
	f-score:	0.978260869565217
phonological rules:	(none)	

These results are particularly interesting because the precision and recall for the identification of roots' URs is so high, even though the recall of the clustering analysis is very low. The reason that the recall of the clustering analysis is very low is quite simple: when the linguits' grammar contains only one inflection class, any failure to merge inflection classes in the learner's grammar will count against the system's recall.

In the case of these experiments on Modern Turkish, it is not reasonable to present the full sets of inflection classes, even for the smaller experiments. Within each inflection class, the lemmas are the forms of each word that would serve as a dictionary

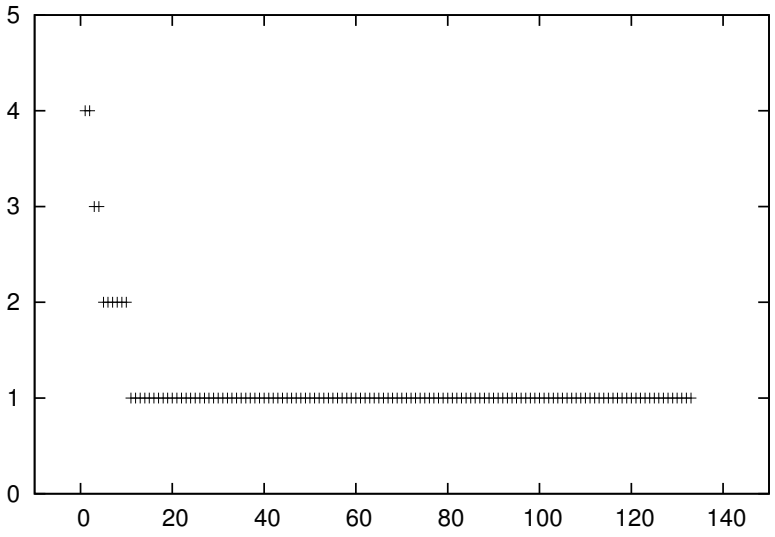

Figure 4.6: Lemmas vs. inflection class by rank, Turkish (small data set).

key word, while the URs are the underlying representations that the learner posits for those lemmas.

Ouput inflection classes:

inflection class 1:

lemma: afetler, UR: fetler

lemma: abid, UR: bid

lemma: aker, UR: ker

lemma: aileler, UR: ileler

inflection class 2:

lemma: adres, UR: dres

lemma: asiret, UR: siret

lemma: ajet, UR: jet

lemma: amel, UR: mel

inflection class 3:

lemma: avusturjalılar, UR: vusturjalılar

lemma: aklınız, UR: klınız

lemma: alaj, UR: laj

inflection class 4:

lemma: anit, UR: nit

lemma: alkal, UR: lkal

lemma: abaz, UR: baz

inflection class 5:

lemma: angl, UR: ngl

lemma: ateʃkes, UR: teʃkes

inflection class 6:

lemma: akinc, UR: kinc

lemma: alkan, UR: lkan

inflection class 7:

lemma: azer, UR: zer

lemma: ahmet, UR: hmet

inflection class 8:

lemma: altun, UR: ltun

lemma: atar, UR: tar

inflection class 9:

lemma: anap, UR: nap

lemma: aram, UR: ram

inflection class 10:

lemma: azerbajcanlɨlar, UR: azerbajcanlɨlar

inflection class 11:

lemma: afjonkarahisar, UR: afjonkarahisar

inflection class 12:

lemma: arkadaʃlar, UR: arkadaʃlar

Remaining inflection classes ommited.

Sample rules:

inflection class 1:

rule: a l ⟨ ⟩ 0

rule: ~~i~~n r ⟨ numbersg casegen ⟩ 2

rule: a r ⟨ casedat numbersg ⟩ 3

rule: dan r ⟨ caseabl numbersg ⟩ 4

rule: ~~i~~ r ⟨ caseacc numbersg ⟩ 5

rule: lar r ⟨ casenom numberpl ⟩ 6

inflection class 2:

rule: a l ⟨ ⟩ 0

rule: e r ⟨ casedat numbersg ⟩ 1

rule: i r ⟨ caseacc numbersg ⟩ 2

rule: in r ⟨ numbersg casegen ⟩ 3

rule: eler r ⟨ casedat numberpl ⟩ 4

rule: inler r ⟨ numberpl casegen ⟩ 5

rule: den r ⟨ caseabl numbersg ⟩ 6

rule: ler r ⟨ casenom numberpl ⟩ 7

Notice that the two most populous inflection classes essentially capture back-harmonic and front-harmonic roots, respectively. Both inflection classes posit a spurious prefix, as with the example in Latin. The reasons for this are exactly the same as it is with Latin: these words happen to all start with the same segment. The search procedure cannot help but identify a spurious thematic prefix in these cases, and in the case of this data, which comes primarily from the beginning of the alphabet, there is little chance for this to be corrected by brining in a more diverse set of forms: all the word forms suffer from essentially the same deficit.

At the same time, however, no phonological rules can be found in the Turkish example. In certain ways, this is not surprising: vowel harmony is not a surface-true

fact about the training data, so the learner cannot hope to learn that vowel harmony is a generalization that has any standing in Turkish. In the absence of clear evidence for vowel harmony, it is not clear that there are any phonological rules which ought to be identified in Turkish—although the possibility still exists that, as is the case in the Latin experiments, some fact about the morphology of the language will be attributed to its phonological system.

As discussed in the context of Arabic, the analysis will be exactly the same whether the search for phonological rules is allowed, and no such rules are found, or whether only the merge from scratch procedure is employed.

Turkish 149 lemmas without phonological rules:

Overview:	Distinct surface forms:	929
	Lemmas:	149
	ICs:	132
Cluster analysis:	true positives:	32
	false positives:	0
	false negatives:	10994
	precision:	1
	recall:	0.00290223109015055
	f-score:	0.00578766503888587
URs:	true positives:	675
	false positives:	1
	false negatives:	29
	precision:	0.998520710059172
	recall:	0.958806818181818
	f-score:	0.978260869565217
phonological rules:	(none)	

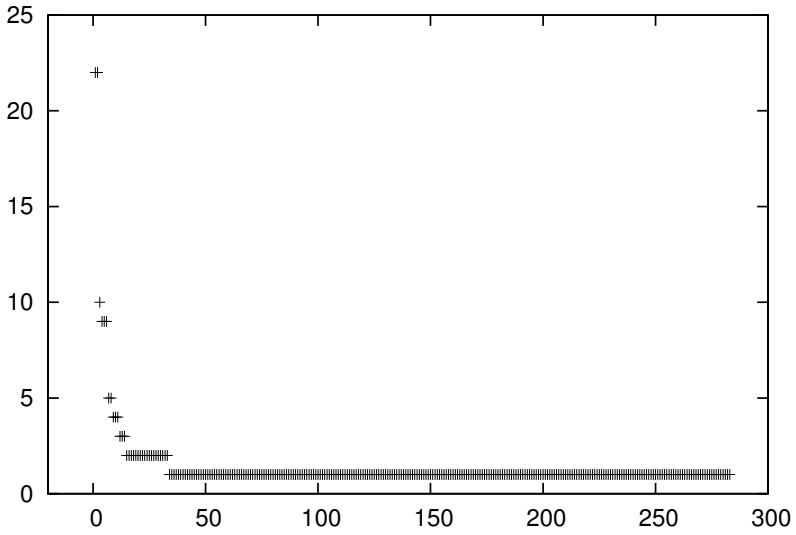

Figure 4.7: Lemmas vs. inflection class by rank, Turkish (large data set).

The output inflection classes for both experiments are identical, and they are not repeated here. Instead, the results for the larger data experiment are given below:

Turkish 400 lemmas without phonological rules:

Overview:	Distinct surface forms:	2427
	Lemmas:	400
	ICs:	283
Cluster analysis:	true positives:	681
	false positives:	0
	false negatives:	79119
	precision:	1
	recall:	0.00853383458646617
	f-score:	0.0169232489655944
URs:	true positives:	1854
	false positives:	1
	false negatives:	156
	precision:	0.999460916442048
	recall:	0.922388059701493
	f-score:	0.959379042690815
phonological rules:	(none)	

These results are notable because, once again, in terms of the clustering analysis, the precision is very high but the recall is fairly low. This is in contrast with the situation involving the URs posited for roots—in this case, both the precision and the recall are high, similar to the situation in the previous two experiments.

