Chapter 3

Identifying bases algorithmically

In chapter 2, I informally demonstrated the approach of comparing different forms of the paradigm to determine their "informativeness," and showed that the most informative form was also the one that acted as the base of a paradigm leveling that took place in the history of Yiddish. In this chapter, I propose an algorithm for comparing informativeness in a more systematic fashion. The algorithm starts by considering each member of the paradigm as a candidate for base status, and constructs grammars that use each as an input to derive the remainder of the paradigm, as seen in Figure 3.1.

More formally, the model starts by taking paradigms of related forms (A, B, C, D, E, ...), and considering all of the pairwise relations between them (A→B, A→C, ..., B→A, B→C, B→D, ...). I will call such pairwise relations *mappings*. For each mapping between two slots in the paradigm (X→Y), the model constructs a set of rules transforming X's into Y's. The result is what I will call a *sub-grammar*, consisting of all of the rules that describe a particular morphological mapping between just two slots in the paradigm (X→Y). A complete morphological grammar is a set of such sub-grammars, including at least one sub-grammar to derive each slot in the paradigm.

Given this terminology, it is possible to give a more precise formulation of the learner's task:

- The learner must be able to produce and comprehend forms from all parts of the paradigm (A, B, C, D, E, ...) accurately and confidently
- Thus, the learner must identify which member of the paradigm contains the most information about how to derive the remainder of the paradigm

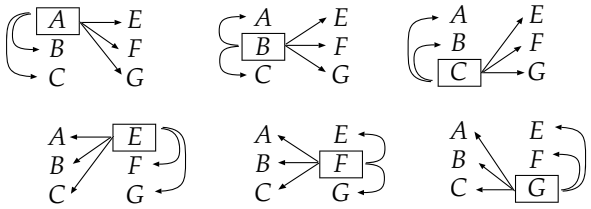


Figure 3.1: Comparison of different candidates for base status

- The most informative form is the one that permits the most *accurate* or *efficient* subgrammars (under a form definition of accuracy and efficiency, to be given below)
- Once the learner has compared the relative informativeness of various forms in the paradigm, the most informative one is selected as the base, and the grammar that is chosen is the one which operates on that base form to derive the remaining forms in the paradigm

Before we can find the optimal mappings for a language, we need two things: (1) a rule discovery system to learn the sub-grammars originating from different candidate base forms, and (2) a set of metrics that can score the resulting sub-grammars as more or less optimal. For the first task, I will use the *minimal generalization* rule induction system developed by Albright and Hayes (1999a, to appear). After giving a brief outline of this model in section 3.2, I will propose a set of metrics to evaluate the goodness of sub-grammars in section 3.3. Before describing the implemention, however, it is useful to review some of the considerations that led to the choice of this particular system.

3.1 Desiderata for an automated learner

Not all of the currently available systems for rule discovery produce grammars that are equally suitable for comparing their efficiency or certainty. Moreover, the choice of model crucially determines the types of comparisons which will be possible. The model of morphological learning that I will adopt is one developed by Albright and Hayes (1999a, to appear). This choice is not merely a matter of convenience, however, and in this section I will argue briefly why some other models are in fact unsuited to the task at hand.

Consider the problem of evaluating how well a model is able to perform on a particular mapping X→Y. One test that we might want to use is whether the model is generating the right output (Y) for each input (X). However, recall from the discussion in section 1.1 that there are potentially two different ways to generate outputs: either by retrieving the form from memory, or by synthesizing it using a set of generalizations (such as a grammar). If the model is evaluating its own performance on X→Y based on words that it already knows, then in some sense the easiest and most reliable strategy would be to memorize every form, so that the correct output can always be generated no matter how accurate the grammar is. This puts us in a bit of a bind; on the one hand, we want morphological models to be able to memorize output forms in order to handle irregulars, but we also want to be able to test the accuracy of the synthesized forms. In essence, we want to know how much of the input data the model considers irregular, and how much it could reproduce productively using just the grammar. Thus, our first desideratum is that a model must make a clear distinction between outputs that have been retrieved from memory and outputs that are synthesized productively.

A prominent class of models that do not meet this criterion are connectionist models (Rumelhart and McClelland 1987; MacWhinney and Leinbach 1991; Daugherty and Seidenberg 1994; Westermann 1997), since these intentionally blur the distinction between rote memorization and generalization. Even though it may be possible to use various techniques to diagnose to what extent a neural network is using generalization or word-specific memorization for each word, interpreting the end state of networks can be difficult and laborious (Clark and Karmiloff-Smith 1993; Hutchinson 1994). For this reason, connectionist models are not especially well

suited to the current problem.[1]

Another framework that often does not make a distinction between lexicalized and grammatically generated patterns is Optimality Theory. In fact, one of the original claims of OT is that it is *impossible* to memorize a lexical entry that is not transformed into the grammatically preferred pattern on the surface (*Richness of the Base*, Prince and Smolensky 1993, Smolensky 1996). Versions of OT that assume Richness of the Base are not especially well suited to the task at hand, and I will not use OT for the analyses of Latin or Lakhota phonology in the next two chapters. However, it must be noted that one could probably recast at least portions of these analyses into some modified version of OT, like that proposed by (Zuraw 2000), in which speakers can reason about constraints and lexical entries and allows us to distinguish between lexically listed forms and grammatically derived forms.

A second way in which we might try to diagnose the performance of grammars is by seeing if they derive their outputs unambiguously and confidently. If a particular sub-grammar X→Y is attempting to project from a form with many neutralizations to a form with more contrasts, there will be many cases in which the input is ambiguous, and there are two or more possible outputs. In order to evaluate this, we need a model that can generate multiple outputs for each input (where appropriate). In addition, it is helpful if the model can offer some form of confidence or well-formedness scores, allowing us to diagnose which forms the model prefers, how close the competition is, etc. There are several models that are similar in spirit to the Albright and Hayes minimal generalization model, but which do not provide such scores; these include Ling and Marinov's inductive decision tree approach (Ling and Marinov 1993), and Neuvel's whole word morphology model (Neuvel and Singh, 2001; Neuvel, to appear; Neuvel and Fulop, to appear).

