# Learning Probabilistic Paradigms for Morphology in a Latent Class Model

**Erwin Chan**
Dept. of Computer and Information Science
University of Pennsylvania
Philadelphia, PA 19104
echan3@seas.upenn.edu

## Abstract

This paper introduces the *probabilistic paradigm*, a probabilistic, declarative model of morphological structure. We describe an algorithm that recursively applies Latent Dirichlet Allocation with an orthogonality constraint to discover morphological paradigms as the latent classes within a suffix-stem matrix. We apply the algorithm to data preprocessed in several different ways, and show that when suffixes are distinguished for part of speech and allomorphs or gender/conjugational variants are merged, the model is able to correctly learn morphological paradigms for English and Spanish. We compare our system with Linguistica (Goldsmith 2001), and discuss the advantages of the probabilistic paradigm over Linguistica's signature representation.

## 1 Introduction

In recent years researchers have addressed the task of unsupervised learning of declarative representations of morphological structure. These models include the signature of (Goldsmith 2001), the conflation set of (Schone and Jurafsky 2001), the paradigm of (Brent et. al. 2002), and the inflectional class of (Monson 2004). While these representations group morphologically related words in systematic ways, they are rather different from the paradigm, the representation of morphology in traditional grammars. A paradigm lists the prototypical morphological properties of lexemes belonging to a particular part of speech (POS) category; for example, a paradigm for regular English verbs would include the suffixes {$,ed$,ing$,s$}[1]. Hand-built computational implementations of paradigms as inheritance hierarchies include DATR (Evans and Gazdar 1996) and Functional Morphology (Forsberg and Ranta 2004). The two principal ways in which learned models have differed from paradigms are that: 1) they do not have POS types, and 2) they are not abstractions that generalize beyond the words of the input corpus.

There are important reasons for learning a POS-associated, paradigmatic representation of morphology. Currently, the dominant technology for morphological analysis involves mapping between inflected and base of forms of words with finite-state transducers (FSTs), a procedural model of morphological relations. Rewrite rules are handcrafted and compiled into FSTs, and it would be beneficial if these rules could be learned automatically. One line of research in computational morphology has been directed towards learning finite-state mapping rules from some sort of paradigmatic structure, where all morphological forms and POS types are presumed known for a set of lexemes (Clark 2001, Kazakov and Manandhar 2001, Oflazer et. al. 2001, Zajac 2001, Albright 2002). This can be accomplished by first deciding on a base form, then learning rules to convert other forms of the paradigm into this base form. If one could develop an unsupervised algorithm for learning paradigms, it could serve as the input to rule-learning procedures, effectively leading to an entirely unsupervised system for learning FSTs from raw data. This is our long-term goal.

[1] $ is the null suffix.



*Proceedings of the Eighth Meeting of the ACL Special Interest Group on Computational Phonology at HLT-NAACL 2006*, pages 69–78,
New York City, USA, June 2006. 

An alternative approach is to skip the paradigm formulation step and construct a procedural model directly from raw data. (Yarowsky and Wicentowski 2000) bootstrap inflected and base forms directly from raw data and learn mappings between them. Their results are quite successful, but the morphological information they learn is not structured as clearly as a paradigmatic model. (Freitag 2005) constructs a morphological automaton, where nodes are clustered word types and arcs are suffixation rules.

This paper addresses the problem of finding an organization of stems and suffixes as probabilistic paradigms (section 2), a model of morphology closer to linguistic notion of paradigm than previously proposed models. We encode the morphological structure of a language in a matrix containing frequencies of words, and formulate the problem of learning paradigms as one of finding latent classes within the matrix. We present a recursive LDA, a learning algorithm based on Latent Dirichlet Allocation (section 3), and show that under certain conditions (section 5), it can correctly learn morphological paradigms for English and Spanish. In section 6, we compare the probabilistic paradigm to the signature model of (Goldsmith 2001). In section 7, we sketch some ideas for how to make our system more unsupervised and more linguistically adequate.

We assume a model of morphology where each word is the concatenation of a stem and a single suffix representing all of the word's morphological and POS properties. Although this is a very simplistic view of morphology, there are many hitherto unresolved computational issues for learning even this basic model, and we consider it necessary to address these issues before developing more sophisticated models. For a stem/suffix representation, the task of learning a paradigm from raw data involves proposing suffixes and stems, proposing segmentations, and systematically organizing stems and suffixes into classes. One difficulty is suffix allomorphy: a suffix has multiple forms depending on its phonological environment (e.g. `s$/es$`). Another problem is suffix categorial ambiguity (`s$` is ambiguous for noun and verb uses). Finally, lexemes appear in only a subset of their potential forms, due to sparse data. An unsupervised learner needs to be able to handle all of these difficulties in order to discover abstract paradigmatic classes.

