Universal Grapheme-to-Phoneme Prediction Over Latin Alphabets

Young-Bum Kim and Benjamin Snyder
University of Wisconsin-Madison
{ybkim,bsnyder}@cs.wisc.edu

Abstract

We consider the problem of inducing grapheme-to-phoneme mappings for unknown languages written in a Latin alphabet. First, we collect a data-set of 107 languages with known grapheme-phoneme relationships, along with a short text in each language. We then cast our task in the framework of supervised learning, where each known language serves as a training example, and predictions are made on unknown languages. We induce an undirected graphical model that learns phonotactic regularities, thus relating textual patterns to plausible phonemic interpretations across the entire range of languages. Our model correctly predicts grapheme-phoneme pairs with over 88% F1-measure.

1 Introduction

Written language is one of the defining technologies of human civilization, and has been independently invented at least three times through the course of history (Daniels and Bright, 1996). In many ways written language reflects its more primary spoken counterpart. Both are subject to some of the same forces of change, including human migration, cultural influence, and imposition by empire. In other ways, written language harkens further to the past, reflecting aspects of languages long since gone from their spoken forms. In this paper, we argue that this imperfect relationship between written symbol and spoken sound can be automatically inferred from textual patterns. By examining data for over 100 languages, we train a statistical model to automatically relate graphemic patterns in text to phonemic sequences for never-before-seen languages.

We focus here on the the alphabet, a writing system that has come down to us from the Sumerians. In an idealized alphabetic system, each phoneme in the language is unambiguously represented by a single grapheme. In practice of course, this ideal is never achieved. When existing alphabets are melded onto new languages, they must be imperfectly adapted to a new sound system. In this paper, we exploit the fact that a single alphabet, that of the Romans, has been adapted to a very large variety of languages.

Recent research has demonstrated the effectiveness of cross-lingual analysis. The joint analysis of several languages can increase model accuracy, and enable the development of computational tools for languages with minimal linguistic resources. Previous work has focused on settings where just a handful of languages are available. We treat the task of grapheme-to-phoneme analysis as a test case for larger scale multilingual learning, harnessing information from dozens of languages.

On a more practical note, accurately relating graphemes and phonemes to one another is crucial for tasks such as automatic speech recognition and text-to-speech generation. While pronunciation dictionaries and transcribed audio are available for some languages, these resources are entirely lacking for the vast majority of the world's languages. Thus, automatic and generic methods for determining sound-symbol relationships are needed.

Our paper is based on the following line of reasoning: that character-level textual patterns mirror

Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pages 332–343, Jeju Island, Korea, 12–14 July 2012.

phonotactic regularities; that phonotactic regularities are shared across related languages and universally constrained; and that textual patterns for a newly observed language may thus reveal its underlying phonemics. Our task can be viewed as an easy case of lost language decipherment – one where the underlying alphabetic system is widely known.

Nevertheless, the task of grapheme-to-phoneme prediction is challenging. Characters in the Roman alphabet can take a wide range of phonemic values across the world's languages. For example, depending on the language, the grapheme "c" can represent the following phonemes:[1]

- /k/ (unvoiced velar plosive)
- /c/ (unvoiced palatal plosive)
- /s/ (unvoiced alveolar fricative)
- /ǀ/ (dental click)
- /d͡ʒ/ (affricated voiced postalveolar fricative)
- /t͡ʃ/ (affricated unvoiced postalveolar fricative)
- /t͡s/ (affricated unvoiced alveolar fricative)

To make matters worse, the same language may use a single grapheme to ambiguously represent *multiple* phonemes. For example, English orthography uses "c" to represent both /k/ and /s/. Our task is thus to select a *subset* of phonemes for each language's graphemes. We cast the subset selection problem as a set of related binary prediction problems, one for each possible grapheme-phoneme pair. Taken together, these predictions yield the grapheme-phoneme mapping for that language.

We develop a probabilistic undirected graphical model for this prediction problem, where a large set of languages serve as training data and a single held-out language serves as test data. Each training and test language yields an instance of the graph, bound together through a shared set of features and parameter values to allow cross-lingual learning and generalization.

In the graph corresponding to a given language, each node represents a grapheme-phoneme pair $(g : p)$. The node is labeled with a binary value to indicate whether grapheme g can represent phoneme p in the language. In order to allow coupled labelings across the various grapheme-phoneme pairs of the language, we employ a connected graph structure, with an automatically learned topology shared across the languages. The node and edge features are derived from textual co-occurrence statistics for the graphemes of each language, as well as general information about the language's family and region. Parameters are jointly optimized over the training languages to maximize the likelihood of the node labelings given the observed feature values. See Figure 1 for a snippet of the model.

We apply our model to a novel data-set consisting of grapheme-phoneme mappings for 107 languages with Roman alphabets and short texts. In this setting, we consider each language in turn as the test language, and train our model on the remaining 106 languages. Our highest performing model achieves an F1-measure of 88%, yielding perfect predictions for over 21% of languages. These results compare quite favorably to several baselines.