One last desideratum is that the model should operate incrementally, and that it should be able to provide an analysis at every stage of the learning process. The reason is that if we want base identification to simplify the task of morphological learning, then we need to be able to do it early on, so we do not have to keep exploring all sub-grammars in all directions. Therefore, we must be able to identify the best base using a small subset of the data, and for this reason, we want grammars based on a few forms to be reasonably similar to grammars based on the whole data. (I will discuss the issue of size of learning set again at the end of section 3.2.) Many approaches to morphological learning do not meet this criterion, taking instead a top-down approach that tries to find the most general rules first. Examples of such models include the First Order Inductive Decision List (FOIDL) framework (Mooney and Califf 1996) along with other Inductive Logic Programming (ILP) approaches, Kazakov's "Naïve Theory of Morphology" employing genetic algorithms and ILP (Kazakov and Manandhar 2001), and Minimum Description Length (MDL) approaches (Brent, Murthy, and Lundberg 1995; Goldsmith 2001). Unfortunately, in order to have a good idea of what is the most general pattern in the language, one must have access to all of the forms in language; therefore, these systems tend to require the

[1] Hare and Elman (1995) discuss a similar problem, of evaluating whether connectionist networks are easier to train on some patterns than on others, as an explanation for why certain patterns in Old English verb classes were eliminated while others were extended. They trained their models in such a way that the model was still making some errors at the end of the training period, and these were fed into the next "generation"; thus, their models were not always able to reproduce the input data perfectly, but the model does not "know" that it is making an error. In the task of base identification, we want the model to be able to diagnose how often its grammar will produce the wrong output. The easiest way to do this is to let the model memorize the correct output for every single word, and then compare its memorized and synthesized outputs.

entire learning set in advance.[2] This requirement is incompatible with the goal of discovering early on which mappings to focus on, and would only be useful for making post-hoc decisions about which mappings yielded the cleanest generalizations.

The Albright and Hayes learning model satisfies all of these requirements, in ways that will be explained in the next section.

3.2 The minimal generalization learner

The minimal generalization learner takes a bottom-up (smallest generalization first), incremental approach to morphological learning. As its input, it takes ordered pairs of morphologically related words, such as the following:

(27) Sample input for minimal generalization

absolutive		ergative
lag	∼	*lagi*
basag	∼	*basagi*
sub	∼	*subi*
lot	∼	*loti*

Given the first pair of forms (*lag*,*lagi*), it constructs a morphological rule which could apply to the first to yield the second. It does this by locating the structural change (in this case, $\varnothing \rightarrow i$), and considering the entire rest of the word as the structural environment (*lag*__).[3] The result is shown in (28).

(28) $\varnothing \rightarrow$ i / lag__#

Generalization begins when a second pair of forms is encountered with the same structural change. For example, the second pair of forms in (27) also contains the structural change $\varnothing \rightarrow i$, and yields a word-specific rule with the structural environment *basag*__#. The generalization algorithm then compares the structural descriptions of these two rules, factoring them into shared portions and non-shared portions. The shared portion has two parts: a continuous string of shared segments strictly adjacent to the structural change, and the phonological features shared by the first segment in which the two forms differ. A new rule is formed by retaining the shared portions as the structural description, and converting the non-shared portions into variables. I will call such rules *generalized rules*, in contrast to word-specific rules like the one in (28).

[2] There is no reason why we could not apply a top-down system to progressive subsets of the data, but I am not aware of any attempts to do this, and it is not clear to me how the results would differ from simply using a bottom-up algorithm.

[3] In its current implementation, the parsing component of the learner is able to consider only structural changes which involve prefixation, suffixation, and infixation, but not simultaneous combinations of these. It can also consider structural environments which involve only contiguous strings on both sides of the change, plus featural descriptions of the segments immediately before and after the change. These limitations should not be an insurmountable problem for any of the languages which I propose to examine.

(29) Minimal generalization across two word-specific rules:

	change	residue	shared features	shared segments	change location
comparing A:	∅ → i /		l	ag	___#
with B:	∅ → i /	ba	s	ag	___#
yields C:	∅ → i /	X	[+cons, +cont, +cor, etc…]	ag	___#

="suffix -*i* after *ag*, preceded by any coronal continuant"

When the input pairs are quite similar, as is the case in (29), the generalized rule that results from this procedure is quite specific. However, when generalization is iterated across the entire lexicon, comparing forms with a variety of phonological shapes, the resulting rules can also be quite general. Continuing with the forms in (27), the pair (*sub,subi*) tells us that -*i* suffixation can occur not only after *ag*, but after the class of voiced non-coronal stops. The next pair (*lot,loti*) further shows us that -*i* can be suffixed not just after voiced non-coronal stops, but in fact after any stop.[4]

The forms in (27) show a consistent pattern—all of the words take the suffix -*i* in the absolutive. Real languages are rarely so regular, however; there are often several different processes competing to express the same morphological category. Suppose, for example, that in addition to the forms in (27), the target language had the following two forms:

(30) A competing suffix

absolutive		ergative
rag	∼	*ragu*
tip	∼	*tipu*

When the first of these forms is heard (*rag*∼*ragu*), it spawns its own word-specific rule, just as the previous words did:

(31) ∅ → u / rag___#

However, this structural change (∅ → *u*) is not shared by any of the words in (27); therefore, the generalization algorithm can not compare it to any other form to create a more general rule of -*u* affixation at this point.