Ouput inflection classes:

inflection class 1:

lemma: abazalar, UR: bazalar

lemma: asɨrlar, UR: sɨrlar

lemma: ajrɨntɨlar, UR: jrɨntɨlar

lemma: avusturjalɨlar, UR: vusturjalɨlar

lemma: alaj, UR: laj

lemma: araʃtɨrmalar, UR: raʃtɨrmalar

lemma: amcam, UR: mcam

lemma: alaʃɨm, UR: laʃɨm

lemma: aɟahlar, UR: ɟahlar

lemma: antrenman, UR: ntrenman

lemma: amerikalɨlar, UR: merikalɨlar

lemma: arablar, UR: rablar

lemma: araʃtɨrmacɨlar, UR: raʃtɨrmacɨlar

lemma: altajlar, UR: ltajlar

lemma: aklɨnɨz, UR: klɨnɨz

lemma: ajaklanmalar, UR: jaklanmalar

lemma: avantaɟ, UR: vantaɟ

lemma: atinalɨlar, UR: tinalɨlar

lemma: anglikanlar, UR: nglikanlar

lemma: apartman, UR: partman

lemma: asurlular, UR: surlular

lemma: akɨncɨlar, UR: kɨncɨlar

inflection class 2:

lemma: akciɣerler, UR: kciɣerler

lemma: ansiklopediler, UR: nsiklopediler

lemma: adetler, UR: detler

lemma: alimler, UR: limler

lemma: analiz, UR: naliz

lemma: albɯm, UR: lbɯm

lemma: aker, UR: ker

lemma: alperen, UR: lperen

lemma: abid, UR: bid

lemma: asker, UR: sker

lemma: abbasiler, UR: bbasiler

lemma: amel, UR: mel

lemma: arʃiv, UR: rʃiv

lemma: aleviler, UR: leviler

lemma: askerler, UR: skerler

lemma: afetler, UR: fetler

lemma: acil, UR: cil

lemma: arʃivler, UR: rʃivler

lemma: asiret, UR: siret

lemma: annem, UR: nnem

lemma: anarʃizm, UR: narʃizm

lemma: aileler, UR: ileler

inflection class 3:

lemma: aspir, UR: spir

lemma: angl, UR: ngl

lemma: ailes, UR: iles

lemma: archiv, UR: rchiv

lemma: anem, UR: nem

lemma: ateʃkes, UR: teʃkes

lemma: alaʃehir, UR: laʃehir

lemma: alemler, UR: lemler

lemma: asiretler, UR: siretler

inflection class 4:

lemma: arɨc, UR: rɨc

lemma: aɟah, UR: ɟah

lemma: andɨr, UR: ndɨr

lemma: avatar, UR: vatar

lemma: amelijat, UR: melijat

lemma: avukat, UR: vukat

lemma: adɨm, UR: dɨm

lemma: aramaj, UR: ramaj

lemma: arad, UR: rad

inflection class 5:

lemma: ajet, UR: jet

lemma: anket, UR: nket

lemma: asid, UR: sid

lemma: arif, UR: rif

lemma: afet, UR: fet

lemma: acem, UR: cem

lemma: adres, UR: dres

lemma: amblem, UR: mblem

lemma: arabesk, UR: rabesk

inflection class 6:

lemma: anit, UR: nit

lemma: albert, UR: lbert

lemma: anc, UR: nc

lemma: aɟ, UR: ɟ

lemma: abaz, UR: baz

lemma: akkaj, UR: kkaj

lemma: aer, UR: er

lemma: atm, UR: tm

lemma: agor, UR: gor

lemma: akim, UR: kim

inflection class 7:

lemma: almanja, UR: lmanja

lemma: araptʃ, UR: raptʃ

lemma: adam, UR: dam

lemma: anadolu, UR: nadolu

lemma: amuc, UR: muc

inflection class 8:

lemma: azerbejcan, UR: zerbejcan

lemma: atmac, UR: tmac

lemma: alac, UR: lac

lemma: anajas, UR: najas

inflection class 9:

lemma: alic, UR: lic

lemma: arkadas, UR: rkadas

lemma: anlatim, UR: nlatim

lemma: a, UR: (null)

lemma: adm, UR: dm

inflection class 10:

lemma: ahmet, UR: hmet

lemma: album, UR: lbum

lemma: azer, UR: zer

lemma: alkal, UR: lkal

inflection class 11:

lemma: amat, UR: mat

lemma: arɨz, UR: rɨz

lemma: abdal, UR: bdal

lemma: altun, UR: ltun

inflection class 12:

lemma: ajhan, UR: jhan

lemma: alaʧam, UR: laʧam

lemma: ajdoɣan, UR: jdoɣan

Smaller inflection classes ommitted.

The facts about [a] as a spurious prefix hold for this experiment as well as for the first two. Note also that the two largest inflection classes contain a larger fraction of the lemmas in this output grammar than was true for either Arabic or Latin. This is a reflection of the facts about the inflection classes that exist in these various languages: in the linguists' grammars, there are five inflection classes for Arabic, and fourteen for Latin. As a result, even their largest clusters are much smaller. In the case of Turkish, however, there is only a single inflection class. This is reflected in the inflection classes that are posited by the learner: the first inflection class is devoted essentially to regularly-inflecting back-harmonic roots, whereas the second inflection class is devoted to regularly-inflecting front-harmonic roots.

Another fact to notice is that the search for feature syndromes is fruitless, even in Turkish, where, given the language's agglutinative morphology, one would hope to find the best evidence in favor of small syndromes. The reason for this is that the data is simply too sparse in the set of word forms associated with each lemma in the input data for the search to be successful. Although the facts about agglutination in Turkish make it the best candidate, out of the three languages tested, to get strong results for the syndrome search function, the fact remains that this particular search function demands virtually complete paradigms in order to correctly identify the syndromes that exist in a set of surface forms. Although the syndrome search is successful on toy data, the extremely rich set of tags used in this training data and the realistic

nature of the training data together mean that the syndrome search function will not return meaningful results.

The next chapter addresses several theoretical issues raised by the present work. It begins by discussing certain issues pertaining to morphological and phonological repreentations, and it goes on from there to discuss how aspects of the search procedures interact with the objective function to give the kinds of results seen here.

Chapter 5

Theoretical issues

This chapter deals with the theoretical issues that are raised by this system, particularly in the context of the results discussed in the previous chapter. It addresses the weaknesses of the system in terms of the representations of morphological and phonological generalizations that it deploys, and in terms of the particulars of the objective function and the search function. It also addresses certain aspects in which the representations, evaluation metrics, and search procedures perform surprising well, against expectation.

5.1 How complex are morphosyntactic representations?

The chapter begins by addressing the question of how complex morphosyntactic representations really are, and the simplifying assumptions that this work makes about morphosyntactic representations. Recall that the MSRs considered by the system are invariably "flat", in the sense that they consist of a list or set of features, each of which can take on a particular feature value. In no case are these representations

nested. For example, a MSR for an adjective in Spanish might be represented thus:

[gender: masculine , number: singular].

As Anderson [7] points out, though, there are languages in which a great deal more information is carried by a word's MSR; Anderson provides examples from Georgian and Potawatomi in which the verb clearly carries information from both the subject and the verb. This situation might be represented thus, with the more deeply nested features belonging to the object, and the less nested features belonging to the subject:

[gender: masculine , number: singular , person: first ,
[number: singular , person: second]].

A similar situation—though not as complex as the examples discussed in Anderson [7]—holds in Arabic, where the verb also carries information about both the subject and the object. However, it is not the case that this information must be represented with a nested structure per se. For example, it may be adequate to use a single feature to represent each possible combination of object features:

[gender: masculine , number: singular , person: first , object: second-singular]
[gender: masculine , number: singular , person: first , object: second-plural]
[gender: masculine , number: singular , person: first , object: third-singular]
[gender: masculine , number: singular , person: first , object: third-plural]
etc.

This kind of representation is sensible when each combined person-number value is associated with an exponent that is wholly distinct from the other exponents of object information, since it draws no connection between second-singular and second-plural, or between third-singular and third-plural, or between second-singular and

third-singular, or between second-plural and third-plural. Even so, though, a flat representation of the features can still be employed to capture relationships among second-plural objects and second singular objects:

[subj-gender: masculine , subj-number: singular , subj-person: first ,
obj-person: second , obj-number: singular]
[subj-gender: masculine , subj-number: singular , subj-person: first ,
obj-person: second , obj-number: plural]
etc.