Our experiments lead to several conclusions. (i) Character co-occurence features alone are not sufficient for cross-lingual predictive accuracy in this task. Instead, we map raw contextual counts to more linguistically meaningful generalizations to learn effective cross-lingual patterns. (ii) A connected graph topology is crucial for learning linguistically coherent grapheme-to-phoneme mappings. Without any edges, our model yields perfect mappings for only 10% of test languages. By employing structure learning and including the induced edges, we more than double the number of test languages with perfect predictions. (iii) Finally, an analysis of our grapheme-phoneme predictions shows that they do not achieve certain global characteristics observed across true phoneme inventories. In particular, the level of "feature economy" in our predictions is too low, suggesting an avenue for future research.

[1]For some brief background on phonetics, see Section 2. Note that we use the term "phoneme" throughout the paper, though we also refer to "phonetic" properties. As we are dealing with texts (written in a roughly phonemic writing system), we have no access to the true contextual phonetic realizations, and even using IPA symbols to relate symbols across languages is somewhat theoretically suspect.

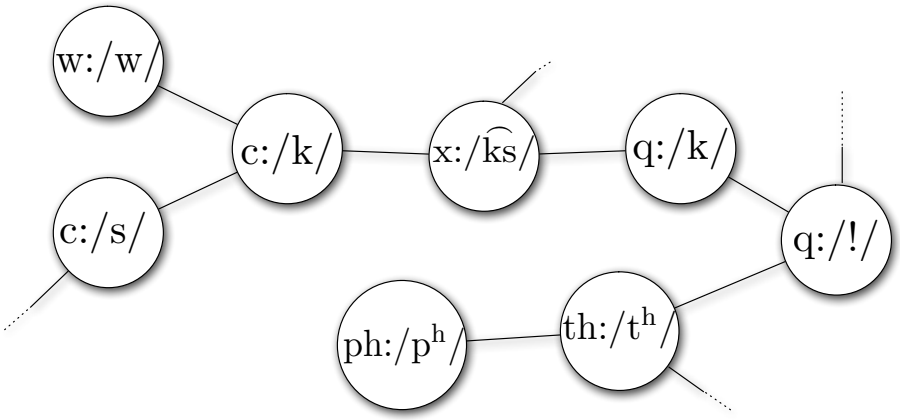

Figure 1: **A snippet of our undirected graphical model.** The binary-valued nodes represent whether a particular grapheme-phoneme pair is allowed by the language. Sparse edges are automatically induced to allow joint training and prediction over related inventory decisions.

2 Background and Related Work

In this section, we provide some background on phonetics and phoneme inventories. We also review prior work on grapheme-to-phoneme prediction and multilingual modeling.

2.1 Phoneme Inventories

The sounds of the world's languages are produced through a wide variety of articulatory mechanisms. Consonants are sounds produced through a partial or complete stricture of the vocal tract, and can be roughly categorized along three independent dimensions: *(i) Voicing:* whether or not oscillation of the vocal folds accompanies the sound. For example, /t/ and /d/ differ only in that the latter is voiced. *(ii) Place of Articulation:* where in the anatomy of the vocal tract the stricture is made. For example, /p/ is a bilabial (the lips touching one another) while /k/ is a velar (tongue touching touching the soft palate). *(iii) Manner of Articulation*: the manner in which the airflow is regulated. For example, /m/ is a nasal (air flowing through the nostrils), while /p/ is a plosive (obstructed air suddenly released through the mouth).

In contrast, vowels are voiced sounds produced with an open vocal tract. They are categorized primarily based on the position of the tongue and lips, along three dimensions: *(i) Roundedness:* whether or not the lips are rounded during production of the sound; *(ii) Height:* the vertical position of the tongue; *(iii) Backness:* how far forward the tongue lies.

Linguists have noted several statistical regularities found in phoneme inventories throughout the world. *Feature economy* refers to the idea that languages tend to minimize the number of differentiating characteristics (e.g. different kinds of voicing, manner, and place) that are used to distinguish consonant phonemes from one another (Clements, 2003). In other words, once an articulatory feature is used to mark off one phoneme from another, it will likely be used again to differentiate other phoneme pairs in the same language. The principle of *Maximal perceptual contrast* refers to the idea that the set of vowels employed by a language will be located in phonetic space to maximize their perceptual distances from one another, thus relieving the perceptual burden of the listener (Liljencrants and Lindblom, 1972). In an analysis of our results, we will observe that our model's predictions do not always follow these principles.

Finally, researchers have noted that languages exhibit set patterns in how they sequence their phonemes (Kenstowicz and Kisseberth, 1979). Certain sequences are forbidden outright by languages, while others are avoided or favored. While many of these patterns are language-specific, others seem more general, either reflecting anatomical con-

straints, common language ancestry, or universal aspects of the human language system. These phonotactic regularities and constraints are mirrored in graphemic patterns, and as our experiments show, can be explicitly modeled to achieve high accuracy in our task.

2.2 Grapheme-to-Phoneme Prediction

Much prior work has gone into developing methods for accurate grapheme-to-phoneme prediction. The common assumption underlying this research has been that some sort of knowledge, usually in the form of a pronunciation dictionary or phonemically annotated text, is available for the language at hand. The focus has been on developing techniques for dealing with the phonemic ambiguity present both in annotated and unseen words. For example, Jiampojamarn and Kondrak (Jiampojamarn and Kondrak, 2010) develop a method for aligning pairs of written and phonemically transcribed strings; Dwyer and Kondrak (Dwyer and Kondrak, 2009) develop a method for accurate letter-to-phoneme conversion while minimizing the number of training examples; Reddy and Goldsmith (Reddy and Goldsmith, 2010) develop an MDL-based approach to finding subword units that align well to phonemes.

