"Generalize and Sift" as a Model of Inflection Acquisition

by

Raymond A. Molnar

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2001

Author . . . .
Department of Electrical Engineering and Computer Science
August 31, 2001

Certified by . . . .
Patrick Henry Winston
Ford Professor of Artificial Intelligence and Computer Science
Thesis Supervisor

Accepted by . . . .
Arthur C. Smith
Chairman, Department Committee on Graduate Students

“Generalize and Sift” as a Model of Inflection Acquisition

by

Raymond A. Molnar

Submitted to the Department of Electrical Engineering and Computer Science
on August 31, 2001, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Electrical Engineering and Computer Science

Abstract

In this thesis, I propose the “Generalize and Sift” model for the acquisition of inflection in natural language. This model uses a two step learning process along with probabilistic selection when there is uncertainty. To learn, the model first creates rules in a specific to general search over the phonemic feature space of words, allowing for a fast first degree approximation to the language’s rules. Then, in cases where rules overlap, the model weights the applicable rules and adjusts these weights according to observed data, eventually converging on the correct rule. I have implemented this model in a computer program and run simulations with data mimicking the words a child might hear as he learns inflections. The implementation displays behavior very similar to those of children: the same learning curve, similar mistakes while learning, and equivalent behavior once it reaches “adulthood.”

Thesis Supervisor: Patrick Henry Winston
Title: Ford Professor of Artificial Intelligence and Computer Science

Acknowledgments

First, I would like to thank my parents and brother. There really is no way for me to express my gratitude for everything they have done for me through the years. I would also like to thank my advisor, Patrick Winston, for allowing me the freedom to work on what I wanted, the way I wanted. Then I would like to thank my friends (especially Steve) who spent their nights reviewing drafts of this document, helping me to distill it into an intelligible form. Finally, I want to thank my other colleagues, friends, professors, and family for supporting me through this and all of my endeavors.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The field of language acquisition holds special importance because humans are the only species that can understand and generate language. Because of this, the mechanisms used to learn language must be unique to humans, and are therefore a glimpse into how humans differ from other animals. Since we cannot yet peer inside a child's head to see how he thinks, discovery in this field has inherent difficulties. Additionally, while there are thousands of different languages distributed across the world, any child born into a culture speaking any of them will correctly learn that language. To add complexity, children learn language at an alarmingly fast rate that complicates any theory of language acquisition. Because of this, most people focus on a small part of the process.

1.1 Problem Statement

One area of language learning that has received a great deal of attention over the last 15 years is that of learning the past tense in English. While it is just one small section of one language, it is an appropriate example because it contains many of the quirks seen in all languages. In this way, it represents some of the larger issues seen in language and learning research.

Additionally, because of its prominence, a large amount of linguistic data has been collected on the English past tense, making it fertile ground for modeling and testing.

As a result, many models and implementations of these models have been set forth, setting standards of comparison. This thesis aims to surpass these previous attempts with a more cognitively plausible and accurate model of learning the English past tense.

However, because the English past tense is only one small part of a larger area of language, this work also attempts to keep a view of the larger picture. Narrowing the problem to such a small area runs the risk of creating a model that only works on that small area. Instead, this thesis approaches the larger problem of learning general word inflection and class change. While the English past tense is a good place to start, because it allows the research to relate to previous work, any successful model should also be easily extended to other types of inflection, and to inflection in other languages.

Lastly, because a model is only as good as its results, this thesis attempts a computational implementation of the model presented in order to test its claims and verify its conformity to that of human behavior. This allows the model to be compared, on equal ground, not only to previous work, but also to data gathered about humans as they learn language.

1.2 Goals

The central goal of this research is to create and test a computational model that can learn the correct transformations of words as they change parts of speech. To do this, it must contain the ability to perform the following functions:

- Learn Quickly - Humans learn rules of language based on only a very few examples. They can usually learn a specific fact after seeing it only 1 or 2 times and create accurate rules with only 5 to 10 examples.

- Generalize Effectively - When shown unfamiliar words, humans can immediately transform the word into other forms. Additionally, as children learn how to change words, they apply these transformation to words that they have just learned.

- Use Only Positive Examples - Children do not receive much evidence of incorrect usage. Most of what they hear is correct usage, and even the small amount of negative feedback is given with mixed signals and shown to be ignored by them.

- Contain Robust Methods - Depending on where a child is born, he may have to learn an arbitrary language and dialect. In order to do that, his learning methods must be robust enough to account for all of the possible languages of the world as well as variations in local dialects. These methods must also allow for different types of inflectional change such as the many tense formations, pluralization, and word class change (ie. from noun to verb or from adjective to adverb).

In addition to the features noted above, Chapter 2 explains some more detailed behaviors that a model must be able to account for.

1.3 Approach

The central approach taken by this research is that learning word transformation is a two step process. The first step hypothesizes rules that cause observed changes, while the second step decides which of the possible rules apply to particular words. Separately each of these processes uses simple mechanisms, but together the overall system creates complex behavior that is similar to that of a child learning the same inflectional transformations.

After seeing words in both the uninflected and inflected form, the first step, *generalization*, postulates rules that change the word from one form into the other. At first, this change only applies to the particular words that have been seen. As more words are seen that exhibit the same change, the rule opens up to include the new examples in a pattern that is basically a specific to general search through the space of all possible words. This method takes advantage of the fact that words that change in the same way are generally clustered in an otherwise sparse phonological space. While it does end up covering more words than it should and also produces overlapping rules, these problems are corrected in the second stage of learning.

This second learning step, *sifting*, only comes into play when the *generalization* process creates more than one rule that can apply to a particular word. For every word that has conflicting rules, the model assigns weights to the rules and then adjusts the weights according to the examples it has seen. By doing this, it slowly discovers the correct rule for that word. In addition, this process cancels out noisy data as well as allowing for subtle differences seen from individual to individual.

Speech production follows directly from the sifting process. When the model is required to create an inflected form, it takes the root and applies the appropriate rule. When there are conflicting rules, it merely chooses randomly between the applicable rules based upon their current weights.

This model has been implemented on a computer and tested with data reflecting the words children hear while they learn inflection. The system produces learning and speech patterns that are consistent with, and virtually indistinguishable from those of natural speakers as they learn language. Specifically, while learning, it shows the same learning curve and types of errors as humans. Furthermore, once it completes learning, it generalizes novel words the same way humans do.

1.4 Structure of Thesis

This thesis is arranged in several sections. Chapter 2 gives the linguistic background of the problem, describing its finer issues and the linguistic foundation for the model presented. Chapter 3 describes some of the previous models and, where appropriate, implementations of these models. Chapter 4 lays out the "Generalize and Sift" learning model in detail. Chapter 5 explains an implementation of this model and the results of comparing this implementation's output with that of children. Chapter 6 concludes with a summary of the model and suggests further research in this area.

Chapter 2

Linguistic background

Over the course of language study, various linguistic structures have been proposed in order to account for regularities seen throughout languages. The first part of this chapter will describe some of these smaller structures that words are made up of and their relation to inflection.

Additionally, a large portion of this research has been geared toward language acquisition and the specific problem of inflection acquisition. As such, certain trends have been found across speakers of a particular language as well as across many different languages. The second half of this chapter explains the details of these human behaviors that will become the basis of the acquisition models. It goes into some detail on the English past tense and then explains some of the differences seen in other inflections.

2.1 Underlying Word Structure

One of the keys to learning is to exploit the internal structure of the information being learned. Delving deeper into the task usually brings out regularities and patterns that make learning easier. Words are no exception to this. Underlying words are two deeper levels of detail that are useful in learning parts of language and specifically inflectional changes.

2.1.1 Phonemes

The first level smaller than words is the level of phonemes. Phonemes are the elements of pronunciation that all words are made up of. They roughly correlate to letters in written text but instead are the spoken sounds that go along with the letters. Because some letters can stand for more than one phoneme, they are usually more of them than letters in a language. For instance, while English contains only 26 letters, it has 40 phonemes.

Phonemes are important because they can be considered the "letters" of spoken language. While written words often have ambiguous pronunciation, phonemic transcriptions do not. For instance, the words `gel` and `geld` appear almost the same in written text, but their pronunciations differ dramatically. Phonemically, these words are written as `/jh eh l/` and `/g eh l d/`, displaying the differences in the words' sounds.

Different spellings can also have the same sound, though, as exhibited by the words `eke`, `leak`, and `seek`. Phonemically, these words are `/iy k/`, `/l iy k/`, and `/s iy k/`. In these words, the phonemic transcription captures the regularity in the words that would go unnoticed in the written text.

Phonemes are useful from a learning point of view for one main reason. They are what children actually learn language from, not spellings. Children learn how to speak by listening to other people talk. Because, as they learn, they are exposed to only the sounds of the words, their learning must be based on words' sounds.

2.1.2 Distinctive Features

One level deeper than phonemes lie distinctive features, as seen in Table 2.1. Distinctive features capture the way that the phonemes are actually produced by the mouth. Two simple examples of features are the LABIAL feature, representing that a person's lips are involved in producing the sound, and the NASAL feature, representing the fact that sound also goes through your nose, not just your mouth. Examples of phonemes that have these two specific features are labials, `p, b, m, f, v`, & `w` and nasals `m,`

	s	ae	m	p	el
syllabic	-	+	-	-	+
consonantal	+	-	+	+	+
sonorant	-	-	+	-	+
voiced	-	-	+	-	+
...					
back	-	-	-	-	-
low	-	+	-	-	-

Table 2.1: *The phonemes in the word "sample" and some of each phoneme's distinctive feature values.*

`n`, & `ng`. A complete list of phonemes and features can be found in Appendix A.

Distinctive features are important because grammatical rules often depend on a single feature. As explained below, inflection is dependent on the VOICED feature of the phoneme at the end of the word. This feature corresponds to the use of the vocal chords when pronouncing the phoneme. For instance, the phonemes `b` and `p` differ by only the VOICED feature; `b` is voiced, while `p` is not. As such, they receive inflections differently.

Without looking at the level of distinctive features, phonemes would have to be grouped together in an arbitrary manner in order to explain regularities in linguistic rules. Distinctive features allow them to be grouped based on similarities in the way the sounds are produced, grounding the groups in the physical world. This makes distinctive features both an accurate and appropriate tool to aid in learning.

2.1.3 Sparse Space

As explained earlier, words can be defined by the phonemes that make them up. Phonemes can then be defined by the distinctive features that make them up. After grouping words in this way, they can be arranged in a high dimensional space spanning all the combinations of phonemes. When words are mapped into this space, the actual words in a language cover only a small part of all the possibilities.

For example, in English, there cannot be any words that have certain consonantal clusters. Because of this, any part of the space that corresponds to words with a `/k g/` in them will be empty. Similar impossible sound clusters will create many areas of the space that are empty, or nearly so.

The sparseness of this space is very important for the learning of language. By exploiting the natural clusters of words, children can generalize quickly. Words with similar phonemes, as determined by their distinctive features, are often treated similarly. Therefore, once a person has figured out a rule, he can apply it to words in the same cluster without explicit examples of its use on the other words.

2.2 Problem Details

While much of the attention on inflection has been directed at the English past tense, a good model of inflection acquisition should account for the behaviors seen in all parts of English inflections as well as the inflections of other languages. While many of these exhibit similar properties, each has a few of its own quirks. By showing the variety of ways that inflection can occur, these quirks give us a glimpse into the underlying mechanisms of acquisition.

2.2.1 English Past Tense

The English past tense is by far the most studied of all inflections. While most linguists believe that it is a fairly simple problem compared to other parts of grammar and even other inflections, it does exhibit most of the properties that make acquisition interesting.

Regular Formation

As we all learned in grade school, the main way to make a word into the past tense is to add -ed. This accounts for most of the verbs in the English language, such as `walk => walked`, `cause => caused`, and even the rare `copulate => copulated`. This school-taught rule covers over 90% of the words in English.

