Improving morphology induction by learning spelling rules

Citation for published version:

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Peer reviewed version

Published In:
Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI-09)

General rights
Copyright for the publications made accessible via the Edinburgh Research Explorer is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy
The University of Edinburgh has made every reasonable effort to ensure that Edinburgh Research Explorer content complies with UK legislation. If you believe that the public display of this file breaches copyright please contact openaccess@ed.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.
Improving Morphology Induction by Learning Spelling Rules

Jason Naradowsky  
Department of Computer Science  
University of Massachusetts Amherst  
Amherst, MA 01003, USA  
narad@cs.umass.edu

Sharon Goldwater  
School of Informatics  
University of Edinburgh  
Edinburgh, EH8 9AB, UK  
sgwater@inf.ed.ac.uk

Abstract

Unsupervised learning of morphology is an important task for human learners and in natural language processing systems. Previous systems focus on segmenting words into substrings (taking ⇒ taking), but sometimes a segmentation-only analysis is insufficient (e.g., taking may be more appropriately analyzed as take+ing, with a spelling rule accounting for the deletion of the stem-final e). In this paper, we develop a Bayesian model for simultaneously inducing both morphology and spelling rules. We show that the addition of spelling rules improves performance over the baseline morphology-only model.

1 Introduction

In natural language, words are often constructed from multiple morphemes, or meaning-bearing units, such as stems and suffixes. Identifying the morphemes within words is an important task both for human learners and in natural language processing (NLP) systems, where it can improve performance on a variety of tasks by reducing data sparsity [Goldwater and McClosky, 2005; Larkey et al., 2002]. Unsupervised learning of morphology is particularly interesting, both from a cognitive standpoint (because developing unsupervised systems may shed light on how humans perform this task) and for NLP (because morphological annotation is scarce or nonexistent in many languages). Existing systems, such as [Goldsmith, 2001] and [Creutz and Lagus, 2005], are relatively successful in segmenting words into constituent morphs (essentially, substrings), e.g. reporters ⇒ report.ers. However, strategies based purely on segmentation of observed forms make systematic errors in identifying morphological relationships because many of these relationships are obscured by spelling rules that alter the observed forms of words.1 For example, most English verbs take -ing as the present continuous tense ending (walking), but after stems ending in e, the e is deleted (taking), while for some verbs, the final stem consonant is doubled (shutting, digging). A purely segmenting system will be forced to segment shutting as either shut+ing or shut+ing. In the first case, shutting will be correctly identified as sharing a stem with words such as shut and shuts, but will not share a suffix with words such as walking and running. In the second case, the opposite will be true. In this paper, we present a Bayesian model of morphology that identifies the latent underlying morphological analysis of each word (shut+ing)2 along with spelling rules that generate the observed surface forms.

Most current systems for unsupervised morphological analysis in NLP are based on various heuristic methods and perform segmentation only [Monson et al., 2004; Freitag, 2005; Dasgupta and Ng, 2006; Dasgupta and Ng, 2007] also infers some spelling rules. Although these can be effective, our goal is to investigate methods which can eventually be built into larger joint inference systems for learning multiple aspects of language (such as morphology, phonology, and syntax) in order to examine the kinds of structures and biases that are needed for successful learning in such a system. For this reason, we focus on probabilistic models rather than heuristic procedures.

Previously, [Goldsmith, 2006] and [Goldwater and John-son, 2004] have described model-based morphology induction systems that can account for some variations in morphs caused by spelling rules. Both systems are based on the Minimum Description Length principle and share certain weaknesses that we address here. In particular, due to their complex MDL objective functions, these systems incorporate special-purpose algorithms to search for the optimal morphological analysis of the input corpus. This raises the possibility that the search procedures themselves are influencing the results of these systems, and makes it difficult to extend the underlying models or incorporate them into larger systems other than through a strict 1-best pipelined approach. Indeed, each of these systems extends the segmentation-only system of [Goldsmith, 2001] by first using that system to identify a segmentation, and then (in a second step), finding spelling rules to simplify the original analysis. In contrast, the model presented here uses standard sampling methods for inference, and provides a way to simultaneously learn both morphological analysis and spelling rules, allowing information from each component to flow to the other during learning. We

1Human learners encounter an analogous problem with phonological rules that alter the observed forms of spoken words.