Once the second -*u* pair (*tip* ∼ *tipu*) is learned, a word-specific rule of ∅ → *u* / *tip*___# is spawned, and then this is compared with the ∅ → *u* / *rag*___# rule to yield a more general -*u* affixation rule:

[4]Since minimal generalization always tries to find the tightest fit to the data seen so far, the exact natural classes which are hypothesized will depend somewhat on the feature set; if the features distinguish some type of stops other than voiced or voiceless (such as, say, aspirated), or has more places of articulation (such as pharyngeal or palatalized), then minimal generalization over this input set will contain feature specifications to exclude them. In addition, the types of generalizations that are possible will also depend on whether various features are treated as binary or privative. If [nasal] is privative, for example, it would be impossible to construct a generalization that applies just in the context of non-nasals. It is an open empirical question as to what the correct set of features is, and whether the feature combinations that are necessary to describe phonological phenomena are the same as those that are needed to group words for the purposes of morphology.

(32) $\varnothing \rightarrow u$ / $\begin{array}{c} \text{C} \\ \left[\begin{array}{l} -\text{continuant} \\ -\text{coronal} \\ -\text{lateral} \\ -\text{strident} \\ \text{etc.} \end{array}\right] \end{array}$ ___#

Thus we see that when there are competing processes present in the language, each serves as an independent structural description, and the phonological environments where each applies are considered separately.

What does a grammar constructed in this way look like? This type of generalization is not terribly economical; given an input file with thousands of forms, there are possibly tens of thousands of rules to consider. One strategy for rule acquisition is to throw out intermediate hypotheses, keeping only the "best" rule so far. For example, among the forms listed in (27), the good phonology student would realize immediately that the rule we want to learn is *-i* suffixation, and the intermediate hypothesis that *ag* was relevant in the environment can be discarded once we recognize the larger generalization. Indeed, this is what Pinker and Prince (1988) seem to have had in mind when they sketched a similar learning strategy. However, it is not always the case that the most general rule is the best one. Linguistic analyses often make use of intermediate generalizations; for instance, the best description of the distribution of the [t] allomorph of the English past tense suffix is that it occurs after voiceless obstruents, even though the existence of forms like *learnt*, *burnt*, or *dwelt* for some speakers means that strictly speaking, in those dialects it may actually occur after any consonant. (See Albright and Hayes (1999b) for further discussion of such cases, and a procedure for detecting them.)

The answer is that we need a more sophisticated metric of the "goodness" of rules. In the minimal generalization learner, this is done by making rules prove their effectiveness in the lexicon. In particular, rules are annotated with a *reliability* score, which is defined as the ratio of the number of input forms the rule derives correctly (its *hits*) over the number of forms it could potentially apply to (its *scope*):

(33) Definition of a rule's reliability:

$$\text{Reliability} = \frac{\text{number of forms the rule correctly derives (= } \textit{hits}\text{)}}{\text{number of forms included in the rule's structural description (= } \textit{scope}\text{)}}$$

For example, the generalized rule in (32) attempts to add *-u* after any non-coronal stop, yielding the correct result for the words in (30) (*ragu*, *tipu*), but the wrong results for the first three words in (27) (**lagu*, **basagu*, **subu*). Thus, the reliability of this rule is 2/5 = .4.

Once these reliability ratios have been calculated, they are then adjusted using confidence limit statistics, as suggested by Mikheev (1997), yielding a *confidence* score for the rule.[5] The

[5]The lower confidence limit of a reliaiblity ratio is calculated as follows: first, the reliability (probability) ratio, which we may call $\hat{p}$, is adjusted to avoid zeros in the number or denominator, yielding an adjusted value $\hat{p}^*$: $\hat{p}^* = \frac{hits+0.5}{scope+1}$. This adjusted value is then used to calculate an estimate of the true variance of the sample:

$$\text{estimate of variance} = \sqrt{\frac{\hat{p}^* \times (1-\hat{p}^*)}{n}}$$

This value is then used to calculate the lower confidence limit (π_{Lower}), at a particular confidence value α:

rationale for this is that we are more certain about occurrence rates for things that have had more opportunities to occur, and less certain about things that happen more rarely. The result is that unambitious rules which cover only a few forms are penalized substantially, while more ambitious rules are barely penalized at all; for example, a rule with a reliability of 5 out of 5 is downgraded from 1 to .825, while a rule with a reliability of 1000 out of 1000 is downgraded only a minuscule amount, to .999. Various experimental results have shown that confidence values calculated in this way correlate well with human intuitions about the relative well-formedness of morphological processes in different phonological environments (Prasada and Pinker 1993; Albright 1998; Albright, Andrade, and Hayes 2001; Albright and Hayes 2002).

We see that there are therefore two factors which contribute to the goodness of rules in this system: ambitiousness in including substantial numbers of forms in the structural description, and accuracy in deriving the correct outcomes for those forms. In order to collect all of these statistics, the learner needs to keep around all of the rules it hypothesizes, at least until some critical number of forms have been learned. Thus, grammars constructed in this manner can be enormous.

Although the grammar must be learned from existing forms, the strongest argument for the existence of grammar is that we are able to derive forms of words that we have never heard before. Grammars learned by minimal generalization can be used to derive novel forms, as follows: given a novel form of category X and a (possibly quite large) sub-grammar of rules for the mapping from category X to the target category Y, the sub-grammar is searched for all rules for which the novel form meets the structural description. For example, suppose we wanted to use the sub-grammar induced by the forms in (27) to derive the absolutive of a new word *mag*. By inspecting the sub-grammar, we would find that several different rules could apply to this form, including adding *-i* after *ag*, adding *-i* after voiced stops, and adding *-i* after stops in general. Although these rules all produce the same output (*magi*), each carries its own reliability score. When many rules all point to the same output, the output is assigned the confidence score of the best available rule.