In this paper, we are primarily interested in how the co-occurrence of stems and suffixes in a corpus leads them to be organized into paradigms. We use data preprocessed with correct segmentations of words into stems and suffixes, in order to focus on the issue of determining what additional knowledge is needed. We demonstrate that paradigms for English and Spanish can be successfully learned when tokens have been assigned POS tags and allomorphs or gender/conjugational variants are given a common representation. Our learning algorithm is not supervised since the target concept of gold standard "input" POS category of stems is not known, but rather it is an unsupervised algorithm that relies on preprocessed data for optimal performance.

## 2 The Probabilistic Paradigm

We introduce the *probabilistic paradigm*, a probabilistic, declarative model of regular morphology. The probabilistic paradigm model consists of three matrices: the data matrix D, the morphological probabilities matrix M, and the lexical probabilities matrix L. Let `m` be the number of stems, `n` the number of stems, and `p` the number of paradigms. The D matrix encodes the joint distribution of lexical and morphological information in a corpus. It is of size `m x n`, and each cell contains the frequency of the word formed by concatenating the appropriate stem and suffix. The M matrix is of size `m x p`, and each column contains the conditional probabilities of each suffix given a paradigm. The L matrix is of size `p x n`, and contains the conditional probabilities of each paradigm given a stem. Each suffix should belong to exactly one paradigm, and the suffixes of a particular paradigm should be conditionally independent. Each column of the M matrix defines a *canonical paradigm*, a set of suffixes that attach to stems associated with that paradigm. A *lexical paradigm* is the full set of word forms for a particular stem, and is an instantiation of the canonical paradigm for a particular stem.

The probabilistic paradigm is not well-developed as the usual notion of "paradigm" in linguistics. First, the system employs no labels such as "noun", "plural", "past", etc. Second, probabilistic paradigms have only a top-level categorization; induced “verb” paradigms, for example, are not substructured into different tenses or conjuga-

tions. Third, we do not distinguish between inflectional and derivational morphology; traditional grammars place derived forms in separate lexical paradigms. Fourth, we do not handle syncretism, where one suffix belongs in multiple slots of the paradigm. Fifth, we do not yet not handle irregular and sub-regular forms. Despite these drawbacks, our paradigms have an important advantage over traditional paradigms, in being probabilistic and therefore able to model language usage.

## 3 Learning the probabilistic paradigm in a latent class model

We learn the parameters of the probabilistic paradigm model by applying a dimensionality reduction algorithm to the D matrix, in order to produce the M and L matrices. This reduces the size of the representation from `m*n` to `m*p + p*n`. The main idea is to discover the latent classes (paradigms) which represent the underlying structure of the input matrix. This handles two important problems: 1) that words occur in a subset of their possible morphological forms in a corpus, and 2) that the words formed from a particular stem can belong to multiple POS categories. The second problem can be quantified as follows: in our English data, 14.3% of types occur with multiple open-class base POS categories, accounting for 56.5% of tokens; for Spanish, 13.7% of types, 37.8% of tokens.

### 3.1 LDA model for morphology

The dimensionality reduction algorithm that we employ is Latent Dirichlet Allocation (LDA) (Blei et. al. 2003). LDA is a generative probabilistic model for discrete data. For the application of topic discovery within a corpus of documents, a document consists of a mixture of underlying topics, and each topic consists of a probability distribution over the vocabulary. The topic proportions are drawn from a Dirichlet distribution, and the words are drawn from a multinomial over the topic. Probability distributions of documents and words are conditionally independent of topics. LDA produces two non-negative parameter matrices, Gamma and Beta: Gamma is the matrix of Dirichlet posteriors, encoding the distribution of documents and topics; Beta encodes the distribution of words and topics.

The mapping of the data structures of LDA to the probabilistic paradigm is as follows. The document-word matrix is analogous to the suffix-stem D matrix. For morphology, a "document" is a multiset of tokens in a corpus, such that each of those tokens decomposes into a stem and a specified suffix. Different underlying canonical paradigms ("topics") can be associated with suffixes, and each canonical paradigm allows a set of stems ("words"). For a suffix-stem ("document-word") matrix of size `m x n` and `k` latent classes, the Gamma matrix is of size `m x k`, and the Beta matrix is of size `k x n`. The Gamma matrix, normalized by column, is the M matrix, and the Beta matrix, normalized by row, is the L matrix.

### 3.2 Recursive LDA

One standard issue in using these types of algorithms is selecting the number of classes. To deal with this, we have formulated a recursive wrapper algorithm for LDA that accomplishes a divisive clustering of suffixes. LDA is run at each stage to find the local Gamma and Beta matrices. To split the suffixes into two classes, we assign each suffix to the class for which its probability is greater, by examining the Gamma matrix. The input matrix is then divided into two smaller matrices based on this split, and the algorithm continues with each submatrix. The result is a binary tree describing the suffix splits at each node.