Despite this rule's simple appearance, it is not as straightforward as it seems. When you add the letters -ed, you actually could be adding one of three different sounds to the end of the spoken word. The first possible sound is the `d` phoneme that occurs in cases such as `cause => caused` and `happen => happened`. But adding -ed could instead be adding a `t` phoneme, such as in the words `walk => walked` and `laugh => laughed`. Lastly, the rule could actually be adding the new syllable `ed`. Some examples of this are `mount => mounted` and the aforementioned `copulate => copulated`.

The choice between these three options, however, is not an arbitrary decision. Each of the three cases has a distinct class of words that it applies to. The easiest of these to see is when you add the entire syllable `ed`. This occurs when the word ends in either a `t` or `d` phoneme. Thus, it happens for `joust => jousted` and `participate => participated`, but not to `profess => professed` or `view => viewed`.

The other two versions of the add -ed rule split the rest of the verbs in half, based on the word's final phoneme. If that phoneme carries the VOICED feature then it gets the voiced `d` phoneme added to it, while phonemes that are unvoiced get the unvoiced `t` phoneme added to them. Words such as `vow => vowed` and `absorb => absorbed` show the voiced version of the rule while `check => checked` and `glimpse => glimpsed` show the unvoiced version. To confirm that voicing is in fact the important feature in this distinction, we can look at a pair of words that differ only by the VOICED feature in the last phoneme of the word. One example of a pair of words like this is `cab` and `cap`. When inflecting these words into the past tense, they turn into, `cabbed` and `capped`, or phonemically `/k ae b d/` and `/k ae p t/`. This confirms that `cab` gets the feature matching voiced `d` phoneme while the unvoiced `cap` gets its feature match, the `t` phoneme.

Irregular Formation

Although these three word additions account for most of the words in English, they do not account for all of them. What they leave behind are commonly known as the irregular verbs, which require some change other than the standard add -ed rule.

Some examples of these words are `sing => sang`, `hit => hit`, and `feel => felt`.

Of the approximately 200 irregular verbs in the English language, almost all of them fall into one of 10-15 classes of words whose members change in the same way when forming the past tense. One example class is the words that shorten their vowel as they go to the past tense. This includes the words `bleed => bled`, `feed => fed`, and `meet => met`. Another example is the class of verbs that do not change at all. This class includes words such as `bit => bit`, `bid => bid`, and `cut => cut`.

Like the three versions of the default rule, though, the words in these classes are not randomly chosen. Instead, the words in each class cluster around a particular area in the phonological space. For instance, the class containing `sing => sang` and `drink => drank` contains only words that end in a subset of phonemes. These words are `ring, sing, spring, drink, shrink, sink, stink, swim`, and `begin`. Each class of irregulars cuts out a different phonological space for itself in which it applies.

One way to look at the irregular verbs is to see them as exceptions to the regular rule. By doing this, it should come as no surprise that each of these irregular classes has its own exceptions, which lie inside the space it has cut out. For instance, the irregulars `blow => blew`, `grow => grew` and `know => knew` stake out a vowel change for words that end in `-ow`. However, within this space, there is an exception to this irregular class, `show => showed`, which does not follow the irregular change. Every irregular class has exceptions like this, some like `show` that make the regular change and some like `bring => brought` that make a different irregular change.

Other Considerations

One important ability in the system of English past tense is the ease of adding new words to the vocabulary. When someone creates a new word, people do not have to wait around until they are told what its past tense is. Everyone English speaker knows based on the system in place. Some recent additions to English include the words `mosh => moshed` and `dis => dissed`, of which both happen to take the regular inflection. But not every novel word follows the regular pattern. When taught the novel words `gling` and `bing`, native English speakers produced the past tense forms `glang` and `bang`, showing that the irregular verb classes can also attract new members[3].

Another interesting property of the system is that the inflected forms are not set in stone. There are three places where this can be seen. The first is in dialectical variants, such as `bring => brang`, that are acceptable in some parts of the country, conflicting with the standard `bring => brought` change. A second place where this shows up is through historical change. One good example of this is `sneak => snuck`, which recently changed from `sneak => sneaked` in Old English. The last place where the fluidity of inflection can be seen is in a few words that have multiple, equally acceptable forms. An example of this is the pair `dream => dreamed` and `dream => dreampt`, where both of these inflected forms are considered correct. Of particular note is that these three types of ambiguity in the inflected form can shift either from the regular class to the irregular or vice versa. Like novel word introduction, they are not restricted to only the regular rule.

Acquisition

From this, it appears that English has a complex mix of verbs that follow the regular rule (all three varieties of it), exceptions to that rule, which create their own smaller rule, and exceptions to the exceptions, which can either fall back into the regular rule or stake out their own area. Somehow, though, every child exposed to the system does eventually learn the correct adult form of each verb. A good way to shed light on this process is to look at children's behavior as they go through the process of learning.

Before going into detail about the past tense inflection, though, a larger trend in all of language acquisition stands out. Children learn based on only positive examples of the language. The data that children learn from is the adult speech around them, mostly their parents', which is made up of only correct examples of the language. On the rare occasion that a child produces an incorrect form and is corrected, this correction seems to be ignored by the child[21]. Because noted negative examples are important to most learning[33], and language is such a complicated process, this fact makes learning much more difficult than it originally appears to be.

Taking a closer look at inflection, it seems that children learn in a peculiar way. Before the age of two, they merely ignore it in all of their utterances. When they start to use inflections, they do so for only a few words, and not every time they should. When they do inflect, though, they do so correctly. Additionally, the words that they inflect in this early stage are the words they hear most frequently. In the case of the past tense, these tend to be mostly, but not entirely, irregular words such as `come => came`, `go => went`, and `run => ran`.

After a period of inflecting in this way, children's rate of inflection goes up. By the age of two or three they begin to inflect all of the words that they use (although still not all of the time necessitated by semantics). Strangely, though, in this stage they also begin to make overregularization mistakes in the irregular verbs that they use. Words like `go => went`, which they used to inflect correctly, will now sometimes be inflected incorrectly as `goed`.

Children do not choose only the correct or only the incorrect form of the word, however. They seem to switch almost randomly between the two. One good example of this is the noted utterance by a child of, "Daddy *comed* and said, "hey, what are you doing laying down?" And then a doctor *came*.[22]" While this is only anecdotal evidence, the swapping between correct and incorrect forms is very common. As children mature, they eventually do figure out the correct forms, but some overregularization usually lasts as far as the age of ten. Because of the way children inflect these verbs, correctly at first, then incorrectly, and finally correctly again, linguists refer to this learning process as the U-shaped learning curve. An example of this curve for one child, Sara[22], can be seen in Figure 2-1.

Misapplication of a rule is not confined to only the regular rule, though. While it is fairly rare, some of the time children will overgeneralize an irregular pattern and produce forms such as `think => thunk` or `fling => flang`[34]. These forms show that children must also extract and utilize the patterns within the irregulars.

Around the same time that children begin to create these errors they also begin to inflect novel words. From the very beginning of this novel word inflection, they agree with adult performance[3]. This applies to both the regular rule as in `rick =>`

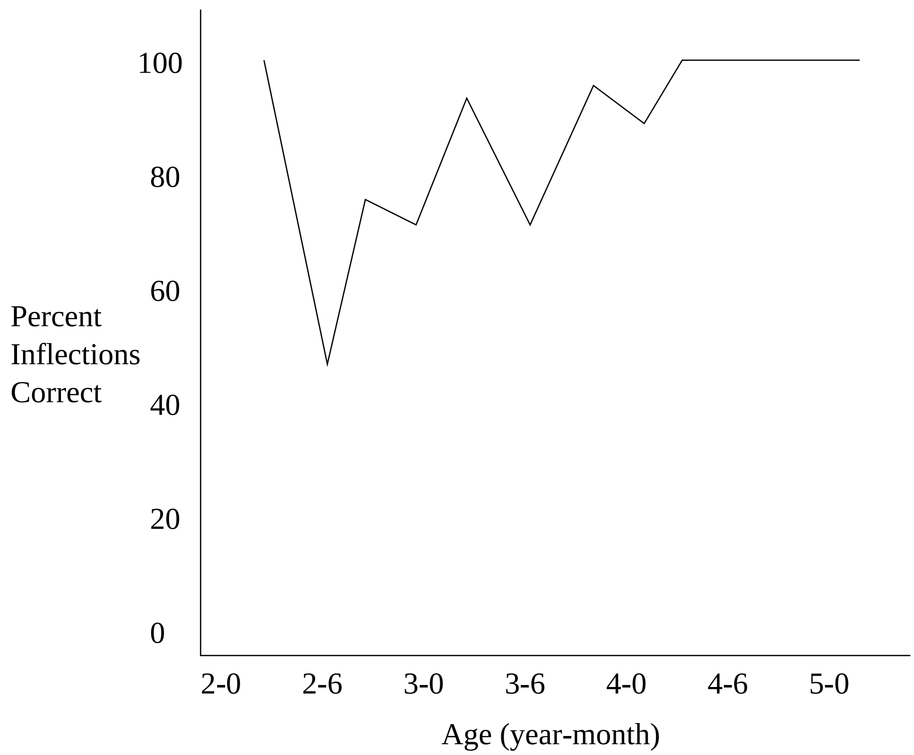

Figure 2-1: *The u-shaped learning curve seen in children.*

`ricked` and irregular rules as in `gling => glang`. Even on the words where adults disagree, children agree with them on the possible forms (as with `spling => splang` or `spling => splung`.

All of this points to the idea that in the beginning, children are basically memorizing the past tense of the most common verbs (which they hear most often) but later start to recognize patterns in the words. Once they notice the patterns, they will use them productively. This applies to the novel words as well as words that they already know. Application of these patterns then explains not only the jump in usage and the ability to handle novel words (because they now apply productively, but also the overgeneralization errors and the U-shaped learning curve (because this also causes overproductive application).

2.2.2 English Plurals

The formation of the plural in English is extremely similar to that of the past tense. It, too, has one regular rule with three parts that differentiate based on the same distinctive features. Instead of adding -ed, plurals are made by adding -s. The three analogous regular rules then follow quite straightforwardly. Those that end in an unvoiced phoneme, such as `cat => cats`, get the `s` phoneme. Those that end in a voiced phoneme, such as `dog => dogs` get the voiced version of `s`, the `z` phoneme. And lastly, those that end in a phoneme that is similar to `s`, such as `bus => buses` get the syllable `ez` added on. This almost exactly mimics the regular words in past tense formation.

The irregular nouns, although more scarce than irregular verbs, also reflect the same behavior as the irregular verbs. Like the verbs, they lie in classes such as the `mouse => mice`, `louse => lice` class. And again these irregular classes have exceptions such as `spouse => spouses`. Because of these similarities, the English plural shows that the behavior of the past tense is not just an isolated case of inflection, but instead one example of general rules.

By looking at the plurals together with the past tense, we can also generalize English inflection one step further. When we inflect by adding a phoneme to the end of the word, such as `z` or `d`, a couple of general rules play a part. The first results when the end of the word is very similar to that of the addition. This causes the addition to gain a vowel and becomes a syllable such as `ez` or `ed`. The second is that the addition takes on the value of the VOICED feature of the end of the word, thus creating the voiced-voiced and unvoiced-unvoiced property noted above. These higher level rules may be a part of the inflection process or they may be a separate process used to make words easier to pronounce. Either way, this regularity shows up throughout English inflection.

2.2.3 German Plurals

As mentioned earlier, though, English inflection is well known for it's simplicity. Most of the words follow the regular pattern, with only a small number changing in some other way. A robust model of inflection must account for the behaviors of other languages also. One language that displays some differences from English is German, specifically the German system for pluralizing.