When a language has competing structural changes (such as $\emptyset \rightarrow i$ and $\emptyset \rightarrow u$), each one gets to apply in this fashion. The result is a set of competing outputs, each assigned a reliability value. It is assumed that the output with the highest reliability is the one which is selected in a forced choice situation, and is the one which is judged best in an acceptability rating task.

The implementation of a minimal generalization learner presented by Albright and Hayes (1999a) has one last feature which is relevant here: a limited capacity to discover phonological rules. In order to do this, it makes use of prior knowledge of what sequences are phonotactically illegal in the language.[6] The way it discovers a rule is this: upon learning two competing structural changes (A→B and A→C), it checks to see whether the choice of B vs. C is motivated phonotactically. For example, given the English past tense suffixation rule $\emptyset \rightarrow d$ / ___# (as in

$$\pi_{Lower} = \hat{p}^* - z_{(1-\alpha)/2} \times \sqrt{\frac{\hat{p}^* \times (1-\hat{p}^*)}{n}}$$

The confidence value α ranges from $.5 < \alpha < 1$, and is a parameter of the model; the higher α is, the greater the penalty for smaller generalizations. In the simulations reported here, I will always assume an α of .75. The value z for a particular confidence level α is found by consulting a standard statistics look-up table.

[6] Providing the learner with this information ahead of time may seem like a cheat. However, there is increasing evidence that infants as young as 10 months old already have an idea of what is phonotactically legal in their language (Jusczyk, Friderici, Wessels, Svenkerud, and Jusczyk 1993; Friederici and Wessels 1993; Jusczyk, Luce, and Charles-Luce 1994); for a review, see Hayes (to appear).

rub∼*rub[d]*), and the competing suffixation rule ∅ → ə*d* / ___# (as in *need*∼*needed*), the learner tries applying ∅ → *d* to *need*, yielding the output *nee*[dd]. The learner also knows that [nidd] is unpronounceable in English, since there is an inviolable ban word-final geminates. Therefore, it can posit a phonological rule that epenthesizes [ə] in this environment (∅ → ə / *d*___*d*#. (For a more detailed description of the mechanisms it uses to do this, the reader is referred to Albright and Hayes, to appear). With this phonological rule in place, the simple *d*-suffixation rule (∅ → *d* / ___#) yields the correct output *need*[əd].

Phonological rules can help to improve the reliability of morphological rules, since they allow more forms to be derived correctly. The reason this is relevant is that there are two types of alternations which may exist within paradigms: those which are phonologically motivated, and those which are not. When alternations have a purely phonological explanation, they arguably should not contribute to the measure of how difficult a morphological mapping is. Some toy languages which involve phonological neutralizations are discussed in sections 3.4.1 and 3.4.2.

It should be clear from the description thus far that the minimal generalization learner has various limitations that restrict the types of processes it can discover and the types of generalizations it can form. The rules it posits, for example, involve just one change (A → B), so it would have difficulty learning morphological processes that involve multiple simultaneous changes (such as circumfixing, and so on). The rule format also requires that the context for a rule be strictly adjacent to the change (C___D, not C X___D), so it would not be able to discover non-local conditioning of a morphological alternation. Most of these limitations are irrelevant to the cases discussed here, and can safely be ignored. There is one limitation that is potentially relevant, however, and that is the fact that its input representations contain only strings of segments, with no place to store additional information such as prosodic information, grammatical gender, or other morphosyntactic distinctions. There is abundant evidence that such factors do play a role in conditioning morphological alternations, and it would be impossible to describe some languages (such as Latin) without taking them into account. Although the present version of the learner has no ability to calculate prosodic features of words or infer that it would be useful to consider factors like grammatical gender, it is nonetheless possible to provide it with these features and allow it to form generalizations with them. Since the input to the learner is strictly a string of segments, this is done (as a hack) by including in each input pair a "segment" that encodes whatever prosodic or morphosyntactic information the analyst wishes to provide to the learner (e.g., '1' for masculine, '2' for feminine, '3' for neuter, etc.). The learner can then generalize over these segments in the same way that it generalizes over phonological segments in the context (e.g., change X occurs only in masculine nouns, change Y only in feminine nouns, change Z in both, and so on). This strategy was employed in the Latin simulations described in chapter 4.

From this presentation of the minimal generalization approach, we can see that the model meets all of the criteria laid out in the previous section: it is specifically designed to learn grammars to project between ordered pairs of forms (present→past, 1sg→infinitive, etc.), so it is perfectly capable of exploring each of these sub-grammars independently. The grammars it learns can include many, competing changes (A→B, A→C, etc.), so it can potentially produce multiple outputs for each input. Each output is associated with the rule that derives it, so it is easy to tell whether an output has been derived by a generalized rule from the grammar, and when it has been derived using a word-specific rule (which could be seen as analogous to resorting to memorization). The confidence values of rules allow the model to assign predicted

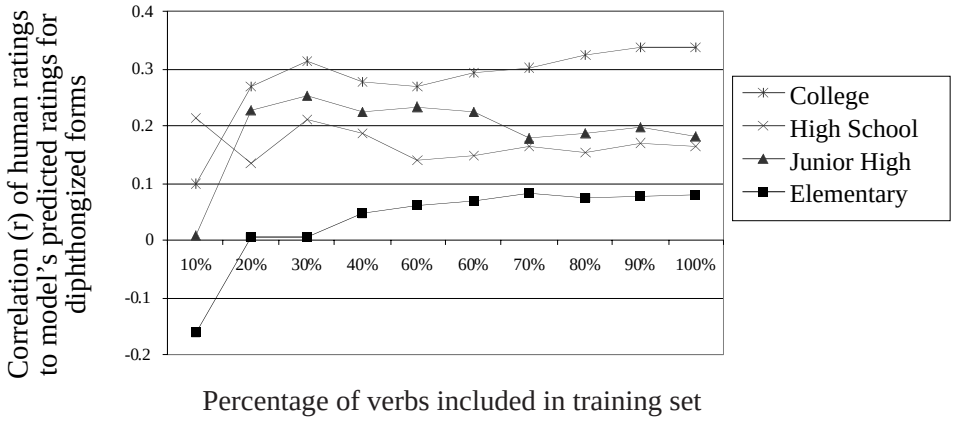


Figure 3.2: Time course of acquisition of the minimal generalization model (from Albright, Andrade, and Hayes 2001, p. 140)