This kind of representation may or may not be preferred for other theoretical reasons; additionally, it may or may not be well-grounded in terms of the psycholinguistic evidence, insofar as psycholinguistic methods can be applied to the question of morphosyntactic representations. The point remains, however, that a simple conversion exists between the fairly simple kinds of nesting that are sometimes posited in morphosyntactic representations, and the flat representations required by this learner. The fact that the learner is built specifically under the assumption that morphosyntactic representations can be given as an unstructured list or set of feature-feature value pairs does not inhibit its capabilities in any way, given the phenomena in natural languages that one would expect the learner to encounter.

5.2 Phonological structure and the representation of infixes and phonological rules

The manner in which this system represents infixes—and, for that matter, the way in which it represents the phonological form of words in general—is not ideal. The phonological form of a word is represented, in the current system, as a simple string of phones. The phones themselves contain a certain amount of internal structure,

since each phone is associated with a set of phonological features, with each feature taking on the value of either "+" or "−". This structure is used during the search for phonological rules. However, no higher-level phonological structure over these phones is recognized, such as the syllable or the foot. This lack of structure is justified, up to a point: since the system is concerned with the form of words in isolation, rather than in a particular phonological context, it makes sense to dispense with certain higher levels of phonological organization, such as the phonological phrase, the intonational phrase, and the utterance. However, there are smaller units of phonological structure that have been demonstrated, time and time again, to figure into the phonological form of lexical items; these include the syllable, the foot, and the prosodic or phonological word. Without representing these units of phonological structure explicitly, there is a limit to the adequacy of the grammars described by the present system.

For example, note that in the present system, the assumption is made that infixes can be placed appropriately simply by counting the number of segments from the left or right edge of the word, and placing the infix there. Any infix is consistently placed the same number of segments from the specified edge of the word into which it is being introduced, regardless of any facts about that particular words internal phonological structure. In many cases, this level of detail is adequate for placing infixes accurately. However, there are clearly cases where it is not—see, for example, the discussion of the placement of infixes in French [21] and Kager [34]. The evidence from French and Kager makes it clear that infixes, at least in certain languages, are more appropriately described in terms of syllable structure, rather than as elements that always appear a certain number of segments from either the left or right edge of a word.

Along similar lines, the same criticism can be leveled against the view that the present system takes of phonological rules. Phonological rules are also stated in such a way that they never refer to phonological constituents or structures at level above

the segment, when in fact there is ample evidence that many phonological generalizations refer to placement of particular segments within syllables, feet, or prosodic words. This is most clearly the case for generalizations concerning prosodic structure: the placement of primary and secondary stress in a word, and the structure of feet and syllables that underlie this stress placement, is obviously a kind of phonological generalization, and not one that can be captured without making reference to phonological organization above the level of the phone but below the level of the lexical word. Of course, there are many other rules that operate on individual segments, like those captured in the present system, that often seem to refer to the position of segments within the syllable. It is these rules in particular that might be captured more readily or more naturally under a system in which syllable structure is included in the phonological representation.

One might ask whether it would be appropriate to attempt to represent syllables and feet (and possibly prosodic words and moras) as part of the present system. Obviously, representing these prosodic constituents would represent a significant investment in the complexity of the system, although one would hope that the system would be able to use these structures in order to discover morphological and phonological generalization that are more in line with those identified by human linguists. For the moment, though, it makes sense to confine the remarks to outlining the changes that would have to be introduced in order to represent this kind of structure, and the kinds of challenges that it would pose for the search procedures.

The representation of these prosodic structures is not, in fact, conceptually difficult if, for each language, the learner is provided with the set of principles that are needed to correctly parse strings into syllables and feet. This is not entirely unreasonable, since one might argue that the parsing of phonological material into syllables and feet is in fact recoverable from the speech signal, although assuming that this information is given does highlight the weakness of the present system in compar-

ison with phonological learners that make use of ranked, violable constraints in an optimality theory context.

In any case, suppose that the learner has access to the principles that are necessary to perform a prosodic parse on the string in the language, or that the parse of each surface form is available to learner as part of the input data. The search for phonological and morphological rules then no longer occurs over flat representations. Take the phonological form of the word *dromedary* in English when represented as a string of segments:

#dramədɛri#

If the information about its syllabification and footing is available, either from the input data or from a set of principles that have either been fed to system or discovered by the system itself, the representation is somewhat different:

$$(((\text{dra})_\sigma(\text{mə})_\sigma)_\varphi((\text{dɛ})_\sigma(\text{ri})_\sigma)_\varphi)_\omega$$

It is now possible to define positions in the word in many more ways. For example, in the present system, infixes are specified in terms of being placed a certain number of segments from either the right or the left edge of the word. One might imagine a system in which infixes can be specified as being placed at a certain number of segments from the right or left edge, or a certain number of vowels from the right or left edge, or a certain number of consonants from the right or left edge. In these cases, no additional information is needed beyond the relative order of the characters in the string, as well as information about their assignment to the set of vowels or the set of consonants. If number of segments from the left or right edge is the only way to characterize the location of an infix, then there is only one way to specify the location of a suffix. If the position of an infix can be specified in terms of the number of segments, the number of vowels, or the number of consonants that it appears from

either the left or right edge, there may be more than one way to specify the position of a particular infix, depending on the set of forms in which one can observe it. Even so, though, the number of ways to specify its location is limited by the number of tiers that one is willing to recognize.

The set of ways in which one can specify the location of an infix is expanded when one works with the structured representation, however. At least at first glance, there may be a very large number of ways in which to specify an infix's position: perhaps it is placed a particular number of segments from the right or left edge, or perhaps it is placed a particular number of syllables from the right or left edge, or perhaps it is placed a particular number of feet from either the right or left edge. Of course, it is probably not reasonable for an infix to be specified as being placed an arbitrary number of syllables or feet from either the left or right edge of a word—it is a commonplace to say that natural language processes cannot count above two. However, it is presumably possible for an infix to be specified as having its target location at particular place in the first syllable. In short, the main point is this: a more appropriate representation of phonological structure allows certain morphological rules of infixation to be represented more appropriately, but at the expense of a much more complex search space for their insertion locations.

Working in this kind of environment, one might wish to entertain more complex hypotheses about phonological rules as well. Again, one finds oneself in a similar situation as one does with the placement of infixes—the number of possible hypotheses that one must consider is substantially larger than it was before. The infix situation is actually special: we have reason to believe, from the the set of infixation processes that are observed in the world's languages, that infixation may be specified in terms of prosodic constituents, such as segments, syllables, and feet, but that it really is not reasonable to expect to find infixes which are specified at an arbitrary number of constituents from a particular edge of the base form.

The situation is different with phonological rules, however: suppose that one finds an alternation between two segments. What is the correct environment for the rule? In the current system, the environment may be the single segments to the left or to the right, the two segments to the left, the two segments to the right, or the segments on both sides of the alternating segment. The environment can be specified in terms of the segments that truly do appear next to the alternating segment, or it can be specified in terms of the segments that appear on a particular tier, namely the vowel tier or the consonant tier. When one recognizes prosodic constituents in addition to consonant and vowel tiers, however, this issue becomes more acute: a particular alternation may be conditioned by the segments to which it is immediately adjacent, or it may be conditioned by any one of a number of environments that refer to prosodic structure in some way, rather than just the segments that appear in the appropriate neighborhood.

Essentially, the issue of prosodic structure in the search for infixes and phonological rules comes down to this: supposing that one has access to the appropriate kinds of prosodic structures—whether this is provided in the training data, or the rules for building the structure are acquired by the learner—one has the opportunity to identify phonological generalizations and infixation procedures that more closely resemble those discussed by linguists. This higher level of realism comes at a cost, however. First, one must manage prosodic structures as part of phonological representations at every level. Second, the space of possible rules is expanded, since it is not immediately clear what level of prosodic structure a particular rule should refer to. The present system avoids this issues regarding levels of prosodic structure by deploying the fiction that phonological representations are simply strings of segments. This is, after all, not an entirely unprecedented fiction: see Chomsky and Halle [15].