A related line of work has grown around the task of machine transliteration. In this task, the goal is to automatically transliterate a name in one language into the written form of another language. Often this involves some level of phonetic analysis in one or both languages. Notable recent work in this vein includes research by Sproat et al (Sproat et al., 2006) on transliteration between Chinese and English using comparable corpora, and Ravi and Knight (Ravi and Knight, 2009) who take a decipherment approach to this problem.

Our work differs from all previous work on grapheme-to-phoneme prediction in that (i) we assume no knowledge for our target language beyond a small unannotated text (and possibly some region or language family information), and (ii) our goal is to construct the inventory of mappings between the language's letters and its phonemes (the latter of which we do not know ahead of time). When a grapheme maps to more than one phoneme, we do not attempt to disambiguate particular instances of that grapheme in words.

A final thread of related work is the task of quantitatively categorizing writing systems according to their levels of phonography and logography (Sproat, 2000; Penn and Choma, 2006). As our data-set consists entirely of Latin-based writing systems, our work can be viewed as a more fine-grained computational exploration of the space of writing systems, with a focus on phonographic systems with the Latin pedigree.

2.3 Multilingual Analysis

An influential thread of previous multilingual work starts with the observation that rich linguistic resources exist for some languages but not others. The idea then is to *project* linguistic information from one language onto others via parallel data. Yarowsky and his collaborators first developed this idea and applied it to the problems of part-of-speech tagging, noun-phrase bracketing, and morphology induction (Yarowsky and Wicentowski, 2000; Yarowsky et al., 2000; Yarowsky and Ngai, 2001), and other researchers have applied the idea to syntactic and semantic analysis (Hwa et al., 2005; Padó and Lapata, 2006) In these cases, the existence of a bilingual parallel text along with highly accurate predictions for one of the languages was assumed.

Another line of work assumes the existence of bilingual parallel texts without the use of any supervision (Dagan et al., 1991; Resnik and Yarowsky, 1997). This idea has been developed and applied to a wide variety tasks, including morphological analysis (Snyder and Barzilay, 2008a; Snyder and Barzilay, 2008b), part-of-speech induction (Snyder et al., 2008; Snyder et al., 2009a; Naseem et al., 2009), and grammar induction (Snyder et al., 2009b; Blunsom et al., 2009; Burkett et al., 2010). An even more recent line of work does away with the assumption of parallel texts and performs joint unsupervised induction for various languages through the use of coupled priors in the context of grammar induction (Cohen and Smith, 2009; Berg-Kirkpatrick and Klein, 2010).

In contrast to these previous approaches, the method we propose does not assume the existence of any parallel text, but instead assumes that labeled data exists for a wide variety of languages. In this regard, our work most closely resembles recent work which trains a universal morphological analyzer us-

	phonemes	#lang	ent
a	/a/ /ɐ/ /ɑ/ /ə/ /ʌ/	106	1.25
c	/c/ /d͡ʒ/ /k/ /s/ /t͡s/ /t͡ʃ/ /ǀ/	62	2.33
ch	/k/ /t͡ʃ/ /x/ /ʃ/	39	1.35
e	/e/ /i/ /æ/ /ə/ /ɛ/	106	1.82
h	/-/ /h/ /x/ /ø/ /ɦ/	85	1.24
i	/i/ /j/ /ɪ/	106	0.92
j	/d͡ʒ/ /h/ /j/ /t͡ʃ/ /x/ /ɟ/ /ʒ/	79	2.05
o	/o/ /u/ /ɒ/ /ʁ/	103	1.47
ph	/f/ /p^{h}/	15	0.64
q	/k/ /q/ /!/	32	1.04
r	/r/ /ɽ/ /ɾ/ /ʀ/ /ʁ/	95	1.50
th	/t^{h}/ /θ/	15	0.64
u	/u/ /w/ /y/ /ɨ/ /ʊ/ /ʏ/	104	0.96
v	/b/ /f/ /v/ /w/ /β/	70	1.18
w	/u/ /v/ /w/	74	0.89
x	/k͡s/ /x/ /‖/ /ʃ/	44	1.31
z	/d͡z/ /s/ /t͡s/ /z/ /θ/	72	0.93

Table 1: Ambiguous graphemes and the set of phonemes that they may represent among our set of 107 languages.

ing a structured nearest neighbor approach for 8 languages (Kim et al., 2011). Our work extends this idea to a new task and also considers a much larger set of languages. As our results will indicate, we found that a nearest neighbor approach was not as effective as our proposed model-based approach.

3 Data and Features

In this section we discuss the data and features used in our experiments.

3.1 Data

The data for our experiments comes from three sources: (i) grapheme-phoneme mappings from an online encyclopedia, (ii) translations of the Universal Declaration of Human Rights (UDHR)[2], and *(iii)* entries from the World Atlas of Language Structures (WALS) (Haspelmath and Bibiko, 2005).