well-formedness scores to its outputs. Finally, the model is incremental, and at every stage it has a working grammar which covers at least as many forms as it has heard. Furthermore, work on modeling various languages has shown that grammars learned by minimal generalization are quite "stable" throughout the training period—that is, they arrive at a basic analysis quickly, and the predictions for novel forms do not change much after that. For example, Albright, Andrade, and Hayes (2001) found in modeling the intuitions of adult Spanish speakers with varying degrees of education that adding more words does not make the predictions significantly better or worse, once 300 out of a training set of 1,698 verbs had been learned (in order of decreasing frequency). This result is reproduced in Figure 3.2.

This finding is also relevant for the task of base identification, because the goal here is to use an initial subset of the training data to make a decision that will reduce the amount of work needed to learn the remainder of the data. In principle, one might imagine that the initial batch of training data could be misleading, and the learner could be tricked into picking a suboptimal base because the initial batch of words was not representative. This is especially plausible because we know that irregular forms tend to be among the most frequent forms in the language (indeed, that is how their irregular status is preserved), and therefore learners would be exposed to disproportionately many irregulars. It turns out, however, that this fact makes the time course of learning *more* stable, not less. The reason is that once the learner has learned enough forms to identify the regular pattern as the predominant, productive one, the rest of the training set will continue to contain more and more regulars. In other words, as long as we do not restrict the learner to just the very top, most irregular batch of extremely frequent words, it is unlikely that it will be misled by a subset of the training data.

3.3 Metrics for sub-grammars

The focus of this thesis is not on *how* to learn morphological mappings, but rather on *which* morphological mappings to learn. The hypothesis is that we are searching for the sub-grammars

which are *easier*, or more effective. In this section, I will consider a variety of metrics for measuring the complexity of a sub-grammar.

Given the statistics which the minimal generalization learner collects about its rules, there are a variety of possible ways to measure the complexity of a sub-grammar, enumerated below.

Accuracy

A very common way to test the performance of machine-learned grammars is by testing it on its ability to reproduce the training set. In this case, given that the model also has word-specific rules at its disposal, we are really interested in whether the generalized (i.e., not word-specific) rules are able to reproduce the training set correctly without resorting to the word-specific (memorized) forms. The accuracy of a sub-grammar can be calculated as follows: first, use the subgrammar to derive outputs for all of the real words using only the generalized rules, and take the highest-scoring output for each word. Then, compare these outputs with the actual outputs, to see if the correct output can be synthesized by the subgrammar. The percentage of correct outputs is the *accuracy* of the sub-grammar.[7]

Mean confidence of rules in the grammar

Grammatical rules in the minimal generalization model have confidence scores attached to them, as described above. When the learner is able to find generalizations that include more forms, and have fewer exceptions, then the rules in the grammar will have higher confidence values. Therefore, we can use the *mean confidence* of the generalized rules in a sub-grammar as a metric of its complexity.

[7] This is similar in spirit to the Minimum Description Length (MDL) approach of comparing grammars (Rissanen 1986; Rissanen 1989; Rissanen and Ristad 1994), but with some crucial differences: The principle behind the MDL approach is to reduce the amount of information that must be memorized by formulating rules that can derive as much of the surface data as accurately as possible. The MDL principle is generally used to compare various possible grammars of the same data; in this case, the grammars have been constructed in exactly the same way, and we are comparing how susceptible different data was to grammatical description, when the grammatical construction procedure is held constant. I am not aware of other attempts to use MDL to compare the complexity of data instead of grammars, but it seems like a useful tool for this task because it explicitly takes into account the sizes of both the grammar and the data. It should be noted that there are some obstacles to using the MDL approach for the current problem, however. First, it is necessary to find a way of calculating the cost of a grammar. Cost is usually thought of as the size of the grammar, but this in itself can be difficult to quantify. (Also, as discussed above, there is a certain amount of evidence that people do not actually use what we might think are the most economical or efficient grammars possible, at least in terms of detailed information about particular environments). In addition, the cost of an MDL grammar includes a penalty for storing whole forms, so the only forms that are stored are irregulars that could not be synthesized by any more economical means. Limiting storage to irregulars is certainly a traditional idea in theoretical linguistics, and has been the focus of a good deal of research in the dual mechanism approach—see, e.g., Pinker 1999, Clahsen 1999. However, various results in recent years have shown that regulars, too, may sometimes be stored (Stemberger and MacWhinney 1986; Sereno and Jongman 1997; Baayen, Dijkstra, and Schreuder 1997; Eddington and Lestrade 2002). Pinker and Prince themselves admit that regulars cannot be excluded categorically from the lexicon, even if the same forms could also have been synthesized productively using the grammar (Pinker and Prince 1994, p. 331). Therefore, I merely mention the parallel with MDL in passing, but I am not convinced that adopting one of the current MDL approaches to morphology would provide an accurate model of human learning.

Mean confidence of winning outputs

Another intuitive idea is that grammars are better when they produce their results unambiguously and confidently. One way to evaluate this is to use the generalized rules to wug-test the sub-grammar on all of the forms known so far, as above, and then collect the winning outputs and calculate their *mean confidence* scores. (Recall that the confidence score of an output is equal to the confidence score of the best rule in the grammar that could derive that output.) A higher mean confidence means that the outputs are supported by more data, and more consistent data.