The main difference in German is that instead of having only one main pattern, it has five: words that exhibit no change, along with words that add `en`, `er`, `s`, or `e`[20]. None of these patterns, though, cover a majority of the words in the language. Instead, they each stake out a different area in the phonological space and then act as a regular rule within that area by exhibiting the same type of exceptions, and exceptions to exceptions that show up in the rules of English.

This behavior complicates the process of learning. Instead of having one regular process with irregular exceptions, there are four (all but the addition of `s`) semi-regular rules along with a few purely irregular rules (including the add `s` rule). Each of these semi-regular rules acts much like the English regular rule within its own domain, but is treated like an irregular when in the other rules' domains. In this way, German looks to be made up of a number of overlapping rules that all act as if they are regular.

Like children learning English, however, children growing up learning German show the same behaviors: the ability to generalize, overregularization errors (with each of the rules), and the U shaped learning curve. Combining the similarities and differences of English and German, it seems that the processes underlying German inflection are not entirely different, but are based on the same mechanisms as English, just more entwined. German simply has a larger number of semi-regular patterns and more overlap in the words they cover. This creates a seemingly much more complicated system.

Chapter 3

Previous Models and Implementations

Because the acquisition of the English past tense has received a great deal of attention, quite a few models have been proposed to account for the behaviors described in the last chapter. These models fit into two basic categories that represent the two main views on linguistic learning and knowledge representation. This chapter describes some of these previous models that take either of the connectionist or symbolic approach as well as computational implementations of each type of model. Additionally, the idea of the "default rule," which may have some bearing on inflectional change is discussed.

3.1 Connectionist

The connectionist theory of language contends that there are no symbols in the brain and that all knowledge is stored as associations between input configurations and output configurations. This theory stems from the fact that our brains are made up of neurons that store information in their connections. Learning, then, corresponds to merely figuring out which inputs should be connected to which outputs.

Because there are no symbols to be manipulated in the brain under this theory, there can also be no rules. This idea is supported by the fact that amongst every proposed rule there are exceptions to refute its general applicability. A connectionist architecture explains regularities by attributing it to similarities in the input. If two words have nearly identical makeups, then they activate many of the same inputs to the system. With similar inputs, the system behaves similarly and gives similar outputs.

3.1.1 Neural Nets

Because of its direct correlation to the connectionist architecture, the obvious choice for an implementation for this type of system is a neural network. There are two major attempts at using neural networks for the task of learning the English past tense that differ in only one major aspect, the representation given as input to them.

Rumelhart and McClelland

The first system to attempt this approach was created by Rumelhart and McClelland in 1985[32]. This neural network correctly learned many regular verbs as well as some irregulars. In addition, the system exhibited some of the overgeneralization errors seen in children. Problems with this system arose, however,when it was asked to inflect words it had not been trained on. For example, when given the novel word `smeeb`, it would produce the completely different past tense `imin`, while almost all people produce `smeebed`. Pinker and Prince gave a more detailed critique of this system[29] that points out that most of its errors stem from it's Wickelfeature representation of the input words. Nonetheless, the fact that this system made substantial progress on the problem of inflection acquisition started the heated symbolic versus connectionist debate.

MacWhinney and Leinbach

In response to criticism of the R&M system, MacWhinney and Leinback created a new neural network that used a different input representation[19]. Instead of Wickelphones, it fit each input word into two templates that relied on distinctive features.

These templates corresponded to the left and right justified versions of the input words. Learning, then, relied on determining connections between words fit into these templates. This network did not produce as many strange forms as the R&M system and correctly learned many more irregular verbs. Only limited testing was done on this system, though, so it's overall accuracy on the general inflection learning task was never determined.

3.1.2 Critique

These systems exhibit many of the problems that arise from all connectionist models of human learning. The first of these is the learning rate. Neural networks generally require an extremely large number of examples (up to 80,000 words) before they can successfully learn how to produce the correct form. People, however, can learn based on as few as 10 examples. In addition, they also rely heavily on their input representation that, conceptually, defeats their whole claim that learning does not require symbols. By leaning so heavily on the their input representation, they can ignore the symbolic manipulation by pushing it into the steps that prepare the data before it is given to the actual learning procedure. In this way their systems do not explicitly manipulate symbols, but they greatly depend on them in their input representation putting a damper on their claims of symbol independence. One last problem is the way in which the neural nets created the U-shaped learning curve seen in children. This was not a result of the system architecture, but instead the result of the ordering of the input data. Early in the training, the system was exposed to a large proportion of irregular verbs, while later on in the training it was exposed to a larger proportion of regulars. This change in the input frequencies caused the change in output that includes overregularization. Children do not receive this change in word frequencies, however. This means that the connectionist models do not successfully explain the U-shaped learning curve.

3.2 Symbolic

The other main approach to the task of inflection acquisition is based on the fact that the words that require the same change when inflecting fall into natural groups based on the phonetic makeup of the word. Because of this, rule-based systems have been formulated for the problem. Motivated by the fact that linguists have been able to create lists of rules for languages, these systems attempt to automatically generate rules to explain observed examples. These rules successfully capture most of the regularity in inflection and often look similar to those found in grammar textbooks.

3.2.1 Symbolic Pattern Associator

In response to the connectionists' models, Ling and Marinov created the first rule-based system to learn the English past tense as well as some other verb inflections[15, 14]. This system, called a "symbolic pattern associator", used an algorithm that created identification trees based on observed data. This method achieved a reasonable amount of accuracy as well as mimicking some of the behaviors of children such as the U-shaped learning curve and overregularization errors. In addition, it did not produce strange outputs to novel words.

3.2.2 Inductive Logic

The second approach to rule-based learning is the quite similar, inductive logic system created by Mooney and Califf[24, 25]. Instead of making a identification tree of rules, it produces an ordered list of rules that pass from the most specific to the most general. This roughly corresponds to using the linguistic Blocking Principle, which says a more specific rule blocks the application of a more general rule. In this work, however, they made no attempt to account for the psychological patterns of learning in humans. So, while it correctly learned the inflections of words and could realistically inflect novel words, it did not reflect the human learning process.

3.2.3 One-Shot Learning

In a somewhat different approach, Yip and Sussman attempted to take advantage of the sparseness of the space of all words[37, 38]. They use a technique that creates rules based on their position in the space of distinctive features. In this space, the rules are clustered in such a way as to make them easily distinguishable based on positive and negative examples. In order to get around the problem that children receive no negative evidence, their system assumed that there can be only one correct form and generated negative examples internally that differ from the observed, correct examples. Like the other rule-based models it successfully learned most of the words given to it, could generalize to novel words, and exhibited much of the same behavior that children do. It's main advantage over the other systems is that it could also generalize on its rules to formulate higher level rules that capture multiple rules' properties. An example of this is the voiced-voiced, unvoiced-unvoiced regularity.

3.2.4 Critique

While these rule-based systems successfully learned the inflected forms of words, they did not correlate completely with observed human data. In particular, at no point could any of these systems produce conflicting forms. Additionally, the representations in the symbolic pattern associator and inductive logic programming methodologies caused them to lose some of the regularity in the data. Words that fall within the class of an exception rule but are not exceptions (ie. `need => needed` among `bleed => bled` and `feed => fed`) were treated as exceptions to the exception and given a separate rule. These words, however, are actually following the more general regular rule and should not be treated individually.

The one-shot learning method tried to account for this problem, but had additional faults with its algorithm. First, in order to generate internal near misses, it stored all of the data that it had seen up to that point, in essence, memorizing all the forms it had seen. If a human had access to all the forms he had previously seen, then he would have no need for rules in the first place. Additionally, the one-shot learning architecture required a choice between splitting rules up into separate parts or not being able to learn the inflections to words that are exceptions. For example, the system would have to choose between having multiple rules that both stand for `add -d` but apply to different sets of words or not learning the `show => showed` exception to the `know => knew` rule. Thus, while this model outwardly reflected many aspects of human behavior on its limited data set, it was internally quite different from what we know about the human brain.

3.3 Default Rule Models

An important idea used by some models to take into account regularities and explain inflection formation, is the idea of a "default rule". With this approach, one rule is singled out from the regularities and treated differently. In theory, this rule can apply to any word, but is often blocked. There are, however, some cases where the default is never blocked and can therefore be discerned. Some of these places where blocking is often forbidden are in application to acronyms and in words borrowed from other languages[20]. Because it is often the case that a single rule applies in all of these unblocked cases, that rule gains a special status that allows it to apply to any word, becoming the "default rule".

While the "default rule" defines behavior that accounts for the regularities across all classes and leaves room for exceptions to be handled, it does not define how the exceptions to the default rule are handled. Different views on this have led to at least two models using the default rule.

3.3.1 Words and Rules

In the Words and Rules model, Pinker proposes a system that is a hybrid between a rule-based approach and a connectionist approach[28, 30]. This system has one default rule and a connectionist memory for the exceptions. In this model there is only one true rule, and that can apply to any word. Any patterns in the irregulars, then, are merely a result of similarities in the input, as the connectionists claim. This idea is supported by the fact that in human learning, irregular words need to be heard fairly frequently, as in a connectionist model, or else they get generalized to some other form.

3.3.2 Rules and Competition

In order to also capture the regularity among the irregulars, Yang proposed a model of multiple rules that compete by way of the input data in order to determine which applies[35]. This model is similar to the strictly rule-based approach, but has two main differences. The first is that one of these rules is raised to the status of the default and treated differently in the ways explained earlier. The second is that instead of ordering the other rules, they all begin on equal ground and compete against each other, via frequency of use in the language, to determine which rule applies to a particular word. Each word and each rule thus has weights associated with it to allow the end form to be determined probabilistically. This competition accounts for many of the behaviors seen in human learning including those left out by the other rule-based models.

3.3.3 Critique

While the default rule approach can account for some human behaviors, it cannot be implemented easily. There is no good way to determine which regularity seen in the language should become the default rule. While it seems straightforward to find the default in English, this is not the case in some other languages such as German. For instance, analysis has shown that using this technique, the default rule for the German plural is the `add -s` rule that applies to less than 10% of the nouns in the language[20].

This difficulty is probably why no implementations of this approach have been attempted, but it also demonstrates a hole in the theory itself. If there is no way to figure out what the default rule is, then how do humans figure out which of the observed regularities should become the default rule? The only proposed answers define it by its behavior and determine the default rule by choosing the rule that applies in the special cases mentioned earlier. Doing that, though, creates a circular definition that must then be hard-coded into the system. For humans that would imply that the properties of a default rule are inherent to the human brain, an unlikely prospect.

Chapter 4

Generalize and Sift

The model presented in this thesis is a rule-based model motivated by children's behavior. As such, it exhibits the same learning curve and makes the same type of mistakes during learning as children. The core of this model is a two-step learning process combined with a probabilistic generation mechanism. The learning mechanism is responsible for determining rules from regularities in the input as well as deciding which rules apply to which words. It is activated every time that the system is exposed to a word, using that instance to assist it in learning the more general rules of the language. The generation mechanism is activated when the system needs to produce a form. It chooses among the possible forms based on the knowledge that the learning system has gained up to that point.

4.1 Learning

The idea behind this learning algorithm is a two step process. The first step, *generalization*, is responsible for discovering patterns in the observed words and creating rules that capture these patterns. These rules only need to be a first approximation to the observed regularities. Their role is to allow inflection based on incomplete knowledge. Specifically, the rules this step produces often turn out to be more general than the rules that would be necessary to cover just the examples seen.

The second step, *sifting*, works out the kinks in the imperfect rules made by *generalization*. It refines the inflections based on more data and is responsible for completing the knowledge about the inflections.

The two methods work in parallel each time they are exposed to an inflected form. *Generalization* updates the rules it has proposed while the sifting method deduces the correct application of the rules. Together they eventually correctly identify every word's inflection.

4.1.1 Generalization

The core of the *generalization* algorithm is basically a specific to general search through the phonemic space of words. It begins with a rule that applies very specifically to only one word. Then the rule generalizes to more words until it eventually covers all of the words it should apply to.