To start, we downloaded and transcribed image files containing grapheme-phoneme mappings for several hundred languages from an online encyclopedia of writing systems[3]. We then cross-referenced the languages with the World Atlas of Language Structures (WALS) database (Haspelmath and Bibiko, 2005) as well as the translations available for the Universal Declaration of Human Rights (UDHR). Our final set of 107 languages includes those which appeared consistently in all three sources and that employ a Latin alphabet. See Figure 2 for a world map annotated with the locations listed in the WALS database for these languages, as well as their language families. As seen from the figure, these languages cover a wide array of language families and regions.

We then analyzed the phoneme inventories for the 107 languages. We decided to focus our attention on graphemes which are widely used across these languages with a diverse set of phonemic values. We measured the ambiguity of each grapheme by calculating the entropy of its phoneme sets across the languages, and found that 17 graphemes had entropy > 0.5 and appeared in at least 15 languages. Table 1 lists these graphemes, the set of phonemes that they can represent, the number of languages in our data-set which employ them, and the entropy of their phoneme-sets across these languages. The data, along with the feature vectors discussed below, are published as part of this paper.

3.2 Features

The key intuition underlying this work is that graphemic patterns in text can reveal the phonemes which they represent. A crucial step in operationalizing this intuition lies in defining input features that have cross-lingual predictive value. We divide our feature set into three categories.

Text Context Features: These features represent the textual environment of each grapheme in a language. For each grapheme g, we consider counts of graphemes to the immediate left and right of g in the UDHR text. We define five feature templates, including counts of (1) single graphemes to the left of g, (2) single graphemes to the right of g, (3) pairs of graphemes to the left of g, (4) pairs of graphemes to the right of g, and (5) pairs of graphemes surrounding g. As our experiments below show, this set of features on its own performs poorly. It seems that

[2]http://www.ohchr.org/en/udhr/pages/introduction.aspx

[3]http://www.omniglot.com/writing/langalph.htm#latin

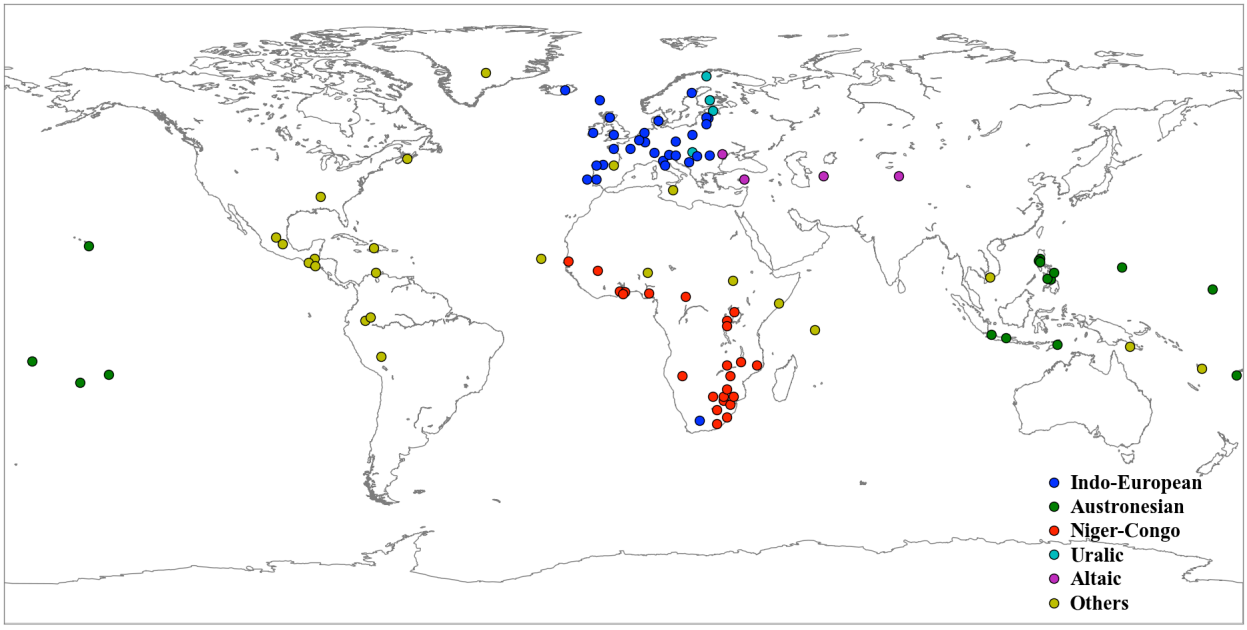

Figure 2: Map and language families of languages in our data-set

these features are too language specific and not abstract enough to yield effective cross-lingual generalization. Our next set of features was designed to alleviate this problem.

Phonemic Context Features: A perfect feature-set would depend on the entire set of grapheme-to-phoneme predictions for a language. In other words, we would ideally map *all* the graphemes in our text to phonemes, and then consider the plausibility of the resulting phoneme sequences. In practice, of course, this is impossible, as the set of possible grapheme-to-phoneme mappings is exponentially large. As an imperfect proxy for this idea, we made the following observation: for most Latin graphemes, the most common phonemic value across languages is the identical IPA symbol of that grapheme (e.g. the most common phoneme for *g* is /g/, the most common phoneme for *t* is /t/, etc). Using this observation, we again consider all contexts in which a grapheme appears, but this time map the surrounding graphemes to their IPA phoneme equivalents. We then consider various linguistic properties of these surrounding "phonemes" – whether they are vowels or consonants, whether they are voiced or not, their manner and places of articulation – and create phonetic context features. The process is illustrated in Figure 3. The intuition here is that these features can (noisily) capture the *phonotactic context* of a grapheme, allowing our model to learn general phonotactic constraints. As our experiments below demonstrate, these features proved to be quite powerful.