2In what follows, we use ‘+’ to indicate an underlying morpheme boundary, and ‘.’ to indicate a surface segmentation.
show that the addition of spelling rules allows our model to outperform the earlier segmentation-only Bayesian model of [Goldwater et al., 2006], on which it is based.

In the remainder of this paper, we begin by reviewing the baseline model from [Goldwater et al., 2006]. We then describe our extensions to it and the sampler we use for inference. We present experiments demonstrating that the combined morphology-spelling model outperforms the baseline. Finally, we discuss remaining sources of error in the system and how we might address them in the future.

2 Baseline segmentation model

We take as our baseline the simple model of morphology described in [Goldwater et al., 2006], which generates a word w in three steps:

1. Choose a morphological class c for w.
2. Choose a stem t conditioned on c.
3. Choose a (possibly empty) suffix f conditioned on c.

Since t and f are assumed to be conditionally independent given c, we have

\[ P(c, t, f) = P(c)P(t | c)P(f | c) \]  

(1)

and

\[ P(w) = \sum_{c,t,f:w=wf} P(c, t, f) \]

where the sum is over all stem-suffix combinations that can be concatenated to form w. This model is of course simplistic in its assumption that words may only consist of two morphs; however, for the test set of English verbs that was used by [Goldwater et al., 2006], two morphs is sufficient. A similar model that allows multiple morphs per word is described in [Goldsmith, 2001].

Goldwater et al. present the model above within a Bayesian framework in which the goal is to identify a high-probability sequence of classes, stems, and suffixes (c, t, f) given an observed sequence of words w. This is done using Bayes’ rule:

\[ P(c, t, f | w) \propto P(w | c, t, f)P(c, t, f) \]  

(2)

Note that the likelihood \( P(w | c, t, f) \) can take on only two possible values: 1 if the observed words are consistent with t and f, and 0 otherwise. Therefore, the prior distribution over analyses \( P(c, t, f) \) is crucial to inference. As in other model-based unsupervised morphology learning systems [Goldsmith, 2001; Creutz and Lagus, 2005], Goldwater et al. assume that sparse solutions – analyses containing fewer total stems and suffixes – should be preferred. This is done by placing symmetric Dirichlet priors over the multinomial distributions from which \( c, t, \) and \( f \) are drawn:

\[ \theta_c | \kappa \sim \text{Dir}(\kappa) \]  

\[ \theta_t | c \sim \text{Mult}(\theta_c) \]  

\[ \theta_f | c \sim \text{Mult}(\theta_f | c) \]  

(3)

where \( \theta_c, \theta_t | c, \) and \( \theta_f | c \) are the multinomial parameters for classes, stems, and suffixes, and \( \kappa, \tau, \) and \( \phi \) are the respective Dirichlet hyperparameters. We discuss below the significance of the hyperparameters and how they can be used to favor sparse solutions. Under this model, the probability of a particular analysis can be computed as

\[ P(c, t, f) = \prod_{i=1}^{N} P(c_i | c_{-i}) \cdot P(t_i | t_{-i}, c_{-i}, c_i) \cdot P(f_i | f_{-i}, c_{-i}, c_i) \]  

(4)

where \( N \) is the total number of words and the notation \( x_{-i} \) indicates \( x_1 \ldots x_{i-1} \). The probability of each factor is computed by integrating over the parameters associated with that factor. For example,

\[ P(c_i = c | c_{-i}, \kappa) = \int P(c_i = c | \theta_c)P(\theta_c | c_{-i}, \kappa) \, d\theta_c \]

\[ = \frac{n_c^{(i-1)} + \kappa}{n_{c-1}^{(i)} + C\kappa} \]  

(5)

where \( n_c^{(i-1)} \) is the number of occurrences of \( c \) in \( c_{-i} \), \( n_{c-1}^{(i)} \) is the length of \( c_{-i} = i - 1 \), and \( C \) is the total number of possible classes. The value of the integration is a standard result in Bayesian statistics [Gelman et al., 2004], and can be used (as Goldwater et al. do) to develop a Gibbs sampler for inference. We defer discussion of inference to Section 4.