For instance, look at Figure 4-1. Assume that the system knows the word `walk` and sees the past tense `walked`. It will then create Rule #1, that says, "for the word `walk`, add `ed` to the end." This rule is so specific that it only applies to one word, `walk`, almost like memorizing the inflection for this one word. Suppose that a bit later, though, it learns the pair `talk => talked`. The system then proposes a second, very similar, rule, Rule #2, "for the word `talk`, add `ed` to the end." In the beginning each of these rules only applies to the one word that instigated it.

From these two rules, the process then generalizes to cover both words with Rule #3, "for the word `*alk` add `ed` (where `*` stands for any number of arbitrary letters)." After seeing a few more words that use this same rule (perhaps `rack => racked`, `turn => turned`) and `kill => killed`, it will eventually end up with Rule #10, that says, "for the word `*` add `ed`." This rule would correspond to the regular past tense inflection rule. Similarly, given examples, the system would also come up with rules such as, "for the word `*eed` (as in bleed, feed, or breed) change the `ee` to a `e` (to get bled, fed, or bred)." Eventually the system would have rules to cover every inflectional change in the language.

These rules almost always cover more words than they would cover if they were perfect. For instance, the `*eed` rule above also covers the word `need` that actually

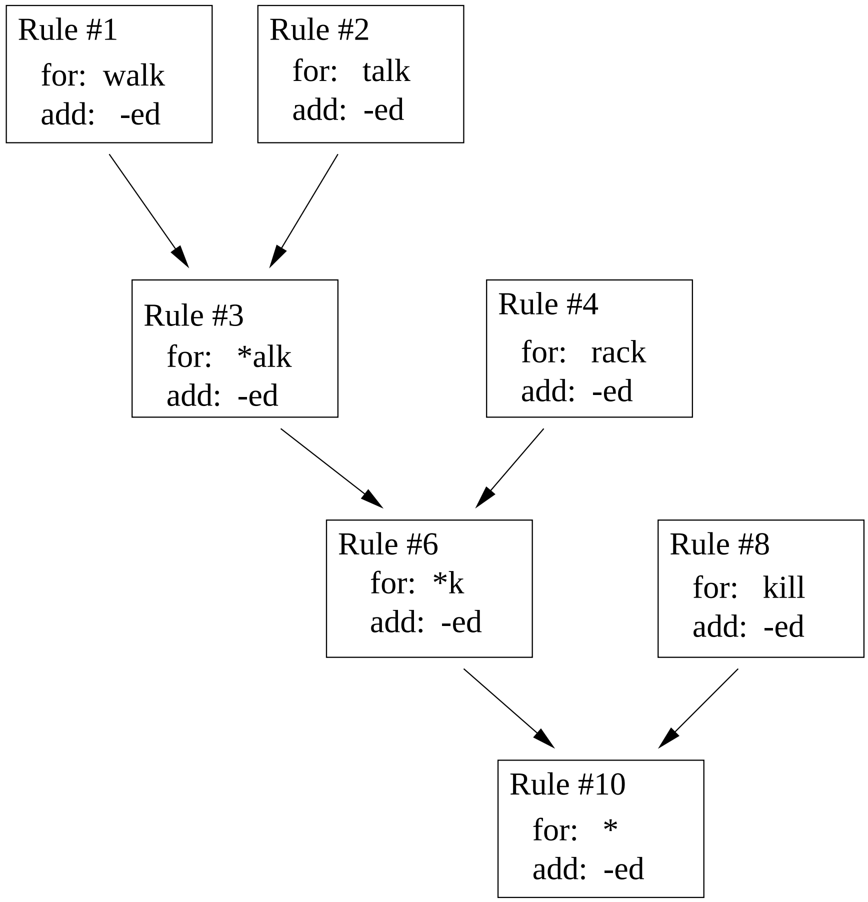

Figure 4-1: *The process of rule generalization.*

follows the add `ed` rule. This is not a problem for *generalization*, however. All it is concerned with is making sure that the rules cover all the words that it should cover, hence the specific to general search through the space.

It is important to note, though, that the example above used letters to illustrate the *generalization* process. This was done in order to demonstrate the procedures and properties of the learning process. The actual system, however, does not learn using letters. Instead, it uses the very detailed space of distinctive features. Because of this, it would not come up with the add `ed` rule as mentioned above, but instead three similar rules that cover the three cases of the regular rule. These rules would look more like, "for the word `*V` add `d`," "for the word `*v` add `t`," and "for the word `*(t|d)` add `ed`" (where `V` stands for any phoneme with the VOICED feature, `v` stands for any phoneme without the VOICED feature, and `(t|d)` stands for either of the phonemes `t` or `d` represented by their shared features). These three rules would then cover the three versions of the the regular rule as seen by the words' pronunciations.

4.1.2 Sifting

As the *generalization* process figures out rules for the words, it often turns out that some of these rules overlap. For instance, even with just the two rules mentioned above, quite a few words, such as `need` and `feed`, fit both the rule that applies to `*`, that matches any word, and the rule that applies to `*eed`. To complicate things, even though the same two rules apply, `need` requires the add `ed` rule while `feed` requires the vowel change rule. To correct for this, the system must have some way to choose between multiple rules and yet allow for any of the rules to be chosen.

This is where *sifting* comes into play. For each word, the system starts out by assigning a probability to each matching rule. This probability represents the chance that the particular rule is the one that applies to that word. Each time the system sees an inflected form of that word, it examines the rules for a match and updates the probabilities accordingly, raising the correct rule's and lowering the rest. Although the system treats every example as if it is correct examples, this cautious movement accounts for the fact that the examples may sometimes have errors in them.

As an example of this process, both `need` and `feed` might start out with a 50/50 chance that each of the two above rules is the correct one. After a few examples of `needed` and `fed`, however, these probabilities would diverge, raising the probability of the "add `ed`" rule being correct for `need` and lowering it for `feed`. As seen in Figure 4-2, despite the initial confusion, once the system sees each of the inflected forms and alters the probabilities, it converges on the correct rule for both of the words.

4.2 Generation

Because we cannot yet peer into children's brains, though, an equally important part of the system is the part that generates the inflected forms of verbs. By making the

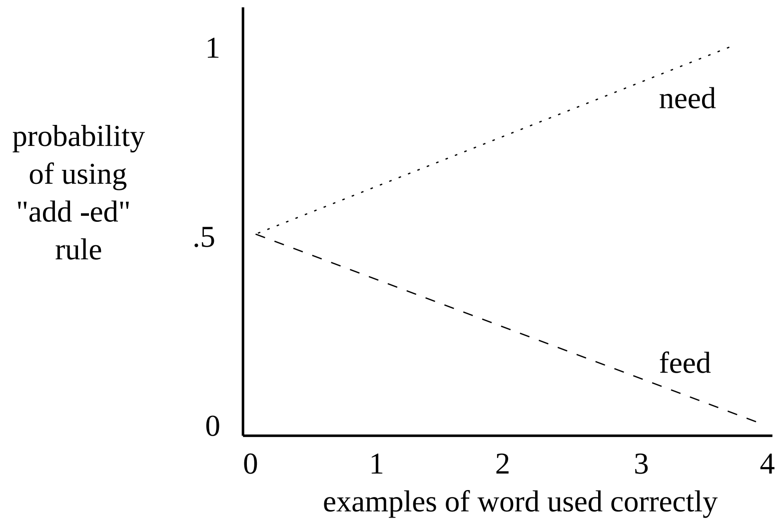

Figure 4-2: *The prominence of the* `add -ed` *rule when competing with the* `ee => e` *rule on two words,* `need` *and* `feed`.

system produce inflections, we can compare its results with the forms people produce while in the process of learning as well as the forms produced for novel words after they have completely grasped the language.

When the system needs to inflect a known word that has only one possible rule, the result follows naturally. It merely applies that particular rule to the word. Similarly, when asked to inflect a novel word that falls under only one rule, the system must assume that the one rule is the correct one, and, once again, apply the single possible rule to the word. When the system has more than one possible rule for a particular word, though, a slightly more complicated process must occur.

4.2.1 Probabilistic Selection

Resorting to probabilistic selection could happen for one of two reasons. The first reason is incomplete knowledge about a word that falls under more than one rule. In these cases there must be two or more different applicable rules and the system must still have not determined which is the correct one. However, at this point, each of the rules will have a probability associated with it that corresponds to the likelihood that the rule applies to the particular word. The system then picks randomly between the possible rules in proportion to the associated probabilities. By doing this, it can determine a unique solution each time it needs to inflect the word despite conflicting possibilities.

The other reason for the system to use probabilistic selection is when novel words are encountered. In these case, the system has never seen the word before and has no idea what the correct inflected form is. In order to produce a form the system checks each of the rules that it knows, finds all of the rules that could apply to the word, and then randomly chooses between them. As mentioned earlier, for many words there is only one rule that could apply, so the system just applies it. For the ones with more than one possible rule, the random selection disambiguates much like it does for the words with incomplete knowledge.

4.3 Behavior

Over a sample of many words, this model accurately models human behavior. It can correctly learn how to inflect any word it sees, exhibits the same behaviors while learning as children do, and can generalize novel words efficiently. *Generalization* creates multiple rules that capture all of the regularity in the language, while *sifting* determines when to apply them. Finally, the probabilistic generation process accounts for the unpredictable, yet bounded behavior seen both while learning and while inflecting novel verbs.

At the beginning of the learning process the system has only a few rules, each of which each applies to only one or two words. When inflecting these words, it will do so correctly, but for other words it cannot guess at any inflection at all. This is comparable to having a few memorized inflected forms, matching the behavior seen in children's early acquisition.

As the system generalizes its rules, however, it begins to exhibit some of the other behaviors seen in children. At this time, the system will greatly increase the number of words it inflects. Each time a rule is generalized, it will not only cover the word that caused the generalization but also a few that lie inside the space it has generalized over. For instance, after seeing just the two pairs `bleed => bled` and `feed => fed` the system proposes a rule that applies to anything of the form `*eed`. It can then use this rule on the word `breed` despite having never seen `bred`.

Like children at this stage of increased inflection use, the system can also begin inflecting novel words. With more general rules, novel words are much more likely to fall within one of the rules and be inflected. Thus, given the novel word, `smeed` it could produce the form `smed`.

In addition, greater generalization causes rules to overlap more often. This, in conjunction with the probabilistic selection mechanism, can cause overgeneralization errors. Because one of the rules corresponds to the correct form while the other an overregularized form, selecting semi-randomly (according to the probabilities) will cause the system to overgeneralize some of the time, just like children in the same stage. Then, once these errors occur, the slow sifting process starts to create a U-shaped learning curve as it figures out the correct form for the words that lie within the overlap area.

Eventually, however, *generalization* will completely cover the area for each of the rules and *sifting* will uniquely determine the correct rule for each word that lies in the overlapping areas. This corresponds to the system in the "adult" stage. Now it correctly inflects all of the words it has seen and the selection process will only be utilized for novel words - a place where adults often differ in their speech.

An important feature of this model is that while it exploits the regularities seen in the words by creating rules, *sifting* allows for a great deal of variation in the output. This variation lets all the irregulars through as well as the exceptions to the irregulars. It also allows the exceptions to either go to a different irregular rule or fall back on the regular rule, whichever is deemed appropriate by the observed forms.

Additionally, this process allows for the ambiguities found in human speech. Dialectic variants merely correspond to differences in the input causing *sifting* to choose the local rule instead of the more widely accepted. Historical changes in a word's inflected form can result from too few examples of the correct form leading to *sifting* choosing the incorrect rule. Lastly, words with multiple, accepted forms can result from *sifting* never converging one way or another, leaving the selection process to choose each time.