5.3 Blocking effects

Up to this point, the discussion has focused on issues relating to the kinds of grammars that the system is willing to consider—the issues have all centered on either phonological representations, or on the way in which morphological and phonological generalizations can be described. The fact that current system does not represent blocking effects well serves as a bridge to issues involving the search procedure and the evaluation of candidate grammars relative to one another. At issue in this section is the fact that the learner cannot recognize blocking effects. This is not a property of the way in which grammars are represented—in fact, the grammatical formalism used in the present system explicity recognizes blocking effects among morphological rules. Instead, it is a property of the search procedure that two morphological rules that exist at the same level of inflectional morphology, with one potentially blocking the other, cannot be discovered.

Consider the form in which grammars in the present work are stated. For both the linguists' grammar and the learner's grammar, a statement of the morphology of a language consists of set of inflection classes, where each inflection class consists of a set of inflection rules, and a set of words upon which those inflection rules may operate. For each inflection rule, a depth is given. This depth indicates relative precedence in the operation of rules—for instance, it specifies that and "inner" suffix must be applied before an "outer" suffix.

There is, in principle, no reason why several rules cannot be assigned to the same depth, and this is seen frequently in linguists' grammars. There are two main cases in which this occurs: first, when several rules all mark distinct syndromes, and they cannot compete to apply to the same forms. The other case is in which there is a default rule that applies to mark (for instance), the plural, while a more specific rule applies to mark the dative plural. When the more specific rule at that depth applies to mark the dative plural—and one can determine that it is more specific because

it is associated with a larger, more restrictive set of morphological feature values—it blocks the application of the more general plural rule. (For more on the application of a specific rule blocking the application of a more general rule at a particular stage in the derivation of a form, see Kiparsky [37], Matthews [45], Anderson [7], and Stump [61].) In particular, notice that the form in which grammars are stated, both by the linguist and by the learner, allows for the possibility of assigning several morphological rules to the same depth or level, and that it is this assignment of several rules to a given depth that allows blocking effects to come into play. This is in contrast with the grammars that the present system actually discovers, in which a new rule is always assigned to its own depth when it is discovered. In other words, the technique that is used to represent grammars allows one to represent grammars in which multiple inflectional rules occupy the same block or depth, with more specific rules edging out more general rules for application in cases where more than one rule at a particular depth would be able to apply. It is a property of the search procedure, however, that the learner is unable to identify morphological rules that belong to the same depth, regardless of whether specific rules block more general rules.

Think back to the technique that it used to merge inflectional classes from scratch. The morphological rules that are first discovered are the outermost prefixes and suffixes; then inner prefixes and suffixes are discovered, and finally infixes are uncovered. Every time a rule is discovered, it is assigned to a depth; each inflectional rule is assigned to its own depth. The learner will never attempt to assign two rules to the same depth, even if it is discovering rules on separate forms—instead, the learner finds a total ordering that correctly specifies the relative application of each morphological rule. It does not make generalizations about which rules, applied to different different word forms, might actually have been applied at the same depth of the derivation. Without this information, it is impossible for the system to identify rule blocking of the kind found in Anderson [7] and Stump [61].

One barrier to finding morphological rule blocking effects is posed by the nature of the search procedure used to merge inflectional rules from scratch. This search procedure identifies the relative order in which rules were applied to a particular word, as it searches for prefixes, suffixes, and infixes from the outside of a word form, working its way in. In order to group several different rules, each of which applies to a different word form, into the same rule block or depth, where they can compete for application, however, it is necessary to perform a further search that attempts to collapse the complete ordering of rules into a partial ordering that only respects the orderings that are required by the relative ordering of rule application on particular forms.

In fact, algorithms exist for translating a complete or total ordering over several lists into a partial ordering over a single list, in which all the precedence relations found in the several input lists are respected in the output partial ordering—see Davey and Priestley [17] on this point. Such a procedure by itself cannot, however, introduce rule blocking effects into the grammars output by the present system. Additionally, note that the present form of the search for feature syndromes does not present the merge function with sets of features in such a way that will allow for blocking effects to be discovered.

The reason springs from the way in which feature syndromes are identified: imagine again a situation in which plural forms are marked with a particular suffix *-a* at a particular depth, except for the dative plural, which is marked with a separate suffix *-b*. The ideal state of affairs, from a linguists point of view, is one in which [number:plural] is recognized as a feature syndrome, associated with a rule that suffixes *-a*. In the same depth is a rule that suffixes *-b* in the context of [number:plural, case:dative]. However, the mixture of plural forms ending *-a* and *-b* will mean that the syndrome search procedure will fail to even identify [number:plural] as a syndrome independent of any case information. The learner will

instead be left with -*b* as the marker for [number:plural, case:dative], and -*a* as the marker for [number:plural, case:nominative], [number:plural, case:accusative], [number:plural, case:genetive], etc., all as independent rules. There is simply no evidence, from the point of view of the syndrome search, that [number:plural] stands as an independent syndrome. In other words, the learner is not just unequipped to consolidate complete orderings over morphological rules into partial orderings. This is a key component to discovering rule blocking effects, to be sure, but the learner is also unequipped to discover the fact that -*a* is a marker of [number:plural] instead of a marker of [number:plural, case:nominative], [number:plural, case:accusative], and [number:plural, case:genetive], independently. The reason is that the syndrome search procedure itself relies on the fact that two forms associated with the same valid syndrome will always have the same substring in common. This is not true in a situation where a specific rule has edged out a more general rule for application, as in the [number:plural, case:dative] situation described above—although many of the forms marked with [number:plural] do indeed share the substring *a*, this is not true for the form marked with [number:plural, case:dative], since it instead has the character *b*.

In fact, algorithms for converting sets of complete orderings into a single partial ordering exist, as discussed above, but the syndrome search procedure serves as a weak link in the attempt to find grammars that a linguist would consider perfectly appropriate. The next section addresses another weakness of the syndrome search procedure.

5.4 Sparse data and the procedure for discovering syndromes

Another aspect that must be addressed is the fact that the system used for discovering syndromes at the outset of the search is only able to identify syndromes reliably when full paradigms are available to it, and that no such paradigms are in fact available in the training data. It should be noted that facts about Zipf's law make it unlikely

that different data sets, with a different distribution of forms, would give different results in terms of sparse data and the search for syndromes.

The key generalization in Zipf's law is this: for any corpus, the number of occurrences of a word of rank n will be roughly $\frac{1}{n}$ times the number of occurrences of the word of rank 1 in the corpus—see Zipf [67], [68] and Li [43]. The meaning of this fact in the present context is that as corpora get larger and larger, one can fairly expect to see more and more exemplars of words of the highest rank, and as more and more exemplars are encountered, one can fairly expect to see paradigms that are more and more complete, at least at the higher ranks. However, it also means that as larger and larger corpora are examined, there will be vastly more words for which only a few exemplars are encountered, and that the more complete paradigms higher on the list will never keep up with this. As Li [43] points out, this $\frac{1}{n}$ effect actually emerges in sets of randomly generated strings, and not just in natural language corpora, indicating that Zipf's law is less a generalization about human behavior or language use than it is a generalization about what happens when one draws stings from a particular kind of distribution, suggesting that there is no register or genre, however artificial, for which one could find a corpus whose word ranks and word frequencies do not follow this pattern. This is simply further encouragement for finding a method of identifying syndromes that does not rely on forms of individual words, but rather on larger sets drawn from more lemmas.

In short, the syndrome search used in the present system assumes that the training data will contain, for each word, enough distinct forms to allow the necessary syndromes to be discovered. When this search procedure is presented with sample complete paradigms, it does indeed perform as expected. When it is presented with naturalistic training data, however, there is simply not enough information present in order to identify syndromes correctly. The assumption that the syndrome search will have access to full paradigms in the training data is not realistic—a superior approach

to the problem of the syndrome search would, at the very least, make use of information found across several word forms. It might make sense to invoke this search as part of a variation on the merge from scratch search procedure, since that is the stage of the search in distinct lemmas are combined into a single inflection class, and reasonable inflectional rules are identified. This would require taking the syndromes associated with each inflection class as a fluid set during the search, however, rather than as a constant.