While the model described above is effective in segmenting words into their stems and inflectional suffixes, such a segmentation misses certain linguistic generalizations, as described in the introduction and illustrated in Figure 1. In order to identify these generalizations, it is necessary to go beyond simple segmentation of the words in the input. In the following section, we describe an extension to the above generative model in which spelling rules apply after the stem and suffix are concatenated together, so that the stem and suffix of each word may not correspond exactly to a segmentation of the observed form.

3 Accounting for spelling rules

To extend the baseline model, we introduce the notion of a spelling rule, inspired by the phonological rules of Chomsky and Halle [1968]. Each rule is characterized by a transformation and a context in which the transformation applies. We develop two models, one based on a two-character context formed with one left context character and one right context character, and the other based on a three-character context.
with an additional left context character. We assume that transformations only occur at morpheme boundaries, so the context consists of the final one or two characters of the (underlying) stem and the first character of the suffix. For example, shut+ing, take+ing, sleep+s have contexts ut,i, ke,i, ep,s. Transformations can include insertions, deletions, or empty rules, and always apply to the position immediately preceding the morpheme boundary, i.e. deletions delete the stem-final character and insertions insert a character following the stem-final character.3 So the rule ε → t/ut,i produces shutting, ε → e/ke,i produces taking, and ε → e/ep,s produces sleeps. Our new model extends the baseline generative process with two additional steps:

4. Choose the rule type y (insertion, deletion, empty) conditioned on x(f, t), the context defined by t and f.
5. Choose a transformation r conditioned on y and x(f, t), which gives us the following joint probability:

\[ P(c, t, f, y, r) = P(c)P(t|c)P(f|c)P(y|x(f, t))P(r|y, x(f, t)) \] (6)

As above, we place Dirichlet priors over the multinomial distributions from which y and r are chosen. Our expectations are that most rules should be empty (i.e., observed forms are usually the same as underlying forms), so we use a non-symmetric Dirichlet prior over rule types, with \( \eta = (\eta_D, \eta_I, \eta_E) \) being the hyperparameters over insertion, deletion, and empty rules, where \( \eta_E \) is set to a much larger value than \( \eta_D \) and \( \eta_I \) (we discuss this in more detail below). In addition, at most one or two different transformations should occur in any given context. We encourage this by using a small value for \( \rho \), the hyperparameter of the symmetric Dirichlet prior over transformations.

4 Inference

We sample from the posterior distribution of our model \( P(c, t, f, y, r | w) \) using Gibbs sampling, a standard Markov chain Monte Carlo (MCMC) technique [Gilks et al., 1996]. Gibbs sampling involves repeatedly sampling the value of each variable in the model conditioned on the current values of all other variables. This process defines a Markov chain whose stationary distribution is the posterior distribution over model variables given the input data. Because the variables that define the analysis of a given word are highly dependent (only certain choices of t, f, y, and r are consistent), we use blocked sampling to sample all variables for a single word at once. That is, we consider each word \( w_i \) in the data in turn, consider all possible values of \( (c, t, f, y, r) \) comprising a consistent analysis \( A(w_i) \) of \( w_i \), and compute the probability of each full analysis conditioned on the current analyses of all other words. We then sample an analysis for the current word according to this distribution and move on to the next word. After a suitable burn-in period, the sampler converges to sampling from the posterior distribution.

3Permitting arbitrary substitution rules allows too much freedom to the model and yields poor results; in future work we hope to achieve better results by using priors to constrain substitutions in a linguistically plausible way.

Computing the conditional probability of \( A(w_i) \) is straightforward because the Dirichlet-multinomial distributions we have constructed our model from are exchangeable: the probability of a set of outcomes does not depend on their ordering. We can therefore treat each analysis as though it is the last one in the data set, and apply the same integration over parameters that led to Equation 5. The full sampling equations for \( A(w_i) \) are shown in Figure 2.