This model also makes some predictions about the behavior of people as they learn. The first prediction is where the overgeneralization errors occur. First, overregularization should occur across the board on all of the irregulars. This behavior does in fact show up in children[22]. Second, overirregularization should occur only within the tight phonological space of the irregular rule. Again this turns out to be true as seen in mistakes such as `trick => truck` and `swing => swang`[34]. Lastly, it proposes that the most difficult words to learn how to inflect will be those that have the most competing rules. One good example is the verb `bring => brought` that also fits in the class that would change it to any of `brang`, `brung`, or `bringed`. This word turns out to be especially troublesome for children and some do not ever learn it correctly, instead keeping it into adulthood and creating dialectical variants in the language[4].

In all of these ways the behaviors of this system reflect the behaviors seen in children as they learn inflection as well as the behaviors of adults once they have mastered inflection. The system is also based on fairly simple, psychologically plausible methods that only create complicated behaviors when combined. Because of this, the "Generalize and Sift" process of learning produces an accurate model for inflection learning in humans.

Chapter 5

Implementation

I have implemented two versions of the "Generalize and Sift" learning model for computational testing. The first version uses a deterministic selection routine that is sensitive to errors in the input data. This version uses the generalization mechanisms, but does not incrementally update the probabilities of rules. Instead, it assumes that the input data is universally correct and directly assigns a rule to a word once it has seen a single example of that rule in use with the word. Because of this, it learns faster than humans, and does not make all of the same types of mistakes. Specifically, it does not exhibit the U-shaped learning curve.

The second version implements the entire "Generalize and Sift" learning model, including the probability updating. This accounts for irregularities in the data, causing the learning to slow to match that of children as well as bringing out many of the behaviors that children exhibit. This chapter describes the details of these implementations and their performance.

5.1 Input and Output

The input to the system is a word, its phonemic transcription, its part of speech, and an index that represents its meaning. Some example entries can be seen in Table 5.1.

Note that words that differ only by tense are given the same meaning index. Because of this, the system knows that two words stand for the same concept, but

Word	Phonemes	Part of Speech	Meaning
feed	/f iy d/	Present Verb	23
fed	/f eh d/	Past Verb	23
cry	/k r ay/	Present Verb	31
cried	/k r ay d/	Past Verb	31

Table 5.1: *Examples of system input data.*

only differ in the grammatical use and pronunciation. Giving words the same meaning index in the input is based on the idea that in order to analyze changes due to tensing, the system (or a child) must already know that two words are actually different forms of the same word.

The outputs from the system are forms requested by specifying a meaning index and tense. For example, A request for `meaning=23, tense=VPa` would return, `/f eh d/`. This request and answer format was chosen in order to simulate the idea of the system trying to express a particular idea. It would have some notion of the concept it wanted and what grammatical context it would need to be in. It would then need to look up the correct way to say the word. The actual output is just the pronunciation of the requested word. This can be done at any point during the learning process.

Additionally, after learning, a list of the words that the system knows and a list of the rules that it uses can be displayed. An example of this can be seen in Appendix B. While these lists cannot be compared to the representations in human brains, they can be examined in light of the known regularities in the language.

5.2 Lexicon

Once put into the system, the words are each represented by a lexical entry. This entry contains all the information input to the system: the word, pronunciation, part of speech, and meaning index. A sample lexical entry is shown in Table 5.2. The spelling of the word, however, is never used by the system. It was included only to make the data easier to interpret.

The pronunciation is stored as a string of phonemes converted into a list of dis-

Word Spelling	sample					
Pronunciation		s	ae	m	p	el
	syllabic	-	+	-	-	+
	consonantal	+	-	+	+	+
	sonorant	-	-	+	-	+
	voiced	-	-	+	-	+
	...					
	back	-	-	-	-	-
	low	-	+	-	-	-
Part of Speech	Noun					
Meaning Index	17					
Rule List	Index	12	4	31		
	Probability	.3	.5	.2		

Table 5.2: *A sample lexical entry.*

tinctive features according to the list in Appendix A. Each phoneme is represented as a list of 18 binary digits and a pronunciation is stored as a series of these binary lists. Because everything is converted to features on input and then converted back upon output, all processing on the phonemes in the system is done on the distinctive feature representation, not on phoneme names themselves.

The part of speech is handled in a similar fashion. It is stored as a list of parts of speech. In this way a past tense or plural form is considered an addition to the present tense or singular form. For example, a noun is given the tag of "noun", while a plural noun merely adds the tag "plural" to the existing "noun" tag to result in a "noun plural". This abstracts the idea of the part of speech into a list of grammatical features and allows it to be treated much the same as the pronunciation, which is a list of phonemes.

In order to allow for the probabilistic selection, each lexical entry has a list of rules that can apply to it, each with an associated probability. When a new word is learned, each rule that fits this word is added to the list. If no rules match it, then the list is left empty until an applicable rule has been formulated.

5.3 Transformation Rules

A transformational rules is represented within the system as a template of words that it matches, a list of the changes that the rule makes, and a number specifying how far to shift the original word before applying the changes. Both the APPLY TO template and the CHANGES template inside the rule are represented as words with one difference from those in the lexicon, that they can leave features unspecified. A sample rule can is shown in Table 5.3

Apply to	Word Spelling	DC (Don't Care)			
	Pronunciation		*	ao	k
		syllabic	DC	+	-
		consonantal	DC	-	+
		sonorant	DC	-	-
		voiced	DC	-	-
		...			
		back	DC	+	+
		low	DC	+	-
	Part of Speech	Verb			
	Meaning Index	DC			
	Rule List	DC			
Shift	1				
Changes	Word Spelling	DC			
	Pronunciation		*	t	
		syllabic	DC	-	
		consonantal	DC	+	
		sonorant	DC	-	
		voiced	DC	-	
		...			
		back	DC	-	
		low	DC	-	
	Part of Speech	Past			
	Meaning Index	DC			
	Rule List	DC			

Table 5.3: *A sample rule generated by the system that would inflect* `/t ao k/ => /t ao k t/` *(*`talk => talked`*) and* `/w ao k/ => /w ao k t/` *(*`walk => walked`*).*

The APPLY TO template is a "word" that contains only those phonemes and features needed to explain what set of words this rule applies to. For instance, the rule for the words `send => sent`, `bend => bent`, and `lend => lent` would have an APPLY TO template of `/* eh n d/` in distinctive feature representation. The `*` means that any number of any phoneme will match[1]. Similarly, the APPLY TO template for the voiced version of the regular rule would be `*V` (where the V stands for a required *voiced* feature with every other feature in the phoneme unspecified).

The "word" that stands for the changes, the CHANGES template, looks very similar to the APPLY TO template. Again, many of the phonemes and features are left unspecified signifying that they remain the same after the rule is applied. A word that the rule applies to is changed based on the specified phonemes and features. Any feature specified in the CHANGES template overwrites the feature in the original word. For the `send => sent` example above, the CHANGES template would be `*v` (where the `v` stands for the unvoiced feature with all others unspecified)[2]. This single feature change turns the `d` to a `t` in these words and completes the change to the past tense.

The SHIFT number only comes into effect when the rule specifies either a prefix or suffix added onto a word. In these cases, the SHIFT adds phonemes that are completely unspecified to the beginning or end of a word. These unspecified phonemes are then filled in by the CHANGES template. In the regular rule example above, there is a SHIFT of 1 and CHANGES template of `*d`. When applied to a word, such as `/t er n/` (turn), the rule first shifts by 1, making it `/t er n _/` (where _ stands for a phoneme with no specified features), and then applies the changes to result in `/t er n d/` (`turned`).

With these two ideas of shifting and feature substitution, a word can be changed in any way from mere additions to entire word suppletion. Furthermore, because of their representation as words, the APPLY TO and CHANGES templates can be easily learned by the system.

---

[1] The `*` can only appear at the beginning or the end of the word in order to avoid having regular expression matching. Psychologically, this means that only local context is necessary to determine if a rule applies.

[2] Here the `*` is used merely as a placeholder to say whether the changes are right or left justified. This is important when deciding whether an addition is a prefix or suffix.

5.4 System Procedures

The procedures of the system is separated into two components. The learning component, encapsulating both the generalization and sifting algorithms, deals with the input and creates the lexicon and rules. The generation component, uses the selection algorithm and produces output when elicited.

5.4.1 Learning

The learning module is activated every time that the system is exposed to a word. Figure 5-1 shows a diagram of this module. It first checks the word against those listed in the lexicon. If the word is contained directly in the lexicon, then the system proceeds to the next word. However, if the word is not directly in the lexicon, the learning process is activated.

The lexicon is designed around the principle that it can only contain one word with a specified meaning index. This corresponds to the idea that different forms of the same word should not all be memorized. Instead, they should be derived through rules applied to a single base form. Because of this constraint, the system checks to see if any lexical words have the same meaning index as the observed word. If there are no matches to the meaning index, the system adds the word to the lexicon and proceeds to the next word.

However, if a word in the lexicon has the same meaning index as the observed word, the system needs to discover which is the root form. It compares the grammars of the lexicalized and observed words and if the observed word has a "smaller" grammar, it switches the role of the two words. For instance, when comparing a lexical word with grammar "Plural Verb" and an observed word with grammar "Verb," the system will swap the two words, replacing the first with the second within the lexicon. Because of this, lexicalized words are generally the root form. Additionally, the lexicalized, inflected forms cannot will not have any rules associated with it.

Next, the system checks to see if there is a rule to derive the inflected word from the word in the lexicon. If the lexical word has an appropriate rule, the sifting

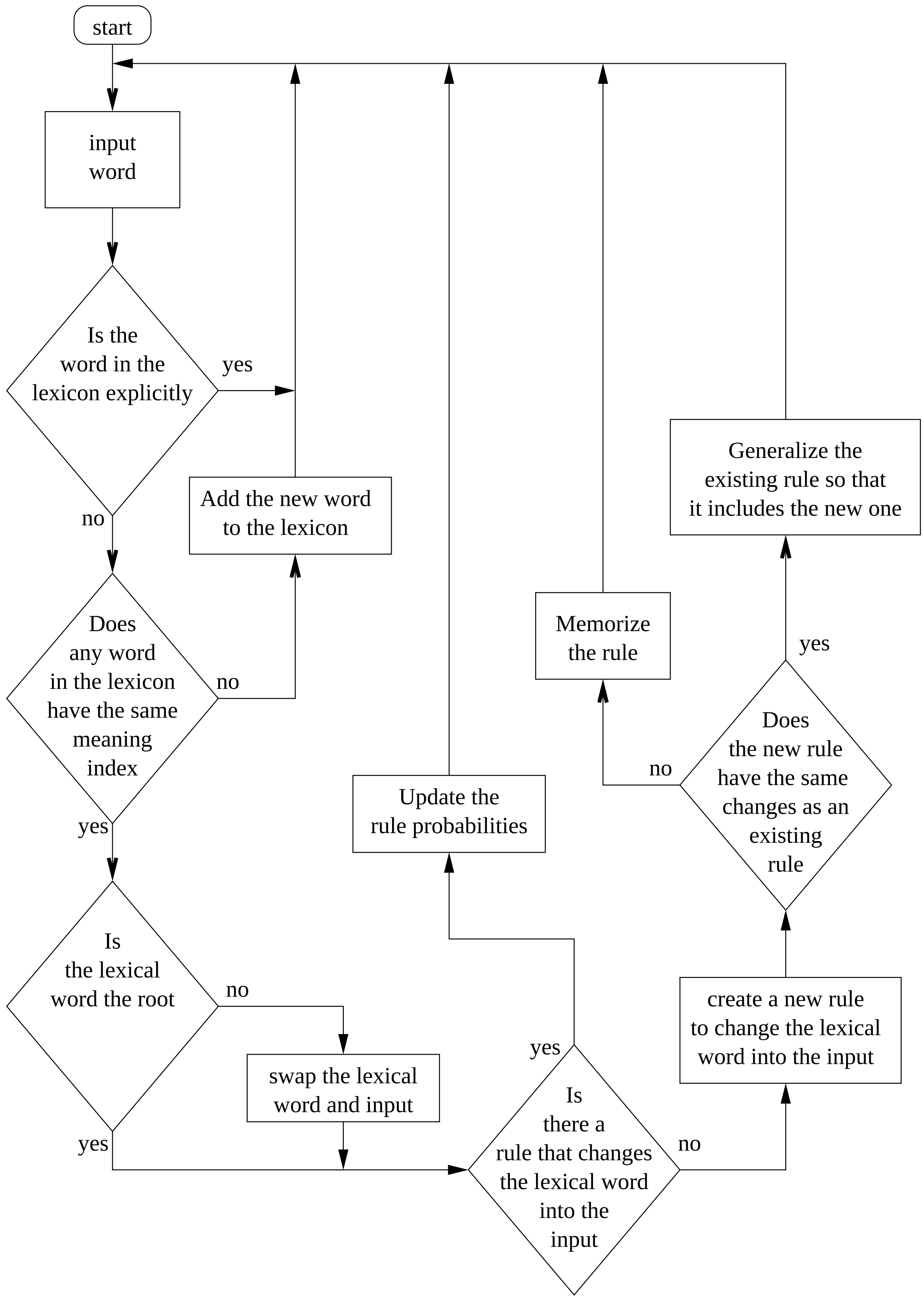

Figure 5-1: *Flowchart of the learning process.*

process takes effect. The system then raises the probability of that rule and lowers the probabilities of the other, competing rules.