5.5 Limitations of surface-true phonological rules

In this section, some of the limitations of a system that assumes that all phonological generalizations must be surface-true, without exception, are discussed. Certainly, the sets of phonological generalizations discovered in the previous chapter is smaller than one might initially expect. This can be attributed, it would seem, to the fact that the present system assumes that all phonological generalizations that are identified with rules must be surface true, rather than to an aspect of the search procedure that discovers phonological rules. This section discusses the consequences of this assumption that all phonological rules must be surface-true, and it discusses several alternative approaches to the treatment of phonological generalizations.

First, consider the fact that many approaches to rule-based phonology specifically allow for the application of certain rules to be rendered opaque by the application of later rules. In other words, the assumption is that rules apply in a certain order, and while the application of any one rule may be without exception, there is no guarantee that the specific generalizations captured in individual rules will necessarily be surface true. Under this kind of framework, one might suppose that any individual phonological rule invariably applies when its structural description is met, and that only the application of later phonological rules might render the generalization captured in a particular phonological rule non-surface-true.

This description certainly captures the systems of ordered phonological rules found in Chomsky and Halle [15], Anderson [5], Kiparsky [37]. However, it is not clear that it is necessarily an improvement to assume that all phonological rules apply categorically when their structural descriptions are met, and that the only chance for a particular rule to be rendered non-surface-true is when a subsequent rule makes the earlier rule's application opaque. For one thing, this assumption poses an extremely difficult search problem: instead of being able to identify phonological rules independently, the learner can only posit a phonological system when all the rules together are able to apply with a probability of one, and to render the proper results together. This means that it would be much more difficult to identify phonological rules piece by piece, and to eventually integrate these into several phonological rules.

At the same time, while ordered rule application does seem to capture certain facts about the phonological systems of the languages of the world, it is not necessarily the case that rule-ordering opacity is the primary source of phonological opacity. In other words, there are clearly many situations in natural languages where a phonological generalization of some kind seems to hold, but where the generalization is not fully surface true. In some such situations, rule ordering effects can be used to explain the situation, but in others, the effect can be more naturally explained in terms of a generalization that applies with complete consistency only to a certain subset of the lexicon, or that applies only with some probability that is less than one.

A more utilitarian approach, therefore, would be to attach probabilities to phonological rules. In this way, a phonological rule would be said to apply with probability p when its structural description is met, not that it must necessarily apply whenever its structural description is met. This approach has several advantages: first, it can naturally capture situations in which a particular rule applies variably, as often happens with post-lexical rules or rules of "phonetic implementation", whose application is often governed by the speaker's speed or register, or whose application is simply

not perfectly consistent. In fact, attaching probabilities to phonological rules captures variable phonological rules perfectly. Probabilistic phonological rules are also not entirely unsuited to capturing phonological generalizations which do not apply uniformly throughout the lexicon, but which rather apply only to a certain set of words.

When developing the objective function to deal with such probabilistic rules, one might find that it is useful to posit a flat distribution over all values of p, or, more likely, one might find that it sensible to use a distribution that favors higher values of p to lower values, in order to favor grammars in which deploy phonological generalizations which are useful in a wide variety of contexts, rather than in just a few. In any case, probabilistic phonological rules would certainly be a useful tool for dealing with the reality of variable phonological rules, phonological rules that apply to only a subset of the lexicon, and simply the somewhat noisy data that one is invariably encounters in natural languages.

Another approach is to continue using rules that apply with probability $p = 1$, but that only apply to certain classes of words within the lexicon. For instance, one might assume that any given phonological rule is associated with a set of inflection classes, and it is only on the forms found in those inflection classes to which the phonological rule applies. This certainly mimics the intuition that certain phonological generalizations exist in natural language, but that they are sometimes confined in their effect to a particular class of words. One downside to this approach, however, is the fact that it opens the door for hyper-specific phonological rules which, although they may allow two inflection classes to be merged, are clearly too specific to be of any use elsewhere in the grammar. A human linguist, after all, can easily judge the generality or specificity of a phonological rule and make a determination as to whether or not it belongs in the grammar; in the case of an automatic learner like the present system, it may or may not be easy to translate this intuition into a term in the objective func-

tion. Another downside is the fact that it is not necessarily the case that inflection classes are contiguous with the classes over which phonological generalizations apply. In short, it seems that although the notion of co-phonologies might be appealing from the point of view of a human linguist, it is more likely that probabilistic phonological rule application will be easier to work with in a machine-learning context.

5.6 Benefits of categorical inflection classes and morphological rules

So far, this chapter has remarked on many aspects of the present system that are not ideal—in particular, it has pointed out the problems associated with the particulars of the representations used for morphosyntactic representations, phonological forms, phonological rules, and the aspects of the search for morphological rules and syndromes. At this point, however, it is appropriate to remark on several aspects of the present system that are remarkably useful, despite initial appearances. First, consider the fact that this system uses categorical inflectional classes and morphological rules: in other words, the fact that the probability that a particular morphological rule will apply, when given the right context, is always one, and that a single word may belong to only one inflection class, with no ifs, ands, or buts about it. This is in contrast with the phonological rules, which are also categorical, but for which the categorical nature has been discussed as a liability, not an asset.

Essentially, representing inflection classes and morphological rules categorically is a very good thing for two important reasons. The first is that this categorical assumption, while unusual from the point of view of a Bayesian induction system, is quite helpful in the fact that it cuts the size of the search space from unimaginably vast to manageably huge. The reasoning here is simple: if a hypothesis includes even a single word associated with an inflection class that fails to correctly predict

an observed training form, the hypothesis can be discarded. Were inflection classes and morphological rules to be treated probabilistically, this would not hold: such a grammar might still generate most of the other data correctly most of the time, and it might generate the data point in question correctly some fraction of the time, so a much wider landscape of hypotheses would be in consideration at any given point during the search. The second is that this categorical assumption is actually quite reasonable, given what we know about the inflectional morphologies of natural languages: while phonology is rife with opacity, variability, and probabilistic effects, morphology is essentially the domain of the idiosyncratic. While there are occasionally cases in which morphological rules are probabilistic, or the assignment of a particular word to a particular inflection class is variable, these effects are much less widespread in the inflectional morphologies of natural languages than in other components of grammar.

One aspect in which the system of strict assignment of words to distinct inflection classes might be adjusted is in terms of the assumption that each inflection class is indeed completely independent of all others. For example, it might be appropriate to represent class inheritance in inflection classes—the idea being that Class A might provide Class B with all of the rules that it needs, except for a certain few rules for which Class B would provide its own distinct rules. Such an approach might, for instance, keep the fourth declension in Latin from being shattered among so many distinct inflection classes on the basis of irregular nominative singular forms; it would also allow a number of neuter paradigms to be included with their masculine and feminine neighbors, as these forms typically differ only in terms of the nominative and accusative plural forms. While it is true that this kind of relationship among inflection classes would more closely approximate the relationships among classes that linguists posit (see, in particular, Stump [61]), it is not clear that this approach would lead to much more than a cosmetic difference. After all, it is probably convenient

for a human to remember the similarities that exist between inflection classes, but it does not necessarily follow that this would lead to analysis of particular words that are different in any significant way. This kind of approach might show a connection between the various fourth declension classes, for example, but it would not lead to an analysis that is any better in terms of the rules or the underlying representation that are associated with a particular word. Overall, the assumptions that the present system makes about the form of inflection classes appear to be well-founded.

5.7 Remarks on the objective function and the search procedures

As a final remark, turn again to the objective function described in chapter 2. Consider first the inverse squares series that appears so many times in this objective function. This inverse squares series is used extensively in Snover, Jarosz, and Brent [60] and the current work shows that it is in fact a surprisingly appropriate way to represent the probability of certain values—such as the the number of inflection classes, the number of roots in an inflection class, and the number of segments in a root—despite the fact that one might initially suppose that it would favor a preference for uniformly-sized inflection classes. After all, in practice, it is not the p(|IC|) term that mitigates against large numbers of inflection classes—rather, a grammar's probability dwindles as it takes on more and more inflection classes because of the probabilities associated with each of those individual inflection classes. There are certainly other distributions—such as the Poisson distribution—which cover a similar set of values in a similar way, but with a much lighter tail.

In general, however, it is not the objective function bur rather the search procedure that sometimes leads to the mis-analysis of inflectional morphology. Consider, for example, the situation in the smaller Turkish experiments. The probability of the

learners's grammar in this case is about 10^{-83463}, whereas the corresponding probability for the linguists' grammar is about 10^{-637}. This can be interpretted as meaning that the linguists' grammar is about 10^{82813} times more probable than the learner's grammar.