Our model contains a number of hyperparameters. Rather than setting these by hand, we optimize them by maximizing the posterior probability of each hyperparameter given all other variables in the model. For example, to maximize \( \tau \) we have

\[ \tau^* = \arg\max_\tau P(\tau | \kappa, \tau, \phi, \rho, \eta, \eta_E, \eta_I, \eta_D, c, t, f, y, r) \]

\[ = \arg\max_\tau \prod_e \Gamma((\eta_{t,e} + \tau)^+) \frac{\Gamma(\tau)}{\prod_i \Gamma(\tau^+)} \] (9)

which can be optimized iteratively using the following fixed point algorithm [Minka, 2003]:

\[ (\tau)^\text{new} := (\tau) \sum_e \frac{1}{\tau^+} \Psi(\eta_{t,e} + \tau) - \frac{1}{\tau^+} \Psi(\tau) \] (10)

5 Experiments

In this section, we describe the experiments used to test our morphological induction system. We begin by discussing our input data sets, then present two distinct evaluation methods, and finally describe the results of our experiments.

5.1 Data

For input data we use the same data set used by [Goldwater et al., 2006], the set of 7487 English verbs found in the Penn Wall Street Journal (WSJ) corpus [Marcus et al., 1993]. English verbs provide a good starting point for evaluating our system because they contain many regular patterns, but also a number of orthographic transformations. We do not include frequency information in our input corpus; this is standard in morphology induction and has both psychological [Pierrehumbert, 2003] and mathematical justifications [Goldwater et al., 2006].

5.2 Evaluation

Although evaluation measures based solely on a gold standard surface segmentation are sometimes used, it should be clear from our introduction that this kind of measure is not sufficient for our purposes. Instead, we use two different evaluation measures based on the underlying morphological structure of the data. Both of our evaluation methods use the English portion of the CELEX database [Baayen et al., 1995] to determine correctness. It contains morphological analyses of 160,594 different inflected wordforms based on 52,446 uninflected lemmata. Each morphological analysis includes both a surface segmentation as well as an abstract morphosyntactic analysis which provides the functional role
\[ P(A(w_i) = (c, t, f, y, r) \mid A(w_{-i})), \kappa, \tau, \phi, \eta, \rho) \]
\[ \propto I(w_i = r(t, f)) \cdot P(c, t, f, y, r \mid A(w_{-i})), \kappa, \tau, \phi, \eta, \rho) \]
\[ \propto P(c \mid x_{-i}), \kappa \cdot P(t \mid t_{-i}, c, \tau) \cdot P(f \mid f_{-i}, c, \phi) \cdot P(y \mid y_{-i}, t, f, \eta) \cdot P(r \mid r_{-i}, t, f, y, \rho) \]
\[ = \frac{n_c^{(i)} + \kappa}{(n^{(i)} + \kappa + \tau)} \cdot \frac{n_{t,c}^{(i)} + \tau}{(n_c^{(i)} + \tau + \tau T)} \cdot \frac{n_{f,c}^{(i)} + \phi}{(n_c^{(i)} + \phi F)} \cdot \frac{n_{g,x(t,f)}^{(i)} + \eta_y}{(n^{(i)} + \eta_y F)} \cdot \frac{n_{r,g,x(t,f)}^{(i)} + \rho}{(n^{(i)} + \rho R)} \]  
(7)

Figure 2: Equations used in sampling to compute the probability of of the analysis \( A(w_i) \) of \( w_i \), conditioned on \( A(w_{-i}) \), the analyses of all other words in the data set. We use the notation \( x_{-i} \) here to indicate \( x_1 \ldots x_{i-1}, x_{i+1} \ldots x_N \). \( I(.) \) is a function taking on the value 1 when its argument is true, and 0 otherwise. \( \kappa, \tau, \phi, \eta, \rho \) are the hyperparameters for the Dirichlet distributions associated with classes, stems, suffixes, rule types, and rules, respectively; and \( C, T, F, R \) specify the total number of possible values for classes, stems, suffixes, and rules. Note that for \( y = delete \) or \( empty \), there is only one possible rule, so \( R = 1 \) and the final factor cancels out. For \( y = insert \), \( R = 26 \).