On the other hand, if the system has a lexical entry with the same meaning index, but no rule to create the observed word, then it creates a new rule to explain the observed change. To do this, it first analyzes the differences between the lexical word and the observed word. From this, it creates a rule with an APPLY TO template equal to the lexical word and appropriate SHIFT and CHANGES template to convert it into the observed word. For example, when the system makes a rule for the pair `/t ao k/ => /t ao k t/` (`talk => talked`), the system notices the greatest similarity between the words when the root is shifted left by 1 phoneme. It then sets the CHANGES template's SHIFT to 1 and adds the `t` phoneme to the end. A generalized form of this rule, which differs only by the APPLY TO template, can be seen in Table 5.3. For comparison, when it makes a rule for the pair `/s ih t/ => /s ae t/` (`sit => sat`), it sets the SHIFT to 0 and creates a CHANGES template that is completely with unspecified values except for a - in the second to last phoneme's HIGH feature, as seen in Table 5.4. This - signifies the change from `ih`, which contains the HIGH feature, to `ae`, which does not.

The system then decides whether the rule is, in fact, a new rule or merely a generalization of a rule that already exists in its main list. To do this, it compares the CHANGES template in the new rule to the CHANGES template in each of the existing rules. If any there is a match, in both part of speech and pronunciation, then the system generalizes the APPLY TO template of the existing rule to also cover the new word.

The generalization is done by merging the new rule's APPLY TO template (which only applies to the one word) with the known rule's APPLY TO template (which may apply to many words). The system goes through each feature of each phoneme and compares the values. When corresponding features of the two templates agree, the merged template retains that value. When they differ, it makes the feature value in the merged template unspecified, allowing either value to match. In this way, it performs the specific to general search over each of the APPLY TO template's phonemes.

Apply to	Word Spelling	sit			
	Pronunciation		s	ih	t
		syllabic	-	+	-
		consonantal	+	-	+
		sonorant	-	-	-
		voiced	-	-	-
		...			
		high	-	+	-
		back	-	-	-
		low	-	-	-
	Part of Speech	Verb			
	Meaning Index	13			
	Rule List	empty			
Shift	0				
Changes	Word Spelling	DC			
	Pronunciation		*	??	DC
		syllabic	DC	DC	DC
		consonantal	DC	DC	DC
		sonorant	DC	DC	DC
		voiced	DC	DC	DC
		...			
		high	DC	-	DC
		back	DC	DC	DC
		low	DC	DC	DC
	Part of Speech	Past			
	Meaning Index	DC			
	Rule List	DC			

Table 5.4: *The initial rule generated by the system to inflect* `/s ih t/ => /s ae t/` *(*`sit => sat`*).*

5.4.2 Selection

When requested to produce a particular form, the system uses a fairly straightforward algorithm as seen in Figure 5-2. Given a meaning index and part of speech, it first checks the lexicon for the specified word. If it succeeds, it outputs the lexical word. If it does not find a complete match, it looks for a word in the lexicon with the same meaning index, but a different part of speech. Without that, the system fails to output anything, much like a person who cannot find the word for the concept he is trying to express.

However, with a word that matches the meaning index, but not the part of speech, the system can then begin applying rules to inflect the word. It checks through the list of applicable rules and finds all rules that cause the correct inflection. If there are none, the system merely outputs the lexicalized word form which has the wrong part of speech. If there is one match, then the system applies that rule and outputs the result. If there is more than one possible rule, then the system chooses randomly between all the matches, weighting them according to their stored probabilities.

5.4.3 Differences in Deterministic Implementation

The system explained in the preceding sections corresponds to the complete "Generalize and Sift" learning algorithm. The second system assumes no errors in the input data and therefore does not rely on the cautious learning by updating probabilities. This only changes one main aspect of the implementation.

Any time that the system sees an inflected form of a word, it immediately changes the probability of the rule that causes that change to 1. This can happen in two different cases during the system's operation. First, when an observed form causes either a new rule to be formed or a rule to be generalized to cover a word, this rule will gain complete hold on the word, allowing no other rules to displace it. The other case occurs when the system has generalized multiple rules that could fit a word. Once it finally sees an inflected form, it will raise the probability of that rule to 1 and eliminate the other possibilities entirely.

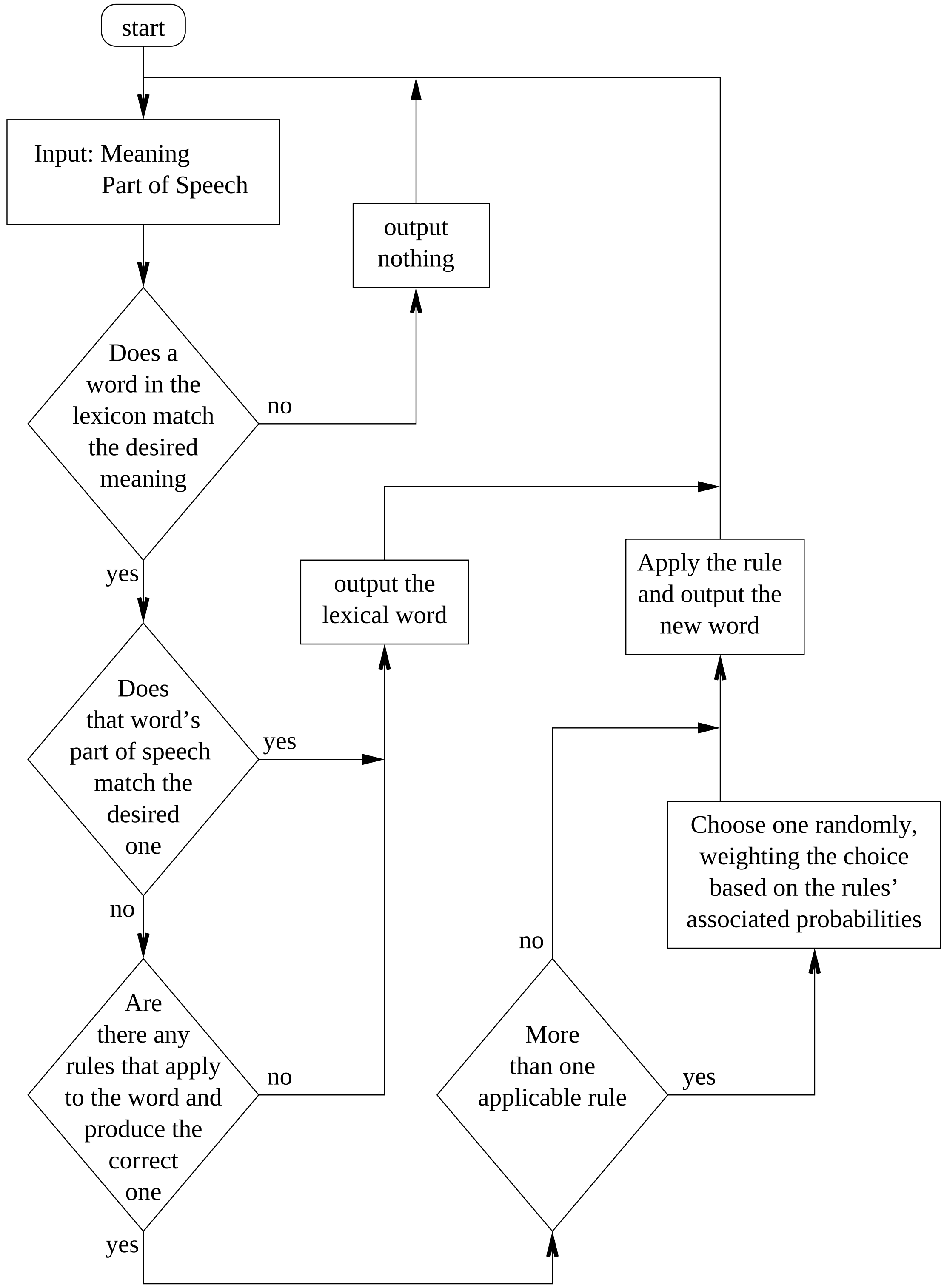

Figure 5-2: *Flowchart of the selection process.*

The selection mechanism relies on this change, but does not need to change itself. Any word where the system has seen the inflected form will have only one applicable rule, resulting in the system selecting that rule. The rest of the words may still have multiple possibilities, but again, the selection mechanism need not change to decide between them.

5.5 Performance

The systems were run on a variety of data sets in order to test individual aspects of the model. All of the data presented was in English and given to the system in the format previously explained. The deterministic model was tested in order to find out how fast the "Generalize and Sift" process could learn the rules. The complete system was tested in order to compare the model's behavior to observed human behavior. Both models performed well, correctly learning the rules presented to them and making reasonable generalizations.

5.5.1 Present Participle

In order to test the model's rule making and generalization capabilities, the systems were tested on the present participle (`run => running`, `fly => flying`). In English, the present participle has only one rule and no exceptions to this rule. Even words that seem like they should be exceptions, such as `swing => swinging` and the notorious `be => being`, follow the rule for this form. This universal regularity allows the rule creation and generalization mechanisms to be tested without interference from the selection mechanisms.

Both the deterministic and complete systems were run with this data set. Because there were no competing rules, both performed identically. Both systems were able to extract the rule and then apply it productively soon thereafter. When asked to inflect a verb, the systems either returned the root form (if they did not yet have a complete grasp of the rule) or provided the correct form. Additionally, after seeing as few as 2 words, the system could correctly apply the rule to novel words.

5.5.2 Past Tense

Small Data Set

The next data set tested the systems on a small set of verbs and their past tense forms[3]. These verbs contained examples of all three versions of the default rule as well a few irregular verbs. This test required the systems to keep track of multiple rules, as well as forcing the systems to select between competing rules.

The two systems again performed quite well. They both extracted appropriate rules from the data and used them correctly on the learned words. They also generalized these rules to novel words in the same way that humans do, such as `mate => mated`. A listing of the lexicon and rules learned by the system after just the first iteration of the data can be found in Appendix B.

Because the data contained only a small number of verbs, only one verb, `feed` was covered by more than one rule, `feed => fed` and `feed => feeded`. In order to see how the system decided between rules, this verb was tracked throughout the learning process. The results of this, showing the learning curve for complete model, can be seen in Figure 5-3. For this word, the system displays the U-shaped learning curve also seen in children.