Average winning margin

In addition to being confident about outputs, we would like to be sure that there are not other, almost as good outputs that we have to agonize over. Therefore, we may be interested in the *average winning margin* of outputs, as a measure of the degree of angst incurred when using a sub-grammar. As before, we can wug-test the grammar on the known words, collecting the outputs with the highest confidence. Then, for each winning output, we can compare its confidence with the confidence of the next best distinct output—this is the margin by which the best output won. We can then average these margins, to get an estimate of how often the sub-grammar is forcing us to agonize over close competitions between two outputs; a low average margin means that the grammar is often deciding between close competitors.

Summary

These are just a few of the many metrics that could be used to measure the complexity of a sub-grammar, but they should provide a starting point of testing the approach. With these metrics in hand, the next logical step is to see how they work on a few cases. I will begin with a few synthetic languages modeled on typical patterns found in introductory problem sets, and then consider two more realistic cases.

3.4 Results for synthetic languages

3.4.1 Synthetic language 1: Neutralization in suffixed forms

It is always a good idea to start testing models on very small and schematic cases. I will keep the first few cases simple by tackling only the problem of comparing two directions along the same mapping (i.e., *form1* → *form2* vs. *form2* → *form1*).

The first case to consider is one in which there is phonological neutralization in a suffixed form.[8] For example, consider a language with palatalization of $k \rightarrow \widehat{tʃ}$ before *i*, as in (34). In such a language, underlying *k* and $\widehat{tʃ}$ are neutralized in this context— a pattern which is found in Fe'fe'-Bamileke (Hyman 1975, p. 70) and many other languages.

[8] I will try to avoid using the term *derived form*, since it presupposes that the suffixed form is built on the unsuffixed form, and the idea here is to diagnose when precisely the opposite is true.

(34) Neutralize I: Palatalization in suffixed forms

a. Stems ending in segments other than *k*

absolutive		ergative
dap	∼	*dapi*
lot	∼	*loti*
gub	∼	*gubi*
satʃ	∼	*satʃi*
rutʃ	∼	*rutʃi*
lag	∼	*lagi*
ban	∼	*bani*
yul	∼	*yuli*

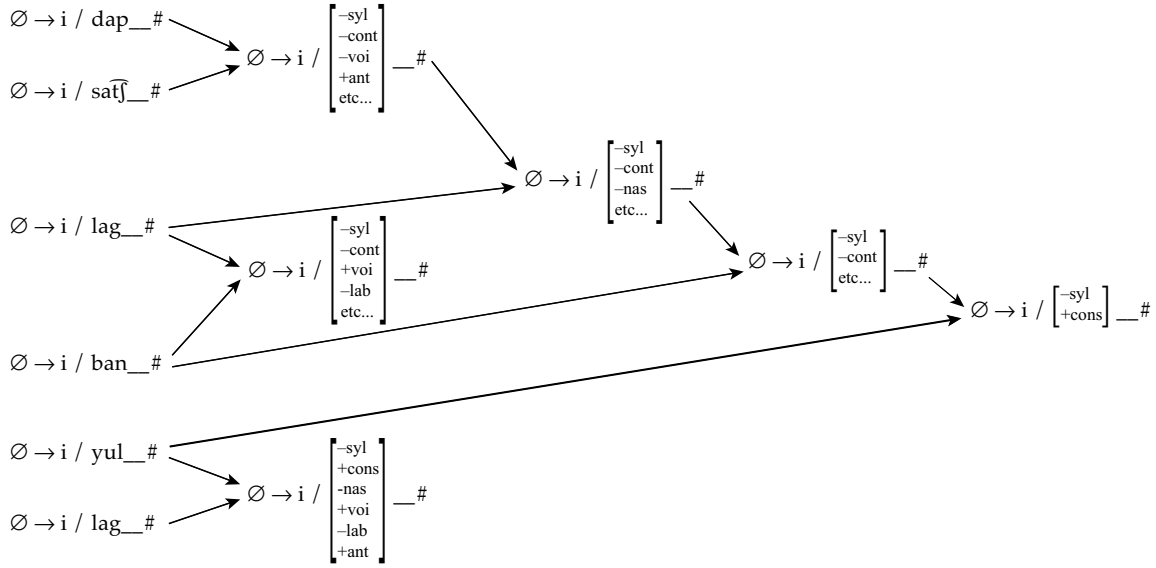
b. Stems ending in *k*

absolutive		ergative
ʔak	∼	*ʔatʃi*
muk	∼	*mutʃi*
lok	∼	*lotʃi*

In this case, the correct answer to the problem set is that the base, or UR, is the unsuffixed (absolutive) form, since this is the form which contains the most distinctive information about the word (in particular, the /k/ vs. /tʃ/ distinction). Therefore, we hope that the metrics suggested above are able to choose the absolutive→ergative mapping as the better one.

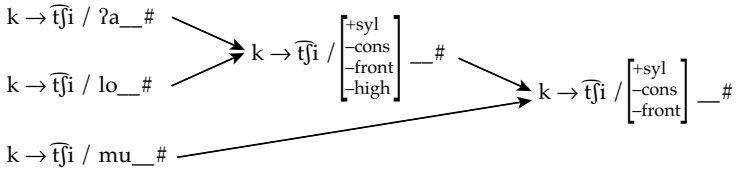
It is instructive to outline how the minimal generalization learner will handle this language, which involves a strictly phonological alternation. Going in the ergative to absolutive direction, it generalizes over the forms in (34a) to find the rule ∅ →*i* after consonants, as shown in (35). (The full set of generalizations, along with the details of the calculation of the metrics for this synthetic language, are given in appendix A).