To construct a classification of suffixes into paradigms, it is necessary to make a cut in the tree. Assuming that suffix splits are optimal, we start at the root of the tree and go down until reaching a node where there is sufficient uncertainty about which class a suffix should belong to. A good split of suffixes is one where the vectors of probabilities of suffixes given a class are orthogonal; we can find such a split by minimizing the cosine of the two columns of the node's Gamma matrix (we call this the "Gamma cosine"). Thus, a node at which suffixes should not be split has a high Gamma cosine, and when encountering such a node, a cut should be made. The suffixes below this node are grouped together as a paradigm; tree structure below the cut node is ignored. In our experiments we have selected thresholds for the Gamma cosine, but we do not know if there is a single value that would be successful cross-linguistically. After the tree has been cut, the Gamma and Beta matrices for ancestor nodes are normalized and combined to form the M and L matrices for the language.

Another issue is dealing with suboptimal solutions. Random initializations of parameters lead the EM training procedure to local maxima in the solution space, and as a result LDA produces differing suffix splits across different runs. To get around this, we simply run LDA multiple times (25 in our experiments) and choose the solution that minimizes the Gamma cosine.

We also experimented with minimizing the Beta cosine. The Beta matrix represents stem ambiguity with respect to a suffix split. Since there are inherently ambiguous stems, one should not expect the Beta cosine value to be extremely low. Minimizing the Beta cosine sometimes made the Beta matrix "too disambiguated" and forced the representation of ambiguity into Gamma matrix, thereby inflating the Gamma cosine and causing incorrect classifications of suffixes.

## 4 Data

We conducted experiments on English and Spanish. For English, we chose the Penn Treebank (Marcus et. al. 1993), which is already POS-tagged; for Spanish, we chose an equivalent-sized portion of newswire (Graff and Galegos 1999), POS-tagged by the FreeLing morphological analyzer (Carreras et. al. 2004). We restricted our data to nouns, verbs, adjectives, and adverbs. Words that did not follow canonical suffixation patterns for their POS category (irregulars, foreign words, incorrectly tagged words, etc.) were excluded. We segmented each word into stem and suffix for a specified set of suffixes. Rare suffixes were excluded, such as many English adjective-forming suffixes and Spanish 2nd person plural forms. Stems were not lemmatized, with the result that there can be multiple stem variants of a particular lemma, as with the words `stemm.ing$` and `stem.s$`. Tokens were not disambiguated for word sense. Stems that occurred with only one suffix were excluded.

We use several different representations of suffixes in constructing the data matrices: 1) merged, labeled suffixes; 2) merged, unlabeled suffixes; 3) unmerged, unlabeled suffixes. For unmerged suffixes, allomorphs[2] are represented in their original spelling. A merged suffix is a common representation for the multiple surface manifestations of an underlyingly identical suffix. Suffixes also can be unlabeled, or labeled with base POS tags. For an example, a verb `created` would be segmented as `create.d$` with an unmerged, labeled suffix, or `create.d/ed$V` with a merged, labeled suffix. Labels disambiguate otherwise categorically ambiguous suffixes.

The gold standard for each language lists the suffixes that belong to a paradigm for stems of a particular POS category. We call this the "input" POS category, which is not indicated in annotations and is the concept to be predicted. This should be differentiated from the "output" POS labels on the suffixes: for example, `ly$R` attaches to stems of the input category “adjective”. Each suffix is an atomic entity, so the system actually has no concept of output POS categories. All that we require is that distinct suffixes are given distinct symbols.

In the English gold standard (Table 1), each slashed pair of suffixes denotes one merged form; the unmerged forms are the individual suffixes. `ally$R` is the suffix `ly$R` preceded by an epenthetic vowel, as in the word `basically`. In the Spanish gold standard (Table 2), each slashed group of suffixes corresponds to one merged form. For adjectives and nouns, `a$` and `o$` are feminine and masculine singular forms, and `as$` and `os$` are the corresponding plurals. `$` and `s$` do not have gender; `es$` is a plural allomorph. `mente/amente$R` is a derivational suffix. The first two groups of verbal suffixes are past participles, agreeing in number and gender. For the other verb forms, when three are listed they correspond to forms for the 1st, 2nd, and 3rd conjugations. When there are two, the first is for the 1st conjugation, and the other is identical for the 2nd and 3rd. `o$V` has the same form across all three conjugations.

| | |
|---|---|
| Adjectives: | $A, d/ed$A, r/er$A, ally/ly$R |
| Nouns: | $N, 's$N, es/s$N |
| Verbs: | $V, d/ed$V, es/s$V, ing$V, ing$A, ing$N, r/er$N |

Table 1. Gold standard for English

[2] We abuse the standard usage of the term "allomorph" to include gender and conjugational variants.

| | |
|---|---|
| Adjectives: | a/o/$A, as/os/es/s$A, |
| | mente/amente$R |
| Nouns: | a/o/$N, as/os/es/s$N |
| Verbs: | ada/ida/ado/ido$V, |
| | adas/idas/ados/idos$V, ando/iendo$V, |
| | ar/er/ir$V, o$V, as/es$V, a/e$V, |
| | amos/emos/imos$V, an/en$V, aba/ía$V, |
| | ábamos/íamos$V, aban/ían$V, |
| | aré/eré/iré$V, ará/erá/irá$V, |
| | aremos/eremos/iremos$V, arán/erán/irán$V, |
| | é/í$V, ó/ió$V, aron/ieron$V, |
| | aría/ería/iría$V, arían/erían/irían$V |