Large Data Set

The main test of the model used a large set of verbs and their past tenses. Only the complete model was tested with this data. For this test, the words were randomly chosen from a large corpus in proportion to their use in the language[12]. By doing this, the input represented, as closely as possible, the same words that a child might hear as he is trying to learn how to inflect.

As in the previous tests, the system correctly learned all of the words it saw and showed the U-shaped learning curve for common verbs that fall under multiple rules. This behavior parallels the behavior seen in children during the same stage of

[3]These verbs were: dance, drop, fix, kiss, laugh, like, look, touch, walk, answer, call, cry, hug, turn, add, need, paint, wait, draw, sing, and feed.

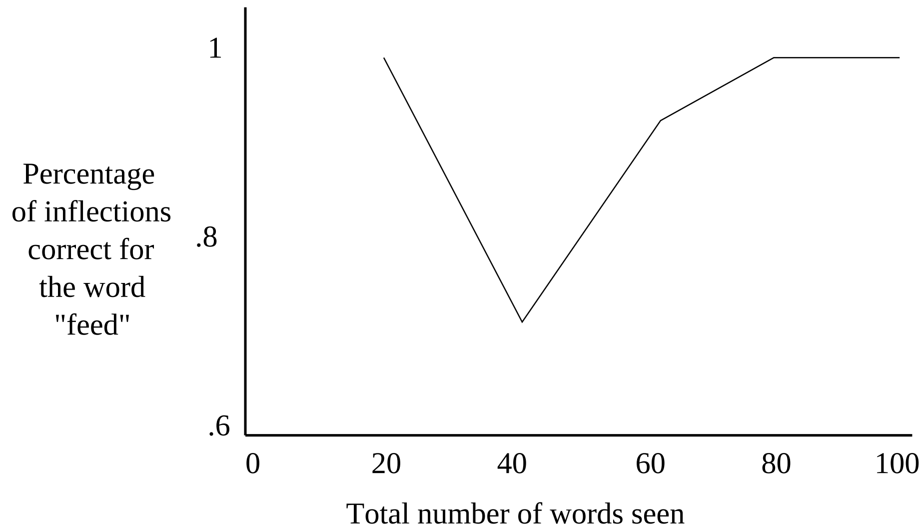

Figure 5-3: *The progression of the system learning the past tense rule for the word* `feed`.

development. Also like children, the system began to overgeneralize around the same time that it started to inflect words that it had not seen in the correct form. After the child-like learning phase, the system eventually converged to an "adult" state where it produced correct forms exclusively and would inflect novel words to the same form as people.

The rules produced by this system looked very much like those found in grammar textbooks. For instance, one rule added the `t` phoneme to any word that ended in an unvoiced phoneme while another added the `d` phoneme to any word that ended in a voiced phoneme. In this way, the system produced rules that were not only accurate for the input words (as displayed by its correct performance) but also useful for analysis of new words. This allows both the system and an outside observer to see how it would perform on any given word.

5.5.3 Plural

The complete system was also tested with a small set of nouns and their plural forms[4] in order to see if it performed comparably to the past tense tests. The results showed similar behavior to that on the small data set of past tense verbs. The system again correctly learned the words presented while making similar mistakes during the process. It also generalized effectively to novel words, both during the learning process and after it had finished learning the training data. This test confirmed this model's value in understanding the general problem of inflection acquisition.

[4]These nouns were: cake, cat, chief, cup, fruit, month, bottle, boy, dog, girl, gun, box, bush, church, dish, glass, horse, nose, house, leaf, foot, and man.

Chapter 6

Conclusions

The “Generalize and Sift” model captures much of the behavior seen in inflection acquisition both in theory and practice. It could also lend itself to extension, with only minor modifications, to other area of learning. Additionally, while the implementation performed well on the task of learning the past tense, it could be run through simulations of other learning tasks.

6.1 Contributions

In this thesis, I have made the following contributions:

- Conceived of a model of inflection acquisition based on the behavior of children as they learn inflection that uses psychologically plausible procedures.
- Implemented two version of this model in order to verify the learning algorithms. The first version uses the assumption of perfect input data in order to prove the methods. The second uses a more conservative approach, accounting for variants among the input data and mimicking the behavior of children as they learn inflection.
- Shown experimentally, through use of the implementation, that this model accounts for many of the aspects of human inflection, including overgeneralization

errors, the U-shaped learning curve, dialectic and historical variants, and the ability to generalize to novel words.

6.2 Future Work

While this model captures the properties of inflection acquisition, this area has room for future improvements. The model could be altered or expanded, while the implementation could be adapted to other tasks to provide more support for its ideas.

6.2.1 Model

One area where the model may need improvement is in the area of the default rule. As mentioned earlier, evidence exists that one or more rules are singled out from the others and treated differently. In the current model, these differences can merely be learned as places where the rule can or cannot apply. If the idea of the default rule turns out to be inherent to humans, though, then it should be added to the model along with an appropriate way to determine it. This would push the rule selection part of the model closer to that of the Rules and Competition model proposed by Yang.

Another possible addition would be to make the amount the weights update dependent on time. While child language is fairly elastic and can change based on a few examples, adult language is quite the opposite. By adjusting the amount that the sifting process updates weights (a large amount at first and then gradually decreasing), the model could account for the fact that adults require much more conflicting input to change their speech patterns.

The model could also be extended to include other constraints on rule application. Currently, it relies on only the constraints of a word's part of speech and pronunciation, the meaning value is used only to match forms of words. Making words' meanings into sets of features, may allow the principles of *generalizing* and *sifting* to be used for semantic distinctions. One possible example would be an animate/inanimate distinction that would help determine the appropriateness of sentences such as, "the man ran home," versus, "the box ran home."

One last possibility for this model is to stretch it into other areas of learning. The ideas of *generalization* followed by *sifting* are not only useful in the area of language, but many tasks that involve creating overlapping rules with exceptions. Thus these principles may be useful in similar tasks such as mapping spelling to pronunciation, or tasks as distant as group classification (such as determining the features of mammals).

6.2.2 Implementation

To date, the systems presented in the this paper have been exposed to the task of learning English inflections. The first step delving further into their applicability would test them on other languages, such as German. While the abstract models do account for the behaviors seen in other languages, the implementations have not been tested on them.

The system could also be set to learning tasks. Within language acquisition, it could be used to learn how to change a word's part of speech. For instance, changing an adjective to an adverb often calls for the addition of the syllable `ly` to the end of the word as in `clear => clearly`. Negating a word is another process similar to this in which a prefix is added to the front of the word. `Possible => impossible`, `conceivable => inconceivable`, and `believable => unbelievable` are good examples of this type of negation. Both of these problems, along with any other problems involving word changes or suffixation, are well suited for the system created for this research.

6.3 Summary

In this thesis, I have presented a psychologically plausible model of inflection acquisition. While the model is composed of fairly simple components, it behaves in a complicated fashion similar to that of humans as they learn. Within this model are the ideas of *generalization* and *sifting* for rule creation in a sparsely filled space, along with the idea of probabilistic selection for the use of incomplete knowledge. These general principles work well for this task because they capture the essence of the imperfect rules that people appear to use. This allows them to accurately reflect the learning process of humans. As such, the "Generalize and Sift" model may also be useful in explaining other areas of human learning.

Appendix A

Phonemes and Distinctive Features

This is a table of English phonemes, an example of each in use, and the distictive features of each of them[2].

	p	b	m	em	t	d	n	en	k	g	ng
	pea	bee	mom	item	tea	day	no	dozen	key	gay	sing
Syllabic				x				x			
Consonantal	x	x	x	x	x	x	x	x	x	x	x
Sonorant			x	x			x	x			x
Voiced		x	x	x		x	x	x		x	x
Continuent											
Nasal			x	x			x	x			x
Strident											
Lateral											
Distributed											
Affricate											
Labial	x	x	x	x							
Round											
Coronal					x	x	x	x			
Anterior	x	x	x	x	x	x	x	x			
High									x	x	x
Back									x	x	x
Low											
Tense											

	f	v	s	z	th	dh	sh	zh	ch	jh
	fin	van	sea	zoo	thin	then	she	azure	chin	jam
Syllabic										
Consonantal	x	x	x	x	x	x	x	x	x	x
Sonorant										
Voiced		x		x		x		x		x
Continuent	x	x	x	x	x	x	x	x		
Nasal										
Strident	x	x	x	x			x	x	x	x
Lateral										
Distributed							x	x	x	x
Affricate									x	x
Labial	x	x								
Round										
Coronal			x	x	x	x	x	x	x	x
Anterior	x	x	x	x	x	x				
High							x	x	x	x
Back										
Low										
Tense										

	l	el	r	er	w	j	hh
	lay	huddle	ray	bird	way	yes	hay
Syllabic		x		x			
Consonantal	x	x					
Sonorant	x	x	x	x	x	x	x
Voiced	x	x	x	x	x	x	
Continuent	x	x	x	x	x	x	x
Nasal							
Strident							
Lateral	x	x					
Distributed							
Affricate							
Labial					x		
Round			x	x	x		
Coronal	x	x	x	x			
Anterior	x	x	x	x			
High					x	x	
Back					x		
Low			x	x			
Tense							

	iy	ih	ey	eh	ae	uw	uh	ah	ow	ao	aa
	beet	bit	bait	bet	bat	boot	book	but	boat	taut	pot
Syllabic	x	x	x	x	x	x	x	x	x	x	x
Consonantal											
Sonorant											
Voiced											
Continuent											
Nasal											
Strident											
Lateral											
Distributed											
Affricate											
Labial											
Round						x	x		x	x	
Coronal											
Anterior											
High	x	x				x	x				
Back						x	x	x	x	x	x
Low					x					x	x
Tense	x		x			x			x		

Table A.1: *English phonemes and their distinctive features.*

Appendix B

Sample Lexicon and Rule Output

This file shows samples of the lexicon and rules that the system creates when run. This file came from showing the system 21 words and their past tense once each. The word format is:

`[` *pronunciation* `] =` *spelling* `(`*part of speech*`) Meaning:` *index*

`Rule List`

`---------`

`rule prob`

num. weight

And the rule format is:

`Rule index:` *num.*

APPLY TO *template in word format*

`Shift:` *amount*

CHANGES *template in word format*

Note that for the APPLY TO and CHANGES templates appear in the same format as words with one exception. Their pronunciation often does not have complete phonemes. Instead they will list a phoneme by its features, stating whether it possesses the feature (yes), does not possess the feature (no), or does not care about that feature (DC).

```
Lexicon
-------
#words = 21
#rules = 6


Words:
[ d ae n s ] = dance (Verb) Meaning: 120
Rule List
---------
rule    prob
0       1.0

[ d r aa p ] = drop (Verb) Meaning: 121
Rule List
---------
rule    prob
0       1.0

[ f ih k s ] = fix (Verb) Meaning: 122
Rule List
---------
rule    prob
0       1.0

[ k ih s ] = kiss (Verb) Meaning: 123
Rule List
---------
rule    prob
```

```
0       1.0

[ l ae f ] = laugh (Verb) Meaning: 124
Rule List
---------
rule    prob
0       1.0

[ l aa y k ] = like (Verb) Meaning: 125
Rule List
---------
rule    prob
0       1.0

[ l uh k ] = look (Verb) Meaning: 126
Rule List
---------
rule    prob
0       1.0

[ t ah ch ] = touch (Verb) Meaning: 127
Rule List
---------
rule    prob
0       1.0

[ w ao k ] = walk (Verb) Meaning: 128
Rule List
---------
rule    prob
```

```
0       1.0

[ ae n s er ] = answer (Verb) Meaning: 129
Rule List
---------
rule    prob
9       1.0

[ k ao l ] = call (Verb) Meaning: 130
Rule List
---------
rule    prob
9       1.0

[ k r aa y ] = cry (Verb) Meaning: 131
Rule List
---------
rule    prob
9       1.0

[ h ah g ] = hug (Verb) Meaning: 132
Rule List
---------
rule    prob
9       1.0

[ t er n ] = turn (Verb) Meaning: 133
Rule List
---------
rule    prob
```

```
9       1.0

[ ae d ] = add (Verb) Meaning: 134
Rule List
---------
rule    prob
14      1.0

[ n iy d ] = need (Verb) Meaning: 135
Rule List
---------
rule    prob
14      1.0

[ p ey n t ] = paint (Verb) Meaning: 136
Rule List
---------
rule    prob
14      1.0

[ w ey t ] = wait (Verb) Meaning: 137
Rule List
---------
rule    prob
14      1.0

[ d r ao ] = draw (Verb) Meaning: 138
Rule List
---------
rule    prob
```