The other experiments are likely to give similarly unbalanced results, although it is more difficult to find these figures when the assignment of lemmas to inflection classes is under-determined, as discussed in section 4.1: although the evaluation certainly makes use of a comparison between the learner's grammar and the linguists' grammars, it is not at all obvious how to assign lemmas to inflection classes in the linguists' grammar for this particular kind of evaluation task, in which the probability of two grammars is compared directly. Additionally, it is not immediately clear which inflectional rules in the linguists' grammar are required in order to account for the data in a particular data set. The example from the smaller Turkish experiment, however, provides an easy solution, because all the lemmas are necessarily clustered into one class in the linguists' grammar, and the set of inflectional rules in that grammar can be easily managed. The fact of the matter is, though, that we have every reason to think that the linguists' grammars would out-perform the learner's grammars based on other training sets in a similar way: in every case, the linguists' grammar partitions the set of lemmas into a much, much smaller set of inflection classes than does the learner's grammar, while the sets of inflectional rules are roughly comparable.

The key fact, however, is that the insight of human linguists results in grammars that are, according to the objective function deployed here, vastly more probable than the grammars discovered by the search procedure. If the learner is arriving at grammars that differ from the target grammars in some way, it is not because it is able to find grammars that surpass the target grammars, from the point of view of the objective function, in some way. Rather, the search procedures are keeping the

learner locked in a realm of lower probability grammars.

Along similar lines, it can actually be shown that a particular aspect of the search function sometimes prefers sub-optimal analyses. Consider the following: the learner's objective function often prefers to avoid some of the spurious morphological analysis that the learner nevertheless identifies. The reasoning here is that introducing a spurious prefix one character long into an inflection class will reduce its probability by about 10^{-5} times, depending on the frequency of the segment in question, the set of morphological feature values that exist in the language, and the number of other inflectional rules that exist in the inflection class. Does introducing this spurious rule pay for itself, in the sense that it can make the roots that are present in the inflection class?

This depends on the facts about the roots that are found in the inflection class. Eliminating a single character from each root in an inflection class will cause the probability of each root to increase by somewhere in the neighborhood of 20 to 40 times, depending on the length of the roots that appear in that inflection class, but primarily the frequency of the segment being pulled out into the spurious affix. This means that identifying spurious thematic affixes in very small inflection classes will actual harm the overal probability of the inflection class: although the probability of each root in the inflection class will go up by a factor of about 20 to 40, give or take, the inflection class will still be left in debt if it only contains two, three, or four lemmas. Remember that the learner will adopt this sub-optimal analysis because it is still a vast improvement on leaving the constituent inflection classes unmerged. The issue here is not whether particular merges are being adopted, but rather whether the analyses that are produced for them are correct. Because of properties of the search function, many inflection classes containing spurious thematic suffixes are distinctly sub-optimal.

One might ask whether it would be appropriate to take any kind of action in the

search function to attempt to mitigate this problem. It is not clear that any action would indeed be appropriate—certainly, when the data set is large and diverse, many instances of these spurious suffixes will be eliminated in the normal course of merging inflection classes that contain a diverse set of lemmas. After all, the learner does have a compelling need to improve the overall probability of the grammar, and in most cases, merging two inflection classes is a good way to accomplish this. Under a more realistic set of training data than that presented in the Turkish examples, for example, spurious thematic affixes should be relatively rare. They are certainly present in the Latin examples, but they do not overwhelm the outputs.

It should be noted, in fact, that the objective function used here seems to perform reasonably well overall, at least insofar as these experiments even test it—the search procedure is really the limiting factor in these experiments. It is not clear that modifying the objective function would improve the performance of the learner on training data in general, or even produce substantially different results. It is not likely, for instance, that using a Poisson distribution rather than the series of inverse squares throughout the objective funciton would impact the big-picture performance of the objective function substantially.

One might potentially want to modify the objective function so that it penalizes thematic elements in some way, so as to mitigate against analyses that contain these kinds of thematic elements. It is not obvious, though, that this functionality really belongs in the objective function rather than the search function. One might just as easily ask whether it makes sense to allow the search function to back off of thematic elements, so that the objective function can correctly distinguish between analyses in which a thematic element is included, and analyses in which it is not. Under this approach, the search procedure would attempt to provide two alternative merges to the objective function—one in which a particular thematic element is identified, and one in which it is ignored. The objective function can then function in its normal way

to distinguish between the two hypotheses. This seems to be the best approach from the point of view of letting the objective function do the job that it was intended to do, and making sure that the objective function does not mitigate inappropriately against phenomena that it really ought to capture—such as legitimate thematic elements.

This discussion serves to highlight the way in which the performance of the learner, measured in terms of the output of the objective function as applied to its terminal states, is far from optimal. The discussion in this section has shown that the fact that the learner's grammars do not approximate the linguists' grammars more closely can be attributed to facts about the search function, not the objective function. This leads to the issue of directions for future research, which is addressed in chapter 6.

Chapter 6

Conclusion

This chapter discusses the accomplishments and the significance of the present system. That is, it discusses the larger context of this work, and it highlights the specific areas in which this work is novel, in terms of our understanding of the problems associated with inducing a grammar of phonology and inflectional morphology, and in terms of the generalizations that can be drawn from realistic natural language data. This chapter goes on to identify several areas in which the potential extensions of this work are the most promising, and it explains the kinds of work that can be expected to follow it in the future.

6.1 Accomplishments and significance of the present system

The present system represents a key landmark in the development of systems of morphological induction. First, given a set of surface forms and tags—and surely these are both more or less available to learners—the system is able to find a morphological and phonological grammar. Moreover, the output grammars are reasonably apt, when compared with the kinds of grammars written by a human linguist, and the search entertains a large space of phenomena that are found in natural languages.

Specifically, the learner entertains hypotheses that make use of prefixes, suffixes, and infixes. Finally, the learner is also unique in the way in which it uses the search for inflection classes in order to find phonological generalizations. The present system certainly represents a significant step forward in terms of the state of the art for systems inducing morphological systems based upon "words and rules", and integrating the morphological grammar with the phonological grammar.

There are also specific aspects of the present system that are particularly notable. For one thing, the objective function is based, in large part, on the work of Snover, Jarosz, and Brent [60]; however, the present system extends it so that it can assign probabilities to grammars very much like those stated in Anderson [7] and Stump [61], and which include a component that expresses phonological generalizations. Furthermore, the present work clarifies the nature of the problem associated with searching for feature syndromes as a prerequisite for searching for morphological rules and inflection classes. Although the previous chapter highlighted certain downsides of the present solution to the problem of finding suitable syndromes, this work defines the problem and proposes a solution that works under certain sets of training data. A perfect solution may not even be possible, given the scale of the search space, but this work provides a solution that can be deployed in certain situations when complete paradigms are available in the training data.

The present work is also notable for bringing the search for inflectional rules and for phonological rules together. In particular, a great deal of the previous work on the induction of phonological generalizations, whether rule-based or constraint-based, has concentrated on discovering phonological generalizations based strictly on surface forms, or on the basis of pairs of underlying representations and surface forms. The present system, on the other hand, treats phonological generalizations as existing completely in the service of morphological generalizations. Clearly, this is not entirely appropriate, as human learners are surely able to identify many phonological

generalizations in their languages that are orthogonal to the issue of compact representations of inflectional morphology. Nevertheless, there is something very natural about using the laboratory of morphological alternations to discover phonological generalizations in a language, and the present system attempts to exploit that.

Finally, while the search for phonological rules is not utterly fruitless, the number and scope of the phonological rules that are discovered is, to be fair, more limited than those that would be identified by a human linguist. The phonological rules that are identified seem to capture generalizations that a human linguist would probably describe as morphological generalizations, rather than phonological generalizations. This is not necessarily a failure of the learner, however. Instead, the present work indicates that it is indeed possible to identify morphological and phonological generalizations in concert, but that phonological generalizations in particular may be best represented in a way that does not assume that they are invariably surface true. Assuming that phonological generalizations are surface true means that many key facts about the phonology of a language are going to be overlooked—a generalization does not have to be exceptionless for it to still warrant attention.