Table 2. Gold standard for Spanish

## 5 Experiments

### 5.1 Merged, labeled suffixes

Figure 1 shows the recursion tree for English data preprocessed with merged, labeled suffixes. To produce a classification of suffixes into paradigms, we start at the root and go down until reaching nodes with a Gamma cosine greater than or equal to the threshold. The cut for a threshold of .0009 produces three paradigms exactly matching the gold standard for verbs, adjectives, and nouns, respectively. Table 3 shows the complete M matrix, which contains suffix probabilities for each paradigm. Table 4 shows a portion of the L matrix, which contains the probabilities of stems belonging to paradigms. We list the stems that are most ambiguous with respect to paradigm membership (note that this table does not specify the *words* that belong to each category, only their *stems*).

| | "Verb" | "Adj" | "Noun" |
|---|---|---|---|
| $A | 0.000 | 0.829 | 0.000 |
| d/ed$A | 0.020 | 0.000 | 0.000 |
| r/er$A | 0.000 | 0.033 | 0.000 |
| ing$A | 0.008 | 0.000 | 0.000 |
| $N | 0.000 | 0.000 | 0.706 |
| 's$N | 0.000 | 0.000 | 0.036 |
| r/er$N | 0.037 | 0.000 | 0.000 |
| ing$N | 0.065 | 0.000 | 0.000 |
| es/s$N | 0.000 | 0.000 | 0.257 |
| ally/ly$R | 0.000 | 0.138 | 0.000 |
| $V | 0.342 | 0.000 | 0.000 |
| d/ed$V | 0.284 | 0.000 | 0.000 |
| ing$V | 0.133 | 0.000 | 0.000 |
| es/s$V | 0.110 | 0.000 | 0.000 |

Table 3. M matrix for English merged, labeled suffixes. Columns: `p(suff|paradigm)`.

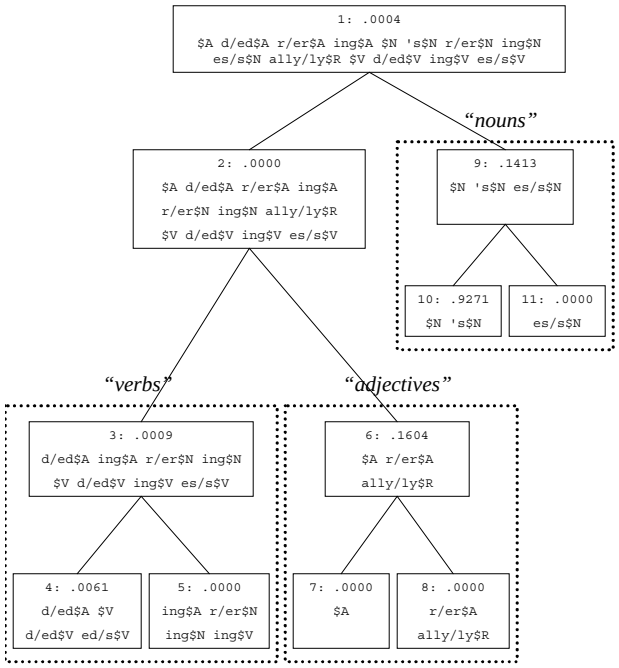


Figure 1. Recursion tree for English merged, labeled suffixes. Each node shows its current suffix set, and the Gamma cosine value for the split. Dotted lines indicate paradigms for a Gamma cosine threshold of .0009.

| | "Verb" | "Adj" | "Noun" |
|---|---|---|---|
| reset | 0.333 | 0.292 | 0.375 |
| blunt | 0.445 | 0.278 | 0.277 |
| calm | 0.417 | 0.375 | 0.209 |
| total | 0.312 | 0.462 | 0.226 |
| clean | 0.478 | 0.319 | 0.203 |
| parallel | 0.222 | 0.278 | 0.500 |
| alert | 0.500 | 0.222 | 0.277 |
| sound | 0.483 | 0.184 | 0.333 |
| compound | 0.372 | 0.171 | 0.457 |
| pale | 0.417 | 0.417 | 0.166 |
| fine | 0.254 | 0.230 | 0.516 |
| premier | 0.235 | 0.235 | 0.529 |
| brief | 0.175 | 0.524 | 0.301 |
| polish | 0.250 | 0.556 | 0.194 |
| ski | 0.378 | 0.108 | 0.513 |
| fake | 0.200 | 0.600 | 0.200 |
| light | 0.092 | 0.427 | 0.481 |
| foster | 0.226 | 0.161 | 0.613 |
| bottom | 0.107 | 0.304 | 0.589 |
| repurchase | 0.333 | 0.095 | 0.571 |

Table 4. Portion of L matrix for English merged, labeled suffixes, sorted by lowest entropy. Columns: `p(paradigm|stem)`.