(35) Minimal generalization over the forms in (34a):



The *k*-final roots in (34b) are somewhat more complicated: the first hypothesis which the learner explores is that there is a substitutive morphological process (*k*→*t͡ʃi*):

(36) Minimal generalization over the forms in (34b):



The learner does not stop with this parochial generalizations, however; it also employs its limited ability to discover phonological rules (p. 41). The generalizations in (35) produced a ∅ → *i* rule. Although this rule by itself cannot produce the correct output for words like *ʔak* or *muk*, the learner tries to discover phonological rules to convert the incorrect outputs (**ʔaki*, **muki*) into the correct ones (*ʔat͡ʃi*, *mut͡ʃi*). It first scans **ʔaki* and **muki*, and notes that they contains an illegal *ki* sequence (furnished ahead of time, as described above). It then compares **ʔaki* with *ʔat͡ʃi*, and posits a phonological rule that fixes the illegal sequence to make it match the observed outcome: *k* → *t͡ʃ* / ___*i*. This phonological rule is sufficient to allow simple *-i* suffixation to derive the forms in (34b) correctly as well. Therefore, the suffixation process applies with 100% reliability in the absolutive→ergative direction.

The grammar for the ergative→absolutive direction is not as clean. For the forms in (34a), minimal generalization results in a rule stripping off *-i* after a consonant. As before, the forms in (34b) are considered first as substitutive morphology (*t͡ʃi*→*k*). However in this case, there is no surface-true phonological rule which can allow *ʔat͡ʃi*→*ʔak* to be viewed as simply removing the final *-i*. Therefore, each of the forms in (34b) must be counted as an exception to the *i* → ∅ rule, and must be handled by a competing (substitutive) process. Thus, we can see that the reliability of rules in this direction will be lower, reflecting the unpredictability of "de-palatalization".

In fact, the absolutive → ergative mapping comes out better than the ergative → absolutive mapping for *all* of the metrics:

(37) Comparison of metrics for the Neutralize I language

	abs→*erg*	*erg*→*abs*
avg. confidence of generalized rules	.762	.585
avg. confidence of best output	.916	.743
accuracy (=ratio correct)	100%	73%
avg. winning margin	1.000	.552

The metric which turns out to be clearest in this example (and will be for all of the synthetic languages discussed here) is the average winning margin, which is designed to measure how much uncertainty is incurred by using the grammar. However, all of the metrics agree on the choice of base, and in general they turn out to be highly correlated in all of the cases that have been examined thus far.

3.4.2 Synthetic language 2: Neutralization in suffixless forms

The next step is to show that the same principles can be used to identify a suffixed form as the base (or UR), when there is a neutralization which affects just the unsuffixed form. This is often caused by final devoicing, or other processes that affect word-final segments.

A synthetic language with this property is given in (38):

(38) Final devoicing in suffixless forms

a. Underlyingly voiceless C's

absolutive		ergative
tak	~	*taki*
dap	~	*dapi*
yup	~	*yupi*
lot	~	*loti*
mut	~	*muti*

b. Underlyingly voiced C's

absolutive		ergative
lak	~	*lagi*
bak	~	*bagi*
sak	~	*sagi*
gup	~	*gubi*
trep	~	*trebi*
flap	~	*flabi*
fut	~	*fudi*
sat	~	*sadi*

For this language, the correct problem set answer is that the UR is the absolutive form minus the *-i* suffix. In a model of morphology which employs only surface form-to-surface form mappings, the correct answer is that the absolutive is formed from the ergative by removing the *-i* suffix and applying final devoicing; it could not be done in the opposite direction, because we would have to guess for each word what the voicing of the consonant must be.

It should not be difficult to see why the same factors that made the ergative → absolutive mapping unreliable in the language with intervocalic voicing should make the absolutive → ergative mapping unreliable in this language. Unsurprisingly, the metrics come out exactly reversed, favoring the ergative as the base:

(39) Comparison of metrics for the Neutralize II language

	abs→erg	*erg→abs*
avg. confidence of generalized rules	.382	.659
avg. confidence of best output	.450	.928
accuracy (=ratio correct)	62%	100%
avg. winning margin	.136	.928

3.4.3 Synthetic language 3: Morphological neutralization

The first two synthetic languages have shown the role that phonological rules can play in making the grammar for a mapping "cleaner" in one direction than in the other. In these cases, the choice of a base for the morphological rule is essentially the same problem as the choice of the UR for phonological purposes.

In many cases, however, there is no phonological reason why one form should be considered the morphological base. One situation where this could happen is when there is "morphological

neutralization"—two or more morphological classes share the same marking in a particular inflection. In the following example, the two classes of nouns (which are labelled *o*-stems and *a*-stems in honor of the vocalism in their absolutive forms) have the same lack of marking in their absolutive forms. (I will assume for the purposes of this discussion that an apocope analysis is ruled out by the existence of other words in the language with final unstressed vowels.)

(40) Neutralize III: morphological neutralization

a. *o*-stems

absolutive		ergative
lag	∼	*lagos*
ʔak	∼	*akos*
gub	∼	*gubos*
dap	∼	*dapos*
fud	∼	*fudos*
lot	∼	*lotos*
ran	∼	*ranos*

b. *a*-stems

absolutive		ergative
sag	∼	*sagas*
bak	∼	*bakas*
reb	∼	*rebas*
yup	∼	*yupas*
sad	∼	*sadas*
mut	∼	*mutas*

In this case, the correct "textbook' answer is not so clear. Since there is no phonological process involved, it is perhaps better to speak of the *stem* (or *root*) rather than the UR. One analysis would be to say that the stem is the same as the absolutive form, and there are two classes of nouns, distinguished by their endings (*-os* vs. *-as*) in the ergative. An alternate account would be to say that the stem is the ergative form minus the *-s* (*lago-*, *ʔako-*, *saga-*, etc.) and the case forms are the result of dropping the theme vowel or adding an *-s*, respectively. (Both approaches have been common in the analysis of noun and verb classes in Indo-European languages, among other places.)