6.2 Directions for future research

In this final section, several areas in which the work in this dissertation leads naturally to areas of future research are explored. For instance, it would be appropriate to revise the objective function so that it can assign probabilities to co-phonologies or probabilistic phonological rules in a sensible way, in order to refine the way in which phonological rules are applied—as discussed in chapter 5. Of course, there are many other ways in which the objective function could be revised, perhaps using other distributions to assign probabilities to the assignment of lemmas to inflection classes, or to particular roots or rules. The way in which these kinds of changes would impact the output is not immediately obvious: they might result in grammars that linguists

judge to be more apt, or they might result in grammars that are less suitable, but most likely they would produce little appreciable change. It is difficult to predict the effect that changing the objective function would have on the way in which the system functions. In general, however, it seems that the outputs of the system are more strongly determined by the search function than by the objective function, so this area appears to be a much more fertile area for research—the possible outcomes for any particular change to the system are simply easier to predict, or at least imagine.

This means that modifications to the search system are an open door for further research: it is the search system, and not the objective function, that is responsible for most of the pathological behaviors that are observed in the present system. In cases where the output fails to follow the human linguists' grammar as closely as one might like, the blame seems to rest with search procedures rather than the objective function. Regardless, even if there are imperfections in the objective function, the search space in this kind of problem is exceptionally large, under any objective function—the search procedures that one adopts are directed not at the space of morphological grammars the formal system admits, but rather at the subset in which one imagines appropriate grammars of human languages can be found. When one makes decisions about the search strategy to adopt, one can focus on the kinds of phenomena that one expects to find in natural languages. As mentioned in the previous chapter, finding a better way to deal with spurious thematic material might be one way in which the search procedure might be modified.

There are certain other aspects of the present search that stand out as being particularly ripe for experimentation as well, though. For example, as discussed in the previous chapter, one weakness of the current version of the search function is the fact that the syndrome search relies upon very complete information about individual word forms at the beginning of the search. One area in which the present system could well be extended would be with a different approach to the syndrome

search. In particular, a search that attempts to find feature syndromes at every point at which inflection classes are merged on the basis of common substrings or subsequences might be promising. Obviously, the challenge here is the fact that there are so many possible syndromes in a group of features, and that searches for common subsequences within strings are often quite costly, at least if one expects the search to be complete. Evidently, a good heuristic function will be required in order to make this kind of approach viable. The current version of the syndrome search is clearly a preliminary solution to the problem of how to identify feature syndromes, but there may be other solutions that are able to make use of facts about natural languages in order to avoid a complete search while identifying a large portion of the feature syndromes that a linguist would identify.

Along similar lines, it does not appear that the hill-climbing search strategy, in which the best possible merge is chosen at each point at which a merge is considered, is inappropriate, given the batch-processing nature of the present system. That is, the present system takes a large set of data as input, it finds an appropriate initial state based on that data, and it searches for an improved grammar by looking for whichever neighboring grammar results in the biggest increase in the objective function.

However, if one were to outfit the system with a different procedure for searching for syndromes and taking in data points gradually—as one supposes that human learners must do—one might want to change the structure of the search. In that situation, a hill-climbing search might not appropriate, since facts about small sets of data available early on might not set the learner up to make good generalizations later on, when more complete data is available. Simply put, the hill-climbing search seems to work well for a system in which the full set of training data is provided at the beginning of the search; a system that mimics a human learner by acquainting itself with a progressively larger set of data during the search might find that the hill-climbing search is less appropriate.

Another aspect of the present system that would likely lead to fruitful research in the future is an exploration of alternative representations of phonological rules. As discussed in chapter 5, one key weakness of the present system is the fact that it treats all phonological rules as generalizations that must be surface true in a language. This is clearly a step up from treating the acquisition of morphology and the acquisition of phonology as two wholly distinct problems, but it certainly involves certain simplifying assumptions about phonological generalizations. For one thing, it clearly cannot hope to capture phonological rules that only apply part of the time—as often happens with post-lexical phonological rules, or rules of "phonetic implementation". For another, it cannot capture phonological generalizations whose effect is sometimes obscured by the ordered application of other phonological rules, or whose effect is simply not consistent throughout the lexicon.

Of these cases, the case in which phonological generalizations are true but are obscured by other phonological generalizations is probably the hardest to discover procedurally. To be sure, it is certainly easy to represent a grammar in which phonological rules apply in a particular order, with the application of later rules possibly obscuring the application of earlier rules—one simply need to write out the rules as an ordered list of finite-state transducers, and to pass the output of each transducer to the next transducer as input. However, the search for such systems is much less simple, since one must look for generalizations that are, individually, not always true, but for which one can find an order that allows the generalizations to apply without exception, and to produce the appropriate surface forms. This is certainly a tempting objective, but there are other ways to improve the search for phonological rules more incrementally.

One such solution is to simply say that phonological rules need not apply without exception—instead, any given phonological rule need only apply with some particular probability. Notice that this kind of approach captures variable phonological

rules quite naturally; it can also allow one to find generalizations which are sometimes rendered opaque or which only apply to a subset of the lexicon, although the association of the rule with a probability is less intuitive in such a case. The most obvious way to introduce this feature to the objective function would be to assign probabilities to phonological rules using a probability distribution such as the beta function, which assigns probabilities to the real numbers over the interval (0,1), or the triangular function, which can be used to assign probabilities to the real numbers over the interval [0, 1]. The exact shape of both of these probability distribution can be adjusted according to the parameters that they take. This allows higher values (such as 0.8) to be favored over lower values (such as 0.2). Such an approach seems appropriate, since it allows the system to favor phonological generalizations that it it observes more frequently in the data. During the search, the system can assign probability to phonological rules based on the number of times that might have applied in the data, divided by the sum of the number of times that they might have applied and the number of times that they did not apply in a way that is surface true. In essence, probabilities can be assigned to phonological rules based on empirical evidence, and the objective function will cause the system to prefer those rules that have the highest probabilities, by assigning a higher probability to grammars that contain high-probability rules.

Another solution to the problem of phonological generalizations that are not always surface true is to say that phonological generalizations are associated with sets of inflection classes. This kind of approach seems to capture the situation posed by co-phonologies quite naturally, but there are certain aspects of this approach that may not be ideal: for one thing, it might open the door for ultra-specific phonological rules that can be used to collapse inflection classes inappropriately, or that are at the very least are not general enough; for another, it is not immediately clear that the lexical classes to which phonological generalizations sometimes refer are necessarily

aligned with inflectional classes. This last issue seems to be particularly acute in cases where the lexical classes are determined essentially by the etymology, rather than by a morphological property.

A final aspect in which the present work can be extended is through the use of knowledge-free techniques to induce the tags present in the training data, rather than relying on tagged data. At first glance, this extension of the present work might not seem interesting—after all, it would presumably compound the error inherent in the present system with the error inherent in a knowledge-free attempt to identify morphological and syntactic categories. Notice, though, that there are several reasons why bringing knowledge-free techniques to bear on the present system would be fruitful. For one thing, it would dramatically expand the number of languages on which the system can be tested, particularly into the realm of languages for which electronic corpora are available (either collected specifically as corpora, or simply scraped from websites), but for which tagged data is not available. For another, the induction of tags in a knowledge-free setting would presumably depend upon both the distributional properties of words, and on the resemblance that certain strings have with others. Bringing string processing techniques and distributional information together to the problem might allow one to identify strings directly, while leaving the identification of actual features values aside. In any case, the integration of knowledge-free techniques with the present system is certain to lead to fruitful results.

Bibliography

[1] Faruk Abu-Chacra. *Arabic: an essential grammar*. Routledge, 2007.

[2] Martin Aigner and Günter M. Ziegler. *Proofs from* THE BOOK. Springer, third edition, 2004.

[3] Adam Albright. *The identification of bases in morphological paradigms*. PhD thesis, University of California, Los Angeles, 2002.

[4] Adam Albright and Bruce Hayes. An automated learner for phonology and morphology. University of California, Los Angeles, manuscript, 1999.

[5] Stephen R. Anderson. *The organization of phonology*. Academic Press, 1974.

[6] Stephen R. Anderson. *Phonology in the twentieth century: theories of rules and theories of representations*. University of Chicago Press, 1985.

[7] Steven R. Anderson. *A-morphous morphology*. Cambridge University Press, 1992.