Table 1: An example illustrating the resources used for evaluation and our two scoring methods. We suppose that \textbf{Found} is the analysis found by the system. \textbf{CX string} is the segmentation of the surface form given in CELEX. \textbf{CX abstract} is the abstract morpheme analysis given in CELEX (with each stem represented by a unique ID, and each suffix represented by a code such as \texttt{pe} for present participle), used to compute pairwise precision (PP) and pairwise recall (PR). \textbf{UF string} is the underlying string representation we derived based on the two CELEX representations (see text), used to compute UF accuracy (UFA). UF strings that do not match those found by the system are shown in bold. In this example, scores for stems are 10/13 (UFA), 8/10 (PP), and 8/15 (PR). Scores for suffixes are 11/13 (UFA), 9/12 (PP), and 9/16 (PR).

<table>
<thead>
<tr>
<th>Found</th>
<th>CX string</th>
<th>CX abstract</th>
<th>UF string</th>
</tr>
</thead>
<tbody>
<tr>
<td>walk+ε</td>
<td>walk.ε</td>
<td>50655+i</td>
<td>walk+ε</td>
</tr>
<tr>
<td>walk+ing</td>
<td>walk.ing</td>
<td>50655+pe</td>
<td>walk+ing</td>
</tr>
<tr>
<td>walk+ed</td>
<td>walk.ed</td>
<td>50655+a1S</td>
<td>walk+ed</td>
</tr>
<tr>
<td>forget+ε</td>
<td>forget.ε</td>
<td>17577+i</td>
<td>forget+ε</td>
</tr>
<tr>
<td>forget+ing</td>
<td>forget.ing</td>
<td>17577+pe</td>
<td>forget+ing</td>
</tr>
<tr>
<td>forget+a</td>
<td>forget.a</td>
<td>17577+e3S</td>
<td>forget+a</td>
</tr>
<tr>
<td>state+ε</td>
<td>state.ε</td>
<td>44380+i</td>
<td>state+ε</td>
</tr>
<tr>
<td>state+ing</td>
<td>state.ing</td>
<td>44380+pe</td>
<td>state+ing</td>
</tr>
<tr>
<td>state+ed</td>
<td>state.ed</td>
<td>44380+a1S</td>
<td>state+ed</td>
</tr>
<tr>
<td>state+a</td>
<td>state.a</td>
<td>44380+s3S</td>
<td>state+a</td>
</tr>
<tr>
<td>state+ion</td>
<td>state.ion</td>
<td>44405+i</td>
<td>state+ion</td>
</tr>
<tr>
<td>jump+ed</td>
<td>jump.ed</td>
<td>24596+a1S</td>
<td>jump+ed</td>
</tr>
</tbody>
</table>

Table 1: An example illustrating the resources used for evaluation and our two scoring methods. We suppose that \textbf{Found} is the analysis found by the system. \textbf{CX string} is the segmentation of the surface form given in CELEX. \textbf{CX abstract} is the abstract morpheme analysis given in CELEX (with each stem represented by a unique ID, and each suffix represented by a code such as \texttt{pe} for present participle), used to compute pairwise precision (PP) and pairwise recall (PR). \textbf{UF string} is the underlying string representation we derived based on the two CELEX representations (see text), used to compute UF accuracy (UFA). UF strings that do not match those found by the system are shown in bold. In this example, scores for stems are 10/13 (UFA), 8/10 (PP), and 8/15 (PR). Scores for suffixes are 11/13 (UFA), 9/12 (PP), and 9/16 (PR).

of any inflectional suffixes. For example, the word \textit{walking} is segmented as \textit{walk.ing}, and is accompanied by a \texttt{pe} label to denote the suffix’s role in marking it as a present tense (ε) participle (p). See Table 1 for further examples.

Our first evaluation method is based on the pairwise relational measure used in the recent PASCAL challenge on unsupervised morpheme analysis.4 Consider the proposed analysis \textit{walk+ing} and its corresponding gold standard entry 50655+\texttt{pe}. Assuming that this analysis is correct, any other correct analysis that shares the stem \textit{walk} should also share the same stem ID 50655, and likewise for the suffixes.

By comparing the pairwise relationships in the system output and the gold standard, we can compute pairwise precision (PP) as the proportion of proposed pairs that are correct, and pairwise recall (PR) as the proportion of true pairs that are correctly identified. This is reported separately for stems and suffixes along with the F-Measures of each, calculated as \( F = \frac{2 \cdot PP \cdot PR}{PP + PR} \).