Next, we examine the morphological and lexical conditional probabilities in the M and L matrices. It is possible that even though the correct classification of suffixes into paradigms was learned, the probabilities may be off. Table 5 shows, however, that the M and L matrices are an extremely accurate approximation of the true morphological and lexical probabilities. We have included statistics for the corresponding Spanish experiment; the paradigms that were discovered for Spanish also match the gold standard.

| | English | Spanish |
|---|---|---|
| # suffixes | 14 | 26 |
| # stems | 7315 | 5115 |
| CRE M | .0002 bits | .0003 bits |
| CRE L | .0006 bits | .0020 bits |

Table 5. Comparison of M and L matrices with true morphological and lexical probabilities, by conditional relative entropy (CRE).

### 5.2 Unmerged, labeled suffixes

The next experiments tested the effect of allomorphy on paradigm discovery, using data where suffixes are labeled but not merged. There are competing pressures at work in determining how allomorphs are assigned to paradigms: on the one hand, the disjointedness of stem sets for allomorphs would tend to place them in separate paradigms; on the other hand, if those stem sets have other suffixes in common that belong to the same paradigm, the allomorphs might likewise be placed in that paradigm. In our experiments, we found that there was much more variability across runs than in the merged suffix cases. In English, for example, the suffix `es$N` was sometimes placed in the "verb" paradigm, although the maximally orthogonal solution placed it in the “noun” paradigm.

Figure 2 shows the recursion tree and paradigms for Spanish. Gold standard noun and adjective categories are fragmented into multiple paradigms in the tree. Although nouns have a common parent node (2), the nouns of the different genders are placed in separate paradigms -- this is because a noun can have only one gender. The verbs are all in a single paradigm (node 10). Node 11 contains all the first-conjugation verbs, and node 12 contains all the second/third-conjugation verbs. The reason that they are not in separate paradigms is that `a$V` is shared by stems of all three conjugations, which leads to a split that is not quite orthogonal.

The case of adjectives is the most interesting. Gendered and non-gendered adjective stems are disjoint, so adjectives appear in two separate subtrees (nodes 4, 13). In node 4, the gender-ambiguous plural `es$A` is in conflict with the plural `s$A`, but it would conflict with two plurals `as$A` and `os$A` if it were placed in node 13. `amente$R` appears together in node 14 because it shares stems with the feminine adjectives. `amente$R` also shares stems with verbs, as it is also the derivational suffix which attaches to verbal past participles in the feminine "`a`" form. This is probably why the group of adjectives at node 13 is a sister to the verb nodes. The allomorph `mente$R` attaches to non-gendered adjectives, and is thus in the first adjective group.

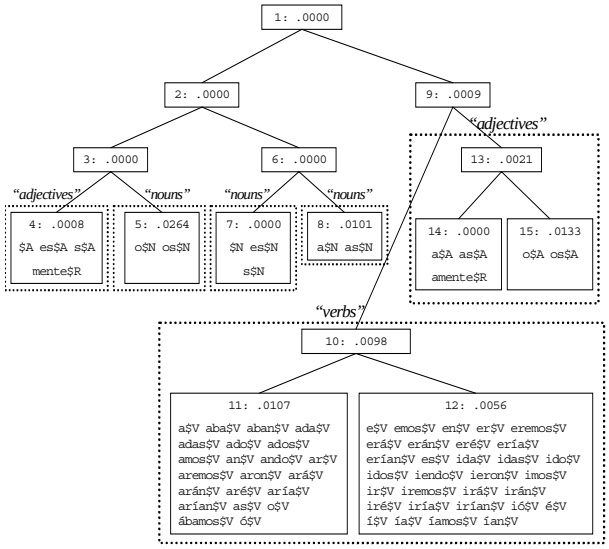


Figure 2. Recursion tree for Spanish, unmerged, labeled suffixes, with Gamma cosine values. Dotted lines indicate paradigms for a Gamma cosine threshold of .0021.

### 5.3 Unmerged, unlabeled suffixes

The case of unmerged, unlabeled suffixes is not as successful. In the Gamma matrix for the root node (Table 6), there is no orthogonal division of the suffixes, as indicated by the high Gamma cosine value of .1705. Despite this, the algorithm has discovered useful information. There is a subpara-

digm of unambiguous suffixes {'s$,ally$}, and another of {d$,ed$,ing$,r$}. The other suffixes ($,er$,es$,ly$,s$) are ambiguous. The ambiguity of ly$ seems to be a secondary effect: since adjectives with the null suffix $ are found to be ambiguous, ly$ is likewise ambiguous.

| | | |
|---|---|---|
| $ | [0.9055] | 0.0703 |
| 's$ | [0.0351] | 0.0000 |
| ally$ | [0.0007] | 0.0000 |
| d$ | 0.0000 | [0.1139] |
| ed$ | 0.0000 | [0.1332] |
| er$ | [0.0087] | 0.0084 |
| es$ | [0.0089] | 0.0001 |
| ing$ | 0.0000 | [0.1176] |
| ly$ | 0.0033 | [0.0603] |
| r$ | 0.0000 | [0.0198] |
| s$ | 0.0378 | [0.4764] |

Table 6. Gamma matrix for root node, English, unmerged, unlabeled suffixes; the categorization is shown with brackets. Columns indicate p(suffix|class).