[8] Marco Baroni, Johannes Matiasek, and Harald Trost. Unsupervised discovery of morphologically related words based on orthographic and semantic similarity. In *Proceedings of the Sixth Meeting of the ACL Special Interest Group in Computational Phonology*, pages 48–57, 2002.

[9] Kenneth R. Beesley and Lauri Karttunen. *Finite state morphology*. CSLI Publications, 2003.

[10] Charles E. Bennett. *New Latin grammar*. Allyn and Bacon, 1918.

[11] Andrew Carstairs. Paradigm economy. *Journal of linguistics*, 19(1):115–125, 1983.

[12] Erwin Chan. Learning probabilistic paradigms for morphology in a latent class model. In *Proceedings of the Eighth Meeting of the ACL Special Interest Group in Computational Phonology*, pages 72–78, 2006.

[13] Erwin Chan. *Structures and distributions in morphology learning*. PhD thesis, University of Pennsylvania, 2008.

[14] Nick Chater and Paul Vitányi. Simplicity: a unifying principle in cognitive science? *Trends in cognitive sciences*, 7(1):19–22, 2003.

[15] Noam Chomsky and Morris Halle. *The sound pattern of English*. Harper & Row, 1968.

[16] David Crystal. *A dictionary of language*. University of Chicago Press, second edition, 2001.

[17] B. A. Davey and H. A. Priestley. *Introduction to lattices and order*. Cambridge University Press, second edition, 2002.

[18] Carl G. de Marcken. Linguistic structure as composition and perturbation. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics*, pages 335–341, 1996.

[19] Carl G. de Marcken. *Unsupervised language acquisition*. PhD thesis, Massachusetts Institute of Technology, 1996.

[20] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the *EM* algorithm. *Journal of the Royal Statistical Society, Series B (Methodological)*, 39(1):1–38, 1977.

[21] Koleen Matsuda French. *Insights into Tagalog reduplication, infixation, and stress from nonlinear phonology*. Summer Institute of Linguistics and the University of Texas at Arlington, 1988.

[22] Daniel Gildea and Daniel Jurafsky. Learning bias and phonological-rule induction. *Computational linguistics*, 22(4):497–530, 1996.

[23] Aslı Göksel and Celia Kerslake. *Turkish: a comprehensive grammar*. Routledge, 2005.

[24] John Goldsmith and Yu Hu. From signatures to finite state automata. In *Midwest Computational Linguistics Colloquium*, 2004.

[25] John A. Goldsmith. Unsupervised learning of the morphology of a natural language. *Computational linguistics*, 27(2):153–189, 2001.

[26] John A. Goldsmith. An algorithm for the unsupervised learning of morphology. *Natural language engineering*, 12(4):353–371, 2006.

[27] Sharon Goldwater, Thomas L. Griffiths, and Mark Johnson. Interpolating between types and tokens by estimating power-law generators. *Advances in Neural Information Processing Systems*, 18:459–466, 2006.

[28] 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*, pages 35–42, 2004.

[29] Ronald L. Graham, Donald E. Knuth, and Oren Patashnik. *Concrete mathematics: a foundation for computer science*. Addison-Wesley, second edition, 1994.

[30] Dan Gusfield. *Algorithms on strings, trees and sequences: computer science and computational biology*. Cambridge University Press, 1997.

[31] Zellig S. Harris. From phoneme to morpheme. *Language*, 31(2):190–222, 1955.

[32] C. Douglas Johnson. *Formal aspects of phonological description*. Mouton, 1972.

[33] Howard Johnson and Joel Martin. Unsupervised learning of morphology for English and Inuktitut. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, volume 2, pages 43–45, 2003.

[34] René Kager. *Optimality theory*. Cambridge University Press, 1999.

[35] Ellen M. Kaisse and Sharon Hargus. Introduction. In Sharon Hargus and Ellen M. Kaisse, editors, *Studies in lexical phonology*, pages 1–19. Academic Press, 1993.

[36] Michael Kenstowicz and Charles Kisseberth. *Generative phonology*. Academic Press, 1979.

[37] Paul Kiparsky. "Elsewhere" in phonology. In Stephen R. Anderson and Paul Kiparsky, editors, *A festschrift for Morris Halle*, pages 93–106. Holt, Rinehart, and Winston, 1973.

[38] Paul Kiparsky. Lexical morphology and phonology. In *Linguistics in the morning calm: selected papers from SICOL-1*, pages 3–91. Hanshin, 1982.

[39] Paul Kiparsky. Blocking in non-derived environments. In Sharon Hargus and Ellen M. Kaisse, editors, *Studies in lexical phonology*, volume 4 of *Phonetics and phonology*, pages 277–313. Academic Press, 1993.

[40] Paul Kiparsky. Opacity and cyclicity. *Linguistic review*, 17(2–4):351–367, 2000.

[41] Jacob M. Landau. *Exploring Ottoman and Turkish history*. Hurst and Company, 2004.

[42] G. L. Lewis. Atatürk's language reform as an aspect of the modernization of Turkey. In Jacob M. Landau, editor, *Atatürk and the modernization of Turkey*, pages 195–214. Westview Press, 1984.

[43] Wentian Li. Random texts exhibit Zipf's-law-like word frequency distribution. *IEEE transactions on information theory*, 38(6):1842–1845, 1992.

[44] David J. C. MacKay. *Information theory, inference, and learning algorithms*. Cambridge University Press, 2003.

[45] P. H. Mattews. *Inflectional morphology: a theoretical study based on aspects of Latin verb conjugation*. Cambridge University Press, 1976.

[46] John J. McCarthy. A prosodic theory of nonconcatenative morphology. *Linguistic inquiry*, 12(3):373–418, 1981.

[47] John J. McCarthy. Linear order in phonological representation. *Linguistic inquiry*, 20(1):71–99, 1989.

[48] John J. McCarthy. *Hidden generalizations: phonological opacity in optimality theory*. Equinox, 2007.

[49] Floyd L. Moreland and Rita M. Fleischer. *Latin: an intensive course*. University of California, 1977.

[50] Jason Naradowsky and Sharon Goldwater. Improving morphology induction by learning spelling rules. In *International Joint Conferences on Artificial Intelligence*, pages 1531–1536, 2009.

[51] Lewis M. Paul, editor. *Ethnologue: languages of the world*. SIL International, sixteenth edition, 2009.

[52] Alan Prince and Paul Smolensky. *Optimality theory: constraint interaction in generative grammar*. Blackwell, 2004.

[53] Ernst Pulgram. *The tongues of Italy*. Harvard, 1958.

[54] William M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336):846–850, 1971.

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

[56] Stuart J. Russell and Peter Norvig. *Artificial intelligence: a modern approach*. Prentice Hall, second edition, 2003.

[57] Patrick Schone and Daniel Jurafsky. Knowledge-free induction of morphology using latent semantic analysis. In *Proceedings of the Fourth Conference on Computational Natural Language Learning (CoNLL)*, pages 67–72, 2000.

[58] Domenico Silvestri. The Italic languages. In Anna Giacalone Ramat and Paolo Ramat, editors, *The Indo-European languages*, pages 322–344. Routledge, 1998.

[59] Matthew G. Snover. An unsupervised knowledge free algorithm for the learning of morphology in natural languages. Master's thesis, Washington University, 2002.

[60] 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 Sixth Meeting of the ACL Special Interest Group in Computational Phonology*, pages 11–20, 2002.

[61] Gregory T. Stump. *Inflectional morphology: a theory of paradigm structure*. Cambridge University Press, 2001.

[62] Lewis V. Thomas and Norman Itzkowitz. *Elementary Turkish*. Harvard University Press, 1967.

[63] C. J. van Rijsbergen. *Information retrieval*. Buttersworths, 1975.

[64] Edoardo Vineis. Latin. In Anna Giacalone Ramat and Paolo Ramat, editors, *The Indo-European languages*, pages 261–321. Routledge, 1998.

[65] Jane Wightwick and Mahmoud Gaatar. *Arabic verbs and essentials of grammar*. McGraw-Hill, 2008.

[66] David Yarowsky and Richard Wicentowski. Minimally supervised morphological analysis by multimodal alignment. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 207–216, 2000.

[67] George Kingsley Zipf. *The psycho-biology of language: an introduction to dynamic philology*. Houghton Mifflin, 1935.

[68] George Kingsley Zipf. *Human behavior and the principle of least effort*. Addison-Wesley, 1949.