Language Family Features: Finally, we consider features drawn from the WALS database which capture general information about the language – specifically, its region (e.g. Europe), its small language family (e.g. Germanic), and its large language family (e.g. Indo-European). These features allow our model to capture family and region specific phonetic biases. For example, African languages are more likely to use *c* and *q* to represents clicks than are European languages. As we mention below, we also consider conjunctions of all features. Thus, a language family feature can combine with a phonetic context feature to represent a family specific phonotactic constraint. Interestingly, our experiments below show that these features are not needed for highly accurate prediction.

3.3 Feature Discretization and Filtering

It is well known that many learning techniques perform best when continuous features are binned and

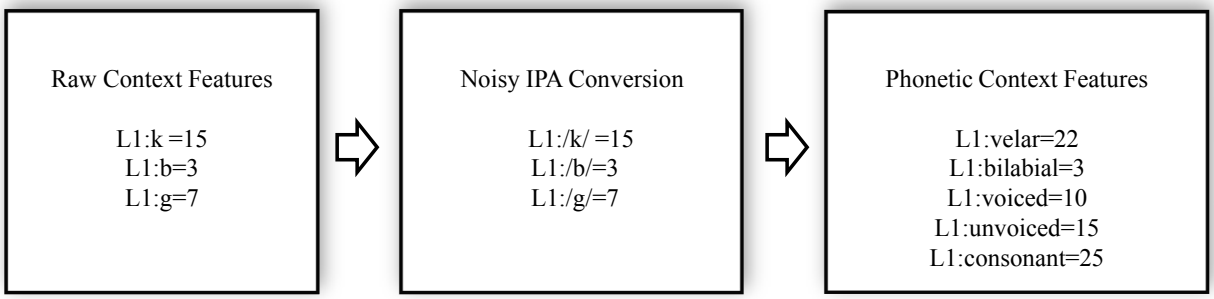

Figure 3: **Generating phonetic context features.** First, character context features are extracted for each grapheme. The features drawn here give the counts of the character to the immediate left of the grapheme. Next, the contextual characters are noisily converted to phones using their IPA notation. Finally, phonetic context features are extracted. In this case, phones /k/ and /g/ combine to give a "velar" count of 22, while /g/ and /b/ combine to give a "voiced" count of 10.

converted to binary values (Dougherty et al., 1995). As a preprocessing step, we therefore discretize and filter the count-based features outlined above. We adopt the technique of Recursive Minimal Entropy Partitioning (Fayyad and Irani, 1993). This technique recursively partitions feature values so as to minimize the conditional entropy of the labels. Partitioning stops when the gain in label entropy falls below the number of additional bits in overhead needed to describe the new feature split. This leads to a (local) minimum description length discretization.

We noticed that most of our raw features (especially the text features) could not achieve even a single split point without increasing description length, as they were not well correlated with the labels. We decided to use this heuristic as a feature selection technique, discarding such features. After this discretization and filtering, we took the resulting binary features and added their pairwise conjunctions to the set. This process was conducted separately for each leave-one-out scenario, without observation of the test language labels. Table 2 shows the total number of features before the discretization/filtering as well as the typical numbers of features obtained after filtering (the exact numbers depend on the training/test split).

	Raw	Filtered
# Text Features	28,474	1,848
# Phonemic Features	28,948	7,799
# Family Features	66	32
Total	57,488	9,679

Table 2: **Number of features** in each category before and after discretization/filtering. Note that the pair-wise conjunction features are not included in these counts.

4 Model

Using the features described above, we develop an undirected graphical model approach to our prediction task. Corresponding to each training language is an instance of our undirected graph, labeled with its true grapheme-phoneme mapping. We learn weights over our features which optimally relate the input features of the training languages to their observed labels. At test-time, the learned weights are used to predict the labeling of the held-out test language.

More formally, we assume a set of graph nodes $1, ..., m$ with edges between some pairs of nodes (i, j). Each node corresponds to a grapheme-phoneme pair $(g : p)$ and can be labeled with a binary value. For each training language ℓ, we observe a text $\mathbf{x}^{(\ell)}$ and a binary labeling of the graph nodes $\mathbf{y}^{(\ell)}$. For each node i, we also obtain a feature vector $f_i(\mathbf{x}^{(\ell)})$, by examining the language's text and extracting textual and noisy phonetic patterns (as detailed in the previous section). We obtain similar feature vectors for edges (i, j): $g_{jk}(\mathbf{x}^{(\ell)})$. We then parameterize the probability of each labeling using a log-linear form over node and edge factors:[4]