Our second evaluation method is designed to more directly test the correctness of underlying forms by using the analyses provided in CELEX to reconstruct an underlying form (UF) for each surface form. To identify the underlying stem for a word, we use the lemma ID number, which is the same for all inflected forms and specifies the canonical dictionary form, which is identical to the stem. To identify the underlying suffix, we map each of the suffix functional labels to a canonical string representation. Specifically, \texttt{pe} \Rightarrow \texttt{ing}, \texttt{a1S} \Rightarrow \texttt{ed}, \texttt{e3S} \Rightarrow \texttt{s}, and all other labels are mapped to the empty string ε. When the CELEX surface segmentation of an inflected form has an empty suffix, indicating an irregular form such as \textit{forget.ε}, we use the surface segmentation as the UF. We can then compute underlying form accuracy (UFA) for stems as the proportion of found stems that match those in the UF's, and likewise for suffixes.

5.3 Inference and hyperparameters

Our inference procedure alternates between sampling the variables in the model and updating the hyperparameters. For both the baseline and spelling-rule system, we ran the algorithm for 5 epochs, with each epoch containing 10 iterations of sampling and 10 iterations of hyperparameter updates. Although it is possible to automatically learn values for all of the hyperparameters in the model, we chose to set the values of the hyperparameters over rule types by hand to reflect our intuitions that empty rules should be far more prevalent than in insertions or deletions. That is, the hyperparameter for empty rules \( \eta_E \) should be relatively high, while the hyperparameters determining insertion and deletion rules, \( \eta_I \) and \( \eta_D \), should be low (and, for simplicity, we assume they are equal). Results reported here use \( \eta_E = 5, \eta_I = \eta_D = 0.001 \) (although other similar values yield similar results). All other hyperparameters were learned.

The remaining model parameters are either determined by

4http://www.cis.hut.fi/morphochallenge2007/
the data or set by hand. For the WSJ verbs data set the number of possible stems, \( T = 7,306,988 \), and the number of possible suffixes, \( F = 5,555 \), are calculated by enumerating all possible segmentation of the words in the data set and accounting for every possible rule transformation. We set the number of classes \( C = 1 \) and the minimum stem length to three characters. Enforcing a minimum stem length ensures that even in the case of the most minimal stem and the application of an insertion rule, the underlying stem will still have two characters to form the left context.

### 5.4 Results

Quantitative results for the two systems are shown in Table 2, with examples of full analyses shown in Figure 3 and the most commonly inferred spelling rules in Figure 4. Overall, the augmented models dramatically outperform the baseline on the UFA stem metric, which is not surprising considering that it is the introduction of rules that allows these models to correctly capture stems that may have been improperly segmented in the baseline (Figure 1).

However, the baseline performs better on suffix UFA by a fair margin. There are at least two contributing factors causing this. First, the addition of spelling rules allows the model to explain some suffixes in alternate undesirable ways. For instance, the \(-ed\) suffix is often analyzed as a \(-d\) suffix with an \( e \)-insertion rule, or, as in the case of \( symbolize \), analyzed as a \(-d\) suffix with an \( s \)-deletion rule. The latter case is somewhat attributable to data sparsity, where the base form, \( symbolize \), is not found in the data. In these circumstances it can be preferable to analyze these as \( symbolizes \) with an empty rule, and \( symbolizes+d \) with the erroneous \( s \)-deletion rule (Figure 4), so that they share the same stem. These analyses would not be likely using a larger data set.

Second, the presence of derivational suffixes in the data is a contributing factor because they are not analyzed correctly in the inflectional verbs section of CELEX, which forms our gold standard. Consider that the baseline provides the most succinct analysis of suffixes, positing just four (\( e, -s, -ed, \) and \(-ing\)), whereas the three-character-context model induces five (the same four with the addition of \(-d\) ). The two-character-context model, the worst-performing system on suffix UFA, learns an additional five suffixes (\( e, -es, -n, -ize, \) and \(-ized\)). Not all of these additional forms are unreasonable; \(-ize \) and \(-n \) are both valid suffixes, and \(-ized \) is the remainder of a correct segmentation. However, because suffixes like \(-ize \) are derivational (they change the part-of-speech of the root they attach to), they are not considered as part of the canonical dictionary of our gold standard. In this situation the UFA metric therefore provides an upper-bound for the baseline, but a lower-bound for augmented systems.