## 6 Comparison with Linguistica

In this section, we compare our system with Linguistica[3] (Goldsmith 2001), a freely available program for unsupervised discovery of morphological structure. We focus our attention on Linguistica's representation of morphology, rather than the algorithm used to learn it. Linguistica takes a list of word types, proposes segmentations of words into stems and suffixes, and organizes them into signatures. A *signature* is a non-probabilistic data structure that groups together all stems that share a common set of suffixes. Each stem belongs to exactly one signature, and the set of suffixes for each signature is unique. For example, running Linguistica on our raw English text, there is a signature {$,ful$,s$} for the stems {resource, truth, youth}, indicating the morphology of the words {resource$, truth$, youth$, resourceful$, truthful$, youthful$, resources$, truths$, youths$}. There are no POS types in the system. Thus, even for a prototypically "noun" signature such as {$,'s$}, it is quite possible that not all of the words that the signature represents are actually nouns. For example, the word structure$ is in this signature, but occurs both as a noun (59 times) and a verb (2 times) in our corpus.

The signature model can be derived from the suffix-stem data matrix, by first converting all positive counts to 1, and then placing in separate groups all the stems that have the same 0/1 column pattern. Another way to view the signature is as a special case of the probabilistic paradigm where all probabilities are restricted to being 0 or 1, for if this were so, the only way to fit the data would be to let there be a canonical paradigm for every different subset of suffixes that some stem appears with. In theory, it is possible for the number of signatures to be exponential in the number of suffixes; in practice, Linguistica finds hundreds of signatures for English and Spanish. Although there has been work on reducing the number of signatures (Goldwater and Johnson 2004; Hu et. al. 2005, who report a reduction of up to 30%), the number of remaining signatures is still two orders of magnitude greater than the number of canonical paradigms we find. The simplest explanation for this is that a suffix can be listed many times in the different signatures, but only has one entry in the M matrix of the probabilistic paradigm.

It is important for a natural language system to handle out-of-vocabulary words. A signature does not predict the forms of potential but unseen forms of stems. To some extent Linguistica could accommodate this, as it identifies when one signature's suffixes are a proper subset of another's, but it does not handle cases where suffixes are partially overlapping. One principal advantage of the probabilistic paradigm is that the canonical paradigm allows the instantiation of a lexical paradigm containing a complete set of predicted word forms for a stem.

Since Linguistica is a system that starts from raw text, it may seem that it cannot be directly compared to our work, which assumes that segmentations and suffixes are already known. However, it is possible to run Linguistica on our data by doing further preprocessing. We rewrite the corpus in such a way that Linguistica can detect correct morphological and POS information for each token. Each token is replaced by the concatenation of its stem, the dummy string 12345, and a single-character encoding of its merged suffix. For example, the token accelerate.d/ed$V is mapped to accelerate12345D, where D represents d/ed$V. The omnipresence of the dummy string enables

[3] http://linguistica.uchigago.edu

Linguistica to discover all the desired stems and suffixes, but no more. By mapping the input corpus in this way, we can examine the type of grammar that Linguistica would find if it knew the information that we have assumed in the previous experiments. Linguistica found 565 signatures from the "cooked" English data (Figure 3). 50% of word types are represented by the first 13 signatures.

```
1. { $N, es/s$N } 1540
   abortion absence accent acceptance
   accident accolade accommodation
2. { $N, 's$N } 1168
   aba abbie abc academy achenbaum aclu
   adams addington addison adobe
3. { $N, 's$N, es/s$N } 224
   accountant acquisition actor
   administration airline airport alliance
5. { $A, ally/ly$R } 319
   abrupt absolute abundant accurate
   actual additional adequate adroit
6. { $A, $N, es/s$N } 173
   abrasive acid activist adhesive adult
   afghan african afrikaner aggregate
7. { $V, d/ed$V, es/s$V } 135
   abate achieve administer afflict
   aggravate alienate amass apologize
9. { $V, d/ed$V, ing$V, es/s$V } 73
   abound absorb adopt applaud assert
   assist attend attract avert avoid
13. { $N, $V, d/ed$V, es/s$N, es/s$V } 44
    advocate amount attribute battle
    bounce cause compromise decline
```

Figure 3. Selected top signatures for merged, labeled suffix English data. Each signature shows the suffix set, number of stems, and several example stems. Ranking is by `log(num stems)* log(num suffixes)`.