[4]The delta function $\delta(p)$ evaluates to 1 when predicate p is

$$\log P\big(\mathbf{y}^{(\ell)}|\mathbf{x}^{(\ell)}\big) = \sum_i \lambda_i \cdot \big[f_i(\mathbf{x}^{(\ell)})\ \delta(y_i^{(\ell)} = 1)\big]$$
$$+ \sum_{j,k} \lambda_{jk1} \cdot \big[g_{jk}(\mathbf{x}^{(\ell)})\ \delta(y_j^{(\ell)} = 1 \wedge y_k^{(\ell)} = 1)\big]$$
$$+ \sum_{j,k} \lambda_{jk2} \cdot \big[g_{jk}(\mathbf{x}^{(\ell)})\ \delta(y_j^{(\ell)} = 1 \wedge y_k^{(\ell)} = 0)\big]$$
$$+ \sum_{j,k} \lambda_{jk3} \cdot \big[g_{jk}(\mathbf{x}^{(\ell)})\ \delta(y_j^{(\ell)} = 0 \wedge y_k^{(\ell)} = 1)\big]$$
$$- \log Z(\mathbf{x}^{(\ell)}, \lambda)$$

The first term sums over nodes i in the graph. For each i, we extract a feature vector $f_i(\mathbf{x}^{(\ell)})$. If the label of node i is 1, we take the dot product of the feature vector and corresponding parameters, otherwise the term is zeroed out. Likewise for the graph edges j, k: we extract a feature vector, and depending on the labels of the two vertices y_j and y_k, take a dot product with the relevant parameters. The final term is a normalization constant to ensure that the probabilities sum to one over all possible labelings of the graph.

Before learning our parameters, we first automatically induce the set of edges in our graph, using the PC graph structure learning algorithm (Spirtes et al., 2000). This procedure starts with a fully connected undirected graph structure, and iteratively removes edges between nodes that are conditionally independent given other neighboring nodes in the graph according to a statistical independence test over all training languages. In our graphs we have 75 nodes, and thus 2,775 potential edges. Running the structure learning algorithm on our data yields sparse graphs, typically consisting of about 50 edges. In each leave-one-out scenario, a single structure is learned for all languages.

Once the graph structure has been induced, we learn parameter values by maximizing the L2-penalized conditional log-likelihood over all training languages:[5]

$$L(\lambda) = \sum_\ell \log P\big(\mathbf{y}^{(\ell)}|\mathbf{x}^{(\ell)}\big) - C||\lambda||^2$$

The gradient takes the standard form of a difference between expected and observed feature counts (Lafferty et al., 2001). Expected counts, as well as predicted assignments at test-time, are computed using loopy belief propagation (Murphy et al., 1999). Numerical optimization is performed using L-BFGS (Liu and Nocedal, 1989).

5 Experiments

In this section, we describe the set of experiments performed to evaluate the performance of our model. Besides our primary undirected graphical model, we also consider several baselines and variants, in order to assess the contribution of our model's graph structure as well as the features used. In all cases, we perform leave-one-out cross-validation over the 107 languages in our data-set.

5.1 Baselines

Our baselines include:

1. A majority baseline, where the most common binary value is chosen for each grapheme-phoneme pair,
2. two linear SVM's, one trained using the discretized and filtered features described in Section 3.2, and the other using the raw continuous features,
3. a Nearest Neighbor classifier, which chooses the closest training language for each grapheme-phoneme pair in the discretized feature space, and predicts its label, and
4. a variant of our model with no edges between nodes (essentially reducing to a set of independent log-linear classifiers).

5.2 Evaluation

We report our results using three evaluation metrics of increasing coarseness.

1. **Phoneme-level**: For individual grapheme-phoneme pairs (e.g. a:/ɐ/, a:/ʌ/, c:/k/, c:/tʃ/) our task consists of a set of binary predictions, and can thus be evaluated in terms of precision, recall, and F1-measure. We report micro-averages of these quantities across all

true, and to 0 when p is false.

[5] In our experiments, we used an L2 penalty weight of .5 for node features and .1 for edge features. Similar results are observed for a wide range of values.

	Phoneme			Grapheme	Language
	Precision	Recall	F1	Accuracy	Accuracy
MAJORITY	80.47	57.47	67.06	55.54	2.8
SVM CONTINUOUS	79.87	64.48	79.87	59.07	3.74
SVM DISCRETE	90.55	78.27	83.97	70.78	8.41
NEAREST NEIGHBOR	85.35	79.43	82.28	67.97	2.8
MODEL: NO EDGES	89.35	82.05	85.54	73.96	10.28
FULL MODEL	91.06	83.98	87.37	78.58	**21.5**
MODEL: NO FAMILY	**92.43**	**84.67**	**88.38**	**80.04**	19.63
MODEL: NO TEXT	89.58	81.43	85.31	75.86	15.89
MODEL: NO PHONETIC	86.52	74.19	79.88	69.6	9.35

Table 3: The performance of baselines and variants of our model, evaluated at the phoneme-level (binary predictions), whole-grapheme accuracy, and whole-language accuracy.

grapheme-phoneme pairs in all leave-one-out test languages.

2. **Grapheme-level**: We also report grapheme-level accuracy. For this metric, we consider each grapheme g and examine its predicted labels over all its possible phonemes: $(g : p_1), (g : p_2), ..., (g : p_k)$. If all k binary predictions are correct, then the grapheme's phoneme-set has been correctly predicted. We report the percentage of all graphemes with such correct predictions (micro-averaged over all graphemes in all test language scenarios).

3. **Language-level**: Finally, we assess language-wide performance. For this metric, we report the percentage of test languages for which our model achieves perfect predictions on all grapheme-phoneme pairs, yielding a perfect mapping.