The pair-wise metrics are also susceptible to this problem, but continue to support the conclusions reached previously on overall system performance. The baseline slightly outperforms the three-character-context model in stem PP, but compares quite poorly in stem PR, and in stem PF. It again performs better than the augmented models on suffix tasks. Worth noting is that the errors made according to this metric are a small set of very pervasive mistakes. For instance, improperly segmenting \(-ed\) suffixes as \(-d\) suffixes or segmenting a stem-final \( e \) as its own suffix together contribute to more than half of all erroneous suffixes proposed by this model.

In addition to improved performance on the morphology induction task, our system also produces a probabilistic representation of the phonology of a language in the spelling rules it learns. The most frequently learned rules (Table 4) are largely correct, with just two spurious rules induced. While many of these are linguistically redundant because of the overspecification of their contexts, most refer to valid, desirable orthographic rules. Examples of these are \( e \)-deletion in various contexts (\( state+ing \Rightarrow state+ing \)), \( e \)-insertions (\( pass+s \Rightarrow passes \)), and consonant doubling when taking the \(-ing\) suffix (\( forget+ing \Rightarrow forgetting \), \( spam+ing \Rightarrow spamming \)).

### 6 Conclusion

As we noted in the introduction, one of the difficulties of unsupervised morphology induction is that spelling rules often act to obscure the morphological analyses of the observed words. A few previous model-based systems have tried to deal with this, but only by first segmenting the corpus into morphs, and then trying to identify spelling rules to simplify the analysis. To our knowledge, this is the first work to present a probabilistic model using a joint inference procedure to simultaneously induce both morphological analyses and spelling rules. Our results are promising: our model is able to identify morphological analyses that produce more accurate stems than the baseline while also inducing a number of spelling rules that correctly characterize the transformations in our data.
Table 2: Performance of the baseline model and two augmented models, measured using pairwise precision (PP), pairwise recall (PR), pairwise F-measure (PF), and underlying form accuracy (UFA).

<table>
<thead>
<tr>
<th>Model</th>
<th>Stems PP</th>
<th>Stems PR</th>
<th>Stems PF</th>
<th>Stems UFA</th>
<th>Suffixes PP</th>
<th>Suffixes PR</th>
<th>Suffixes PF</th>
<th>Suffixes UFA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>.610</td>
<td>.647</td>
<td>.628</td>
<td>.580</td>
<td>.461</td>
<td>.722</td>
<td>.563</td>
<td>.921</td>
</tr>
<tr>
<td>Two-Character-Context</td>
<td>.473</td>
<td>.667</td>
<td>.445</td>
<td>.656</td>
<td>.423</td>
<td>.472</td>
<td>.446</td>
<td>.753</td>
</tr>
<tr>
<td>Three-Character-Context</td>
<td>.584</td>
<td>.911</td>
<td>.712</td>
<td>.786</td>
<td>.415</td>
<td>.578</td>
<td>.483</td>
<td>.856</td>
</tr>
</tbody>
</table>

Of course, our model is still somewhat preliminary in several respects. For example, a single stem and suffix is insufficient to capture the morphological complexity of many languages (including English), and substitution rules should ideally be allowed along with deletions and insertions. Extending the model to allow for these possibilities would create many more potential analyses, making it more difficult to identify appropriate solutions. However, there are also many sensible constraints that could be placed on the system that we have yet to explore. In particular, aside from assuming that empty rules are more likely than others, we placed no particular expectations on the kinds of rules that should occur. However, assuming some rough knowledge of the pronunciation of different letters (or a phonological transcription), it would be possible to use our priors to encode the kinds of transformations that are more likely to occur (e.g., vowels to vowels, consonants to phonologically similar consonants). We hope to pursue this line of work in future research.

Acknowledgments
The authors would like to thank Hanna Wallach for useful discussions regarding hyperparameter inference.

References