From the point of view of having to map one form from the other (rather than positing an abstract stem), we see that the absolutive → ergative mapping forces us to provide an (unknown) vowel, whereas the ergative → absolutive direction merely involves taking off a well-defined amount of phonological material. Notice that since there is no phonological motivation for the alternation between *-as* and *-os*, the two processes cannot be unified by any overarching rule; the most general rules in either direction will be to add/remove *-as* and to add/remove *-os*.[9]

Even without phonology, we see that almost all of the metrics considered here agree that the ("backwards") mapping from ergative to absolutive is the better one:

[9]This points out a limitation of the syntax of morphological rules used by the minimal generalization learner; if we could use features in the structural change as well as in the environment, then the ergative → absolutive direction could be captured by a single rule peeling off a vowel plus *s*. This could turn out to be important in some especially close case, but is not necessary here.

(41) Comparison of metrics for the Neutralize III language

	abs→erg	*erg→abs*
avg. confidence of generalized rules	.353	.673
avg. confidence of best output	.439	.862
accuracy (=ratio correct)	54%	100%
avg. winning margin	.071	.862

3.4.4 Synthetic language 4: Lexical exceptions

One last factor which could lead one mapping to be better than another is the existence of lexical exceptions in a particular part of the paradigm. There are, of course, many different patterns of exceptions, but what I will consider here is the case where there is one general ("regular") process, and a few isolated words which take a different ("irregular") pattern.

A language with this property is given in (42):

(42) Exceptions I: a few lexical exceptions with a completely different marking

a. "Regulars" (suffix *-os*)

absolutive		ergative
lag	∼	*lagos*
ʔak	∼	*akos*
gub	∼	*gubos*
dap	∼	*dapos*
fud	∼	*fudos*
lot	∼	*lotos*
sag	∼	*sagos*
bak	∼	*bakos*
reb	∼	*rebos*
yup	∼	*yupos*

b. Lexical exceptions

absolutive		ergative
sad	∼	*sed*
mut	∼	*myt*

In this language, the generalized rule of adding *-os* after stops works for almost all of the forms (10/12), but the other two forms must be handled by separate ablaut rules (*a→e* and *u→y*). The minimal generalization learner was in fact designed to learn general patterns (like *-os* suffixation) in the face of lexical exceptions, so the focus until now has always been on this direction. However, it is interesting to see whether this is the direction which would in fact be picked as the optimal mapping according to the criteria employed here.

Looking through the forms in (42), we can see that it would actually be easier to guess the absolutive given the ergative than the other way around. Any absolutive form with an *a* in it should be eligible for ablaut (e.g., *bak→*bek*). An ergative form without *-os*, on the other hand, could only be lacking *-os* because it is irregular, and an ergative form with *-os* is always regular.[10]

[10]This is *not* always the case in real languages—for example, English and German have "mixed" irregulars like *keep∼kept* or *brennen∼brannten*.

The comparison metrics reflect this, indicating that the absolutive would be the better base for this language:[11]

(43) Comparison of metrics for the Exceptions I language

	erg→abs	*abs→erg*
avg. confidence of generalized rules	.631	.738
avg. confidence of best output	.834	.908
accuracy (=ratio correct)	83%	83%
avg. winning margin	.834	.908

This leads to the rather counter-intuitive result that the best base in a language with lexical exceptions may often turn out to be the form in which the exceptionality is clearest. For example, in the case of English, it is much easier to tell if a verb has an irregular past tense by looking directly at the past than by looking at the present: if the verb ends in anything other than *-t*, *-d*, or *-ɘd*, it is guaranteed to be irregular (*sung*, *rang*). Furthermore, it is guaranteed to be irregular it ends in *-t* after a vowel or sonorant (*beat*, *wrote*, *caught*, *dwelt*), or *-d* after a lax vowel (*shed*). The only really ambiguous cases are those like *kept* or *rode* which have the voicing-appropriate dental suffix and would be phonologically legal without the final *t* or *d*. However, such cases constitute only a minority of the irregular past tense forms of English, so the problem would be reduced greatly by choosing the past tense as the base. I believe that the reason why English speakers do not seem to do this has to do with the enormous frequency difference between the present and the past tense. I will return to this question in more detail in section 6.2.1, addressing the question of why learners might sometimes be driven to use mappings with many exceptions, even if they could have been avoided using another mapping.

There are, of course, many other patterns of neutralizations and exceptions in languages, many of them more difficult than those exemplified by the synthetic languages in this section. In many cases, neutralizations and exceptions are sprinkled throughout the entire paradigm, so it can be difficult to identify a single slot in the paradigm as the clear best choice, as we could in the synthetic languages. Often, it is necessary to compare the various neutralizations and exceptions, considering also how many lexical items are affected by each, in order to decide which are the least serious. In the next chapter, I will consider a more complicated and realistic example, from Latin noun paradigms. I will show that although neutralizations affected every part of the paradigm, these neutralizations were more serious in the nominative than in any other form. The model, predicts, therefore, that a non-nominative form should be chosen in base, and subsequent historical changes show that this prediction seems to be true.

[11] The exact computation of the average winning margin depends on the treatment of word-specific rules, since this language requires that we sometimes derive forms using these rules. Intuitively, we might think that our confidence in a word-specific rule should be 1, since that rule does a single job and it does it unambiguously. However, the adjustment using confidence limits which is employed here does not allow this, since the confidence limit for an event which occurs just once is undefined. In calculating the average winning margin for this example, I have simply excluded the two cases where the word-specific rule was used, and averaged the margin for the remaining ten cases.