5.3 Results

The results for the baselines and our model are shown in Table 3. The majority baseline yields 67% F1-measure on the phoneme-level binary prediction task, with 56% grapheme accuracy, and about 3% language accuracy.

Using undiscretized raw count features, the SVM improves phoneme-level performance to about 80% F1, but fails to provide any improvement on grapheme or language performance. In contrast, the SVM using discretized and filtered features achieves performance gains in all three categories, achieving 71% grapheme accuracy and 8% language accuracy. The nearest neighbor baseline achieves performance somewhere in between the two SVM variants.

The unconnected version of our model achieves similar, though slightly improved performance over the discretized SVM. Adding the automatically induced edges into our model leads to significant gains across all three categories. Phoneme-level F1 reaches 87%, grapheme accuracy hits 79%, and language accuracy more than doubles, achieving 22%. It is perhaps not surprising that the biggest relative gains are seen at the language level: by jointly learning and predicting an entire language's grapheme-phoneme inventory, our model ensures that language-level coherence is maintained.

Recall that three sets of features are used by our models. (1) language family and region features, (2) textual context features, and (3) phonetic context features. We now assess the relative merits of each set by considering our model's performance when the set has been removed. Table 3 shows several striking results from this experiment. First, it appears that dropping the region and language family features actually improves performance. This result is somewhat surprising, as we expected these features to be quite informative. However, it appears that whatever information they convey is redundant when considering the text-based feature sets. We next observe that dropping the textual context features leads to a small drop in performance. Finally, we see that dropping the phonetic context features seriously degrades our model's accuracy. Achieving robust cross-linguistic generalization apparently requires a level of feature abstraction not achieved by

character-level context features alone.

6 Global Inventory Analysis

In the previous section we saw that our model achieves relatively high performance in predicting grapheme-phoneme relationships for never-before-seen languages. In this section we analyze the predicted phoneme inventories and ask whether they display the statistical properties observed in the gold-standard mappings.

As outlined in Section 2, consonant phonemes can be represented by the three articulatory features of voicing, manner, and place. The principle of feature economy states that phoneme inventories will be organized to minimize the number of distinct articulatory features used in the language, while maximizing the number of resulting phonemes. This principle has several implications. First, we can measure the *economy index* of a consonant system by computing the ratio of the number of consonantal phonemes to the number of articulatory features used in their production: $\frac{\#\text{consonants}}{\#\text{features}}$ (Clements, 2003). The higher this value, the more economical the sound system.

Secondly, for each articulatory dimension we can calculate the empirical distribution over values observed across the consonants of the language. Since consonants are produced as combinations of the three articulatory dimensions, the greatest number of consonants (for a given set of utilized feature values) will be produced when the distributions are close to uniform. Thus, we can measure how economical each feature dimension is by computing the entropy of its distribution over consonants. For example, in an economical system, we would expect roughly half the consonants to be voiced, and half to be unvoiced.

Table 4 shows the results of this analysis. First, we notice that the average entropy of voiced vs. unvoiced consonants is nearly identical in both cases, close to the optimal value. However, when we examine the dimensions of place and manner, we notice that the entropy induced by our model is not as high as that of the true consonant inventories, implying a suboptimal allocation of consonants. In fact, when we examine the economy index (ratio of consonants to features), we indeed find that – on average – our model's predictions are not as economical as the gold standard. This analysis suggests that we might obtain a more powerful predictive model by taking the principle of feature economy into account.

	H(voice)	H(place)	H(manner)	Economy Index
True	0.9739	2.7355	2.4725	1.6536
Predicted	0.9733	2.6715	2.4163	1.6337

Table 4: Measures of feature economy applied to the predicted and true consonant inventories (averaged over all 107 languages).

7 Conclusions

In this paper, we considered a novel problem: that of automatically relating written symbols to spoken sounds for an unknown language using a known writing system – the Latin alphabet. We constructed a data-set consisting of grapheme-phoneme mappings and a short text for over 100 languages. This data allows us to cast our problem in the supervised learning framework, where each observed language serves as a training example, and predictions are made on a new language. Our model automatically learns how to relate textual patterns of the unknown language to plausible phonemic interpretations using induced phonotactic regularities.

Acknowledgments

This work is supported by the NSF under grant IIS-1116676. Any opinions, findings, or conclusions are those of the authors, and do not necessarily reflect the views of the NSF.

References

Taylor Berg-Kirkpatrick and Dan Klein. 2010. Phylogenetic grammar induction. In *Proceedings of the ACL*, pages 1288–1297, Uppsala, Sweden, July. Association for Computational Linguistics.

P. Blunsom, T. Cohn, C. Dyer, and M. Osborne. 2009. A gibbs sampler for phrasal synchronous grammar induction. In *Proceedings of the Joint Conference of the*

47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2, pages 782–790. Association for Computational Linguistics.

David Burkett, Slav Petrov, John Blitzer, and Dan Klein. 2010. Learning better monolingual models with unannotated bilingual text. In *Proceedings of CoNLL*.

G.N. Clements. 2003. Feature economy in sound systems. *Phonology*, 20(3):287–333.

Shay B. Cohen and Noah A. Smith. 2009. Shared logistic normal distributions for soft parameter tying in unsupervised grammar induction. In *Proceedings of the NAACL/HLT*.