17 1.0

[s ih ng] = sing (Verb) Meaning: 139

Rule List

----

rule	prob
18	1.0

[f iy d] = feed (Verb) Meaning: 140

Rule List

----

rule	prob
14	0.8
19	0.2

Rules:

Rule index: 0

[

syl	cons	son	voice	cont	nas	str	lat	dist
DC	DC	DC	DC	DC	No	No	DC	No
aff	lab	round	cor	ant	high	back	low	ten
No	DC	DC	DC	DC	DC	DC	DC	No

syl	cons	son	voice	cont	nas	str	lat	dist
DC	DC	DC	DC	DC	DC	No	No	No
aff	lab	round	cor	ant	high	back	low	ten
No	No	DC	DC	DC	DC	DC	DC	No

syl	cons	son	voice	cont	nas	str	lat	dist
No	Yes	No	No	DC	No	DC	No	DC
aff	lab	round	cor	ant	high	back	low	ten
DC	DC	No	DC	DC	DC	DC	No	No

] = ???? (Verb) Meaning: 0   Rule List Empty

Shift: 1

[t] = ???? (

Noun	Verb	Plural	Past
DC	DC	DC	Yes

) Meaning: 0   Rule List Empty

Rule index: 9

[

syl	cons	son	voice	cont	nas	str	lat	dist
No	DC	DC	DC	DC	DC	No	No	No
aff	lab	round	cor	ant	high	back	low	ten
No	No	DC	DC	DC	DC	DC	DC	No

syl	cons	son	voice	cont	nas	str	lat	dist
DC	DC	DC	DC	DC	No	DC	No	No
aff	lab	round	cor	ant	high	back	low	ten
No	No	DC	DC	DC	No	DC	DC	No

syl	cons	son	voice	cont	nas	str	lat	dist
DC	DC	DC	Yes	DC	DC	No	DC	No
aff	lab	round	cor	ant	high	back	low	ten
No	No	DC	DC	DC	DC	DC	DC	No

] = ???? (Verb) Meaning: 0   Rule List Empty

Shift: 1

[d] = ???? (

Noun	Verb	Plural	Past
DC	DC	DC	Yes

) Meaning: 0   Rule List Empty

Rule index: 14

[

syl	cons	son	voice	cont	nas	str	lat	dist
DC	DC	DC	DC	No	DC	No	No	No
aff	lab	round	cor	ant	high	back	low	ten
No	No	No	DC	DC	DC	No	DC	DC

syl	cons	son	voice	cont	nas	str	lat	dist
No	Yes	No	DC	No	No	No	No	No
aff	lab	round	cor	ant	high	back	low	ten
No	No	No	Yes	Yes	No	No	No	No

] = ???? (Verb) Meaning: 0   Rule List Empty

Shift: 2

[ih d] = ???? (

Noun	Verb	Plural	Past
DC	DC	DC	Yes

) Meaning: 0   Rule List Empty

Rule index: 17

[d r ao] = draw (Verb) Meaning: 138   Rule List Empty

Shift: 0

[

syl	cons	son	voice	cont	nas	str	lat	dist
DC	DC	DC	DC	DC	DC	DC	DC	DC
aff	lab	round	cor	ant	high	back	low	ten

DC	DC	DC	DC	DC	Yes	DC	No	Yes

] = ???? (

Noun	Verb	Plural	Past
DC	DC	DC	Yes

) Meaning: 0 Rule List Empty

Rule index: 18

[s ih ng] = sing (Verb) Meaning: 139 Rule List Empty

Shift: 0

[

syl	cons	son	voice	cont	nas	str	lat	dist
DC	DC	DC	DC	DC	DC	DC	DC	DC
aff	lab	round	cor	ant	high	back	low	ten
DC	DC	DC	DC	DC	No	DC	Yes	DC

DC] = ???? (

Noun	Verb	Plural	Past
DC	DC	DC	Yes

) Meaning: 0 Rule List Empty

Rule index: 19

[f iy d] = feed (Verb) Meaning: 140 Rule List Empty

Shift: 0

[

syl	cons	son	voice	cont	nas	str	lat	dist
DC	DC	DC	DC	DC	DC	DC	DC	DC
aff	lab	round	cor	ant	high	back	low	ten
DC	DC	DC	DC	DC	No	DC	DC	No

DC] = ???? (

Noun	Verb	Plural	Past
DC	DC	DC	Yes

```
) Meaning: 0   Rule List Empty
```

Bibliography

[1] The DARPA TIMIT Acoustic-Phonetic Speech Corpus. NIST Speech Disc CD1-1.1, 1990.

[2] *Contemporary Linguistics*, chapters 2 & 4. Third edition, 1997.

[3] Jean Berko. The child's learning of English morphology. *Diver, William and Martinet, Andree and Weinreich, Uriel*, 14:150–177, 1958.

[4] Joan L. Bybee and Dan I. Slobin. Why small children cannot change language on their own: Suggestions from the English past tense. In Anders Ahlqvist, editor, *Current Issues in Linguistic Theory: Papers from the Fifth International Conference on Historical Linguistics*, pages 29–37. John Benjamins, 1982.

[5] Courtney Cazden. The acquisition of noun and verb inflections. *Child Development*, 39:433–448, 1968.

[6] *Cognition*, 61, 1996. The entire issue is devoted to computational linguistics.

[7] Harold Clahsen, Monika Rothweiler, Andreas Woest, and Gary Marcus. Regular and irregular inflection in the acquisition of German noun plurals. *Cognition*, 45:225–255, 1992.

[8] Bruce Derwing. English pluralization: A testing ground for rule evaluation. In David J. Lipcoll, D. H. Lawrie, and A. H. Sameh, editors, *Experimental Linguistics*, Story Scientia Linguistics Series, pages 81–112. E. Story Scientia, Ghent, Belgium, 1980.

[9] Bruce Derwing and Wm. J. Baker. Rule learning and the English inflections (with special emphasis on the plural). In David J. Lipcoll, D. H. Lawrie, and A. H. Sameh, editors, *Experimental Linguistics*, Story Scientia Linguistics Series, pages 247–272. E. Story Scientia, Ghent, Belgium, 1980.

[10] Alison J. Elliot. *Child Language*. Cambridge Textbooks in Linguistics. Cambridge University Press, Cambridge [Eng], 1981.

[11] J. Hayes and Clark. Experiments on the segmentation of an artificial speech analogue. In John R. Hayes, editor, *Cognition and the development of language*. Wiley, New York, 1970.

[12] Henry Kucera and W. Nelson Francis. *Computational Analysis of Present-Day American English*. Brown University Press, Providence, RI, 1967.

[13] Eric H. Lenneberg, editor. *New Directions in the Study of Language*. MIT Press, Cambridge, Ma, 1964. Articles by Lenneberg, Miller, Ervin, and Brown & Bellugi.

[14] Charles X. Ling. Learning the past tense of English verbs: The symbolic pattern associator vs. connectionist models. *Journal of Aritificial Intelligence Research*, 1:209–229, 1994.

[15] Charles X. Ling and Marin Marinov. Answering the connectionist challenge: A symbolic model of learning the past tense of English verbs. *Cognition*, 49(3):235–290, December 1993.

[16] Brian MacWhinney. *The Acquisition of Morphophonology*, volume 43 of *Monographs of the Society for Research in Child Development*. University of Chicago Press, Chicago, 1978.

[17] Brian MacWhinney, editor. *Mechanisms of Language Acquisition*. Erlbaum Associates, Hillsdale, NJ, 1987. Papers from the 20th Carnegie-Mellon Symposium on Cognition, includes papers by McWhinney, Macken, and Bowerman.

[18] Brian MacWhinney. *The CHILDES Database: Tools for Analyzing Talk, Vol 2L the Database*. Erlbaum Associates, Mahwah, NJ, third edition, 2000.

[19] Brian MacWhinney and Jared Leinbach. Implementations are not conceptualizations: Revising the verb learning model. *Cognition*, 40:121–157, 1991.

[20] G. F. Marcus, U. Brinkmann, H. Clahsen, R. Wiese, and S Pinker. German inflection: The exception that proves the rule. *Cognitive Psychology*, 29:189–256, 1995.

[21] Gary Marcus. Negative evidence in language acquisition. *Cognition*, 46:53–85, 1993.

[22] Gary Marcus, Steven Pinker, Michael Ullman, Michelle Hollander, T. John Rosen, and Fei Xu. *Overregularization in Language Acquisition*, volume 57 of *Monographs of the Society for Research in Child Development*. University of Chicago Press, Chicago, 1992.

[23] Tom M. Mitchell. Generalization as search. *Artificial Intelligence*, 18:203–226, 1982.

[24] Raymond J. Mooney and Mary Elaine Califf. Induction of first-order decision lists: Results on learning the past tense of English verbs. *Cognition*, 49(3):235–290, December 1993.

[25] Raymond J. Mooney and Mary Elaine Califf. Learning the past tense of English verbs using inductive logic programming. *Symbolic, Connectionist, and Statistical Approaches to Learning for Natural Language Processing*, 1996.

[26] Steven Pinker. Language acquisition. In L. Gleitman and M. Liberman, editors, *An Invitation to Cognitive Science*. MIT Press, Cambridge, Ma.

[27] Steven Pinker. Why the child holded the baby rabbit: a case study in language acquisition. In L. Gleitman and M. Liberman, editors, *An Invitation to Cognitive Science*. MIT Press, Cambridge, Ma.

[28] Steven Pinker. *Words and Rules, the Ingredients of Language*. Basic Books, New York, first edition, 1999.

[29] Steven Pinker and Alan Prince. On language and connectionism: Analysis of a parallel distributed processing model of language acquisition. *Cognition*, 28:73–193, 1988.

[30] Sandeep Prasada and Steven Pinker. Generalisation of regular and irregular morphological patterns. *Language and Cognitive Processes*, 8(1):1–56, 1993.

[31] Andrew Radford. *Transformational Syntax: a student's guide to Chompsky's Extended Standard Theory*. Cambridge textbooks in linguistics. Cambridge University Press, Cambridge [Eng], 1981.

[32] D. Rumelhart and J. McClelland. On learning the past tense of English verbs. implicit rules or parallel distributed processing? In D. Rumelhart, J. McClelland, and the PDP Research Group, editors, *Parallel Distributed Processing: Explorations in the microstructure of Cognition*. MIT Press, Cambridge, Ma.

[33] Patrick Henry Winston. Learning structural descriptions from examples. In Patrick Henry Winston, editor, *The Psychology of Computer Vision*. MaGraw-Hill, New York, 1975.

[34] Fei Xu and Steven Pinker. Weird past tense forms. *Journal of Child Language*, 22:531–556, 1995.

[35] Charles D. Yang. Words, rules, and competitions. February 1999.

[36] Charles D. Yang. Language acquisition as grammar competition: A synthesis of universal grammar and learning. 2000.

[37] Kenneth Yip and Gerald Jay Sussman. A computational model for the acquisition and use of phonological knowledge. AI Memo 1575, MIT AI Laboratory, 1996.

[38] Kenneth Yip and Gerald Jay Sussman. Sparse representations for fast, one-shot learning, 1997.