We have formulated two metrics to evaluate the quality of a collection of signatures or paradigms. Ideally, all suffixes of a particular signature would be of the same category, and all the words of a particular category would be contained within one signature. *POS fragmentation* measures to what extent the words of an input POS category are scattered across different signatures. It is the average number of bits required to encode the probability distribution of some category's words over signatures. *Signature impurity* measures the extent to which the suffixes of a signature are of mixed input POS types. It is the expected value of the number of bits required to encode the probability distribution of some signature's suffixes over input POS categories. Table 7 shows that, according to these metrics, the signature does not organize morphological information as efficiently as probabilistic paradigms[4]. Linguistica's impurity scores are reasonably low because many of the signatures with the most stems are categorically homogeneous. Fragmentation scores show that the placement of the majority of words within top signatures offsets the scattering of a POS category's suffixes over many signatures.

(1) POS fragmentation =

$$\left[\sum_{P} h(p(S \mid \text{words of } P))\right] \Big/ \mid P \mid$$

(2) Signature impurity =

$$\left[\sum_{S} S.\text{numstems} \times h(p(P \mid S))\right] \Big/ \sum_{S} S.\text{numstems}$$

*h*: entropy
P: input POS categories
S: signatures / paradigms

| | Linguistica | Recursive LDA |
|---|---|---|
| English fragmentation | 5.422 bits | 0 bits |
| English impurity | .404 bits | 0 bits |
| Spanish fragmentation | 6.084 bits | 0 bits |
| Spanish impurity | .332 bits | 0 bits |

Table 7. Comparison of Linguistica and recursive LDA on merged, labeled suffix data. The maximum possible impurity for 3 POS categories is $log_2(3) = 1.585$ bits.

Finally, a morphological grammar should reflect the general, abstract morphological structure of the language from which a corpus was sampled. To test for consistency of morphological grammars across corpora, we split our cooked English data into two equal parts. Linguistica found 449 total signatures for the first half and 462 for the second. 296 signatures were common to both (in terms of the suffixes contained by the signatures). Of the 3506 stems shared by both data sets, 1831 (52.2%) occurred in the same signature. Of the top 50 signatures for each half-corpus, 45 were in common, and 1651 of 2403 shared stems (68.7%) occurred in the same signature. Recursive LDA found the

[4] Our scores would not be so good if we had chosen a poor Gamma cosine threshold value for classification. However, Linguistica's scores cannot be decreased, as there is only one signature model for a fixed set of stems and suffixes.

same canonical paradigms for both data sets (which matched the gold standard). Differences in word counts between the corpus halves altered stem inventories and lexical probabilities, but not the structure of the canonical paradigms. Our system thus displays a robustness to corpus choice that does not hold for Linguistica.

## 7 Future Work

This section sketches some ideas for future work to increase the linguistic adequacy of the system, and to make it more unsupervised.

1. Bootstrapping: for fully unsupervised learning, we need to hypothesize stems and suffixes. The output of recursive LDA indicates which suffixes may be ambiguous. To bootstrap a disambiguator for the different categorial uses of these suffixes, one could use various types of distributional information, as well as knowledge of partial paradigmatic structure for non-ambiguous suffixes.
2. Automated detection of cut nodes: currently the system requires that the user select a Gamma cosine threshold for extracting paradigms from the recursion tree. We would like to automate this process, perhaps with different heuristics.
3. Suffix merging and formulation of generation rules: when we decide that two suffixes should be merged (based on some measures of distributional similarity and word-internal context), we also need to formulate phonological (i.e., spelling) rules to determine which surface form to use when instantiating a form from the canonical paradigm.
4. Non-regular forms: we can take advantage of empty cells in the data matrix in order to identify non-regularities such as suppletives, stem variants, semi-regular subclasses, and suffix allomorphs. If the expected frequency of a word form (as derived from the M matrix and frequency of a stem) is relatively high but the value in the D matrix is zero, this is evidence that a non-regular form may occupy this cell. Locating irregular words could use methods similar to those of (Yarowsky and Wicentowski 2000), who pair irregular inflections and their roots from raw text. Stem variants and allomorphic suffixes could be detected in a similar manner, by finding sets of stems/suffixes with mutually exclusive matrix entries.
5. Multiple morphological properties per word: we currently represent all morphological and POS information with a single suffix. The learning algorithm and representation could perhaps be modified to allow for multiple morphological properties. One could perform recursive LDA on a particular morphological property, then take each of the learned paradigms and perform recursive LDA again, but for a different morphological property. This method might discover Spanish conjugational classes as subclasses within “verbs”.

## 8 Discussion

This paper has introduced the probabilistic paradigm model of morphology. It has some important benefits: it is an abstract, compact representation of a language's morphology, it accommodates lexical ambiguity, and it predicts forms of words not seen in the input data.