Ido Dagan, Alon Itai, and Ulrike Schwall. 1991. Two languages are more informative than one. In *Proceedings of the ACL*, pages 130–137.

P.T. Daniels and W. Bright. 1996. *The world's writing systems*, volume 198. Oxford University Press New York, NY.

James Dougherty, Ron Kohavi, and Mehran Sahami. 1995. Supervised and unsupervised discretization of continuous features. In *ICML*, pages 194–202.

K. Dwyer and G. Kondrak. 2009. Reducing the annotation effort for letter-to-phoneme conversion. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1*, pages 127–135. Association for Computational Linguistics.

Usama M Fayyad and Keki B Irani. 1993. Multi-interval discretization of continuous-valued attributes for classification learning. In *Proceedings of the International Joint Conference on Uncertainty in AI*, pages 1022–1027.

M. Haspelmath and H.J. Bibiko. 2005. *The world atlas of language structures*, volume 1. Oxford University Press, USA.

R. Hwa, P. Resnik, A. Weinberg, C. Cabezas, and O. Kolak. 2005. Bootstrapping parsers via syntactic projection across parallel texts. *Journal of Natural Language Engineering*, 11(3):311–325.

S. Jiampojamarn and G. Kondrak. 2010. Letter-phoneme alignment: An exploration. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 780–788. Association for Computational Linguistics.

M.J. Kenstowicz and C.W. Kisseberth. 1979. *Generative phonology*. Academic Press San Diego, CA.

Young-Bum Kim, João Graça, and Benjamin Snyder. 2011. Universal morphological analysis using structured nearest neighbor prediction. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 322–332, Edinburgh, Scotland, UK., July. Association for Computational Linguistics.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 282–289.

J. Liljencrants and B. Lindblom. 1972. Numerical simulation of vowel quality systems: the role of perceptual contrast. *Language*, pages 839–862.

D.C. Liu and J. Nocedal. 1989. On the limited memory bfgs method for large scale optimization. *Mathematical programming*, 45(1):503–528.

K.P. Murphy, Y. Weiss, and M.I. Jordan. 1999. Loopy belief propagation for approximate inference: An empirical study. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pages 467–475. Morgan Kaufmann Publishers Inc.

Tahira Naseem, Benjamin Snyder, Jacob Eisenstein, and Regina Barzilay. 2009. Multilingual part-of-speech tagging: two unsupervised approaches. *Journal of Artificial Intelligence Research*, 36(1):341–385.

Sebastian Padó and Mirella Lapata. 2006. Optimal constituent alignment with edge covers for semantic projection. In *Proceedings of ACL*, pages 1161 – 1168.

G. Penn and T. Choma. 2006. Quantitative methods for classifying writing systems. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, pages 117–120. Association for Computational Linguistics.

S. Ravi and K. Knight. 2009. Learning phoneme mappings for transliteration without parallel data. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 37–45. Association for Computational Linguistics.

S. Reddy and J. Goldsmith. 2010. An mdl-based approach to extracting subword units for grapheme-to-phoneme conversion. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 713–716. Association for Computational Linguistics.

Philip Resnik and David Yarowsky. 1997. A perspective on word sense disambiguation methods and their evaluation. In *Proceedings of the ACL SIGLEX Workshop on Tagging Text with Lexical Semantics: Why, What, and How?*, pages 79–86.

B. Snyder and R. Barzilay. 2008a. Unsupervised multilingual learning for morphological segmentation. *Proceedings of ACL-08: HLT*, pages 737–745.

Benjamin Snyder and Regina Barzilay. 2008b. Cross-lingual propagation for morphological analysis. In *Proceedings of the AAAI*, pages 848–854.

Benjamin Snyder, Tahira Naseem, Jacob Eisenstein, and Regina Barzilay. 2008. Unsupervised multilingual learning for POS tagging. In *Proceedings of EMNLP*, pages 1041–1050.

B. Snyder, T. Naseem, J. Eisenstein, and R. Barzilay. 2009a. Adding more languages improves unsupervised multilingual part-of-speech tagging: A bayesian non-parametric approach. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 83–91. Association for Computational Linguistics.

Benjamin Snyder, Tahira Naseem, and Regina Barzilay. 2009b. Unsupervised multilingual grammar induction. In *Proceedings of the ACL*, pages 73–81.

P. Spirtes, C.N. Glymour, and R. Scheines. 2000. *Causation, prediction, and search*, volume 81. The MIT Press.

R. Sproat, T. Tao, and C.X. Zhai. 2006. Named entity transliteration with comparable corpora. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 73–80. Association for Computational Linguistics.

R.W. Sproat. 2000. *A computational theory of writing systems*. Cambridge Univ Pr.

David Yarowsky and Grace Ngai. 2001. Inducing multilingual pos taggers and np bracketers via robust projection across aligned corpora. In *Proceedings of the NAACL*, pages 1–8.

David Yarowsky and Richard Wicentowski. 2000. Minimally supervised morphological analysis by multimodal alignment. In *ACL '00: Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, pages 207–216, Morristown, NJ, USA. Association for Computational Linguistics.

David Yarowsky, Grace Ngai, and Richard Wicentowski. 2000. Inducing multilingual text analysis tools via robust projection across aligned corpora. In *Proceedings of HLT*, pages 161–168.