We have formulated the problem of learning probabilistic paradigms as one of discovering latent classes within a suffix-stem count matrix, through the recursive application of LDA with an orthogonality constraint. Under optimal data conditions, it can learn the correct paradigms, and also models morphological and lexical probabilities extremely accurately. It is robust to corpus choice, so we can say that it learns a morphological grammar for the *language*. This is a new application of matrix factorization algorithms, and an usual one: whereas in document topic modeling, one tries to find that a document consists of multiple topics, we want to find orthogonal decompositions where each suffix (document) belongs to only one input POS category (topic).

We have demonstrated that the algorithm can successfully learn morphological paradigms for English and Spanish under the conditions that segmentations are known, categorically ambiguous suffixes have been distinguished, and allomorphs have been merged. When suffixes have not been merged, there is a tendency to place allomorphic variants in different paradigms. The algorithm is the least successful in the unmerged, unlabeled case, as ambiguous suffixes do not allow for a clear split of suffixes into paradigms. However, the program output indicates which suffixes are potentially ambiguous or unambiguous, and this information could be used by bootstrapping procedures for suffix disambiguation.

Some of the behavior of the learning algorithm can be explained in terms of several constraints. First, LDA assumes conditional independence of

documents (suffixes) given topics (paradigms). A stem should be able to occur with each suffix of a canonical paradigm. But if a stem occurs with one allomorphic variant of a suffix, we know that it necessarily cannot occur with the other. Therefore, allomorphy violates conditional independence of suffixes given a paradigm, and we cope with this by merging allomorphs. Second, LDA also assumes conditional independence of words (stems) given topics (paradigm). As our data contains stem variants, this assumption does not hold either, but it is a less serious violation due to the large number of total stems. Third, we have imposed the constraint of orthogonality of suffixes and paradigms, which is not required by LDA (and actually undesired in document topic modeling, since documents can contain multiple topics). Orthogonal suffix splits are possible when categorically ambiguous suffixes have been disambiguated.

In conclusion, we view morphology learning as a process of manipulating the representation of data to fit a learnable computational model. The alternative would be to complicate the model and learning algorithm to accommodate raw data and all its concurrent ambiguities and dependencies. We hypothesize that successful, fully unsupervised learning of linguistically adequate representations of morphology will be more easily accomplished by first bootstrapping the sorts of information that we have assumed, or, in other words, fitting the data to the model.

## Acknowledgements

This work was supported by the National Science Foundation under grant NSF IIS-0415138. The author thanks Mitch Marcus and anonymous reviewers for their helpful comments.

## References

A. Albright. 2002. The identification of bases in morphological paradigms. Ph.D. thesis, UCLA.

D. Blei, A. Ng, and M. Jordan. 2003. Latent dirichlet allocation. Journal of Machine Learning Research 3, 993-1022.

X. Carreras, I. Chao, L. Padró, and M. Padró. 2004. FreeLing: an open-source suite of language analyzers. Proceedings of LREC. Lisbon, Portugal.

A. Clark. 2001. Learning morphology with pair hidden markov models. Proceedings of the Student Workshop at ACL.

R. Evans and G. Gazdar. 1996. DATR: A language for lexical knowledge representation. Computational Linguistics 22(2), 167-216.

M. Forsberg and A. Ranta. 2004. Functional morphology. Proceedings of the ICFP, 213-223. ACM Press.

D. Freitag. 2005. Morphology induction from term clusters. Proceedings of CoNLL.

J. Goldsmith. 2001. Unsupervised learning of the morphology of a natural language. Computational Linguistics 27(2), 153-198.

S. Goldwater and M. Johnson. 2004. Priors in bayesian learning of phonological rules. Proceedings of SIGPHON.

D. Graff and G. Gallegos. 1999. Spanish newswire text, volume 2. Linguistic Data Consortium, Philadelphia, PA.

Y. Hu, I. Matveeva, J. Goldsmith, and C. Sprague. 2005. Using morphology and syntax together in unsupervised learning. Workshop on Psychocomputational Models of Human Language Acquisition.

D. Kazakov and S. Manandhar. 2001. Unsupervised learning of word segmentation rules with genetic algorithms and inductive logic programming. Machine Learning 43, 121-162.

M. Marcus, B. Santorini and M.A. Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. Computational Linguistics 19(2), 313-330.

C. Monson, A. Lavie, J. Carbonell, and L. Levin. 2004. Unsupervised induction of natural language morphology inflection classes. Proc. of SIGPHON.

K. Oflazer, S. Nirenburg, and M. McShane. 2001. Bootstrapping morphological analyzers by combining human elicitation and machine learning. Computational Linguistics 27(1), 59-85.

P. Schone and D. Jurafsky. 2001. Knowledge-free induction of inflectional morphologies. Proc. NAACL.

M. Snover, G. Jarosz, and M. Brent. 2002. Unsupervised learning of morphology using a novel directed search algorithm: taking the first step. Proceedings of SIGPHON.

D. Yarowsky and R. Wicentowski. 2000. Minimally supervised morphological analysis by multimodal alignment. Proceedings of ACL.

R. Zajac. 2001. Morpholog: constrained and supervised learning of morphology. Proceedings of CoNLL.