A Statistical Model for Lost Language Decipherment

Benjamin Snyder and **Regina Barzilay**
CSAIL
Massachusetts Institute of Technology
{bsnyder,regina}@csail.mit.edu

Kevin Knight
ISI
University of Southern California
knight@isi.edu

Abstract

In this paper we propose a method for the automatic decipherment of lost languages. Given a non-parallel corpus in a known related language, our model produces both alphabetic mappings and translations of words into their corresponding cognates. We employ a non-parametric Bayesian framework to simultaneously capture both low-level character mappings and high-level morphemic correspondences. This formulation enables us to encode some of the linguistic intuitions that have guided human decipherers. When applied to the ancient Semitic language Ugaritic, the model correctly maps 29 of 30 letters to their Hebrew counterparts, and deduces the correct Hebrew cognate for 60% of the Ugaritic words which have cognates in Hebrew.

1 Introduction

Dozens of lost languages have been deciphered by humans in the last two centuries. In each case, the decipherment has been considered a major intellectual breakthrough, often the culmination of decades of scholarly efforts. Computers have played no role in the decipherment any of these languages. In fact, skeptics argue that computers do not possess the "logic and intuition" required to unravel the mysteries of ancient scripts.[1] In this paper, we demonstrate that at least some of this logic and intuition *can* be successfully modeled, allowing computational tools to be used in the decipherment process.

[1] *"Successful archaeological decipherment has turned out to require a synthesis of logic and intuition ... that computers do not (and presumably cannot) possess."* A. Robinson, "Lost Languages: The Enigma of the World's Undeciphered Scripts" (2002)

Our definition of the computational decipherment task closely follows the setup typically faced by human decipherers (Robinson, 2002). Our input consists of texts in a lost language and a corpus of non-parallel data in a known related language. The decipherment itself involves two related subtasks: (i) finding the mapping between alphabets of the known and lost languages, and (ii) translating words in the lost language into corresponding cognates of the known language.

While there is no single formula that human decipherers have employed, manual efforts have focused on several guiding principles. A common starting point is to compare letter and word frequencies between the lost and known languages. In the presence of cognates the correct mapping between the languages will reveal similarities in frequency, both at the character and lexical level. In addition, morphological analysis plays a crucial role here, as highly frequent morpheme correspondences can be particularly revealing. In fact, these three strands of analysis (character frequency, morphology, and lexical frequency) are intertwined throughout the human decipherment process. Partial knowledge of each drives discovery in the others.

We capture these intuitions in a generative Bayesian model. This model assumes that each word in the lost language is composed of morphemes which were generated with latent counterparts in the known language. We model bilingual morpheme pairs as arising through a series of Dirichlet processes. This allows us to assign probabilities based both on character-level correspondences (using a character-edit base distribution) as well as higher-level morpheme correspondences. In addition, our model carries out an implicit morphological analysis of the lost language, utilizing the known morphological structure of the related language. This model structure allows us to capture the interplay between the character-

and morpheme-level correspondences that humans have used in the manual decipherment process.

In addition, we introduce a novel technique for imposing structural sparsity constraints on character-level mappings. We assume that an accurate alphabetic mapping between related languages will be sparse in the following way: each letter will map to a very limited subset of letters in the other language. We capture this intuition by adapting the so-called "spike and slab" prior to the Dirichlet-multinomial setting. For each pair of characters in the two languages, we posit an indicator variable which controls the prior likelihood of character substitutions. We define a joint prior over these indicator variables which encourages sparse settings.

We applied our model to a corpus of Ugaritic, an ancient Semitic language discovered in 1928. Ugaritic was manually deciphered in 1932, using knowledge of Hebrew, a related language. We compare our method against the only existing decipherment baseline, an HMM-based character substitution cipher (Knight and Yamada, 1999; Knight et al., 2006). The baseline correctly maps the majority of letters — 22 out of 30 — to their correct Hebrew counterparts, but only correctly translates 29% of all cognates. In comparison, our method yields correct mappings for 29 of 30 letters, and correctly translates 60.4% of all cognates.

2 Related Work

Our work on decipherment has connections to three lines of work in statistical NLP. First, our work relates to research on cognate identification (Lowe and Mazaudon, 1994; Guy, 1994; Kondrak, 2001; Bouchard et al., 2007; Kondrak, 2009). These methods typically rely on information that is unknown in a typical deciphering scenario (while being readily available for living languages). For instance, some methods employ a hand-coded similarity function (Kondrak, 2001), while others assume knowledge of the phonetic mapping or require parallel cognate pairs to learn a similarity function (Bouchard et al., 2007).

A second related line of work is lexicon induction from non-parallel corpora. While this research has similar goals, it typically builds on information or resources unavailable for ancient texts, such as comparable corpora, a seed lexicon, and cognate information (Fung and McKeown, 1997; Rapp, 1999; Koehn and Knight, 2002; Haghighi et al., 2008). Moreover, distributional methods that rely on co-occurrence analysis operate over large corpora, which are typically unavailable for a lost language.

Finally, Knight and Yamada (1999) and Knight et al. (2006) describe a computational HMM-based method for deciphering an unknown script that represents a known spoken language. This method "makes the text speak" by gleaning character-to-sound mappings from non-parallel character and sound sequences. It does not relate words in different languages, thus it cannot encode deciphering constraints similar to the ones considered in this paper. More importantly, this method had not been applied to archaeological data. While lost languages are gaining increasing interest in the NLP community (Knight and Sproat, 2009), there have been no successful attempts of their automatic decipherment.

3 Background on Ugaritic

Manual Decipherment of Ugaritic Ugaritic tablets were first found in Syria in 1929 (Smith, 1955; Watson and Wyatt, 1999). At the time, the cuneiform writing on the tablets was of an unknown type. Charles Virolleaud, who lead the initial decipherment effort, recognized that the script was likely alphabetic, since the inscribed words consisted of only thirty distinct symbols. The location of the tablets discovery further suggested that Ugaritic was likely to have been a Semitic language from the Western branch, with properties similar to Hebrew and Aramaic. This realization was crucial for deciphering the Ugaritic script. In fact, German cryptographer and Semitic scholar Hans Bauer decoded the first two Ugaritic letters—*mem* and *lambda*—by mapping them to Hebrew letters with similar occurrence patterns in prefixes and suffixes. Bootstrapping from this finding, Bauer found words in the tablets that were likely to serve as cognates to Hebrew words—e.g., the Ugaritic word for *king* matches its Hebrew equivalent. Through this process a few more letters were decoded, but the Ugaritic texts were still unreadable. What made the final decipherment possible was a sheer stroke of luck—Bauer guessed that a word inscribed on an ax discovered in the Ras Shamra excavations was the Ugaritic word for *ax*. Bauer's guess was correct, though he selected the wrong phonetic sequence. Edouard Dhorme, another cryptographer

and Semitic scholar, later corrected the reading, expanding a set of translated words. Discoveries of additional tablets allowed Bauer, Dhorme and Virolleaud to revise their hypothesis, successfully completing the decipherment.

Linguistic Features of Ugaritic Ugaritic shares many features with other ancient Semitic languages, following the same word order, gender, number, and case structure (Hetzron, 1997). It is a morphologically rich language, with triliteral roots and many prefixes and suffixes.

At the same time, it exhibits a number of features that distinguish it from Hebrew. Ugaritic has a bigger phonemic inventory than Hebrew, yielding a bigger alphabet – 30 letters vs. 22 in Hebrew. Another distinguishing feature of Ugaritic is that vowels are only written with glottal stops while in Hebrew many long vowels are written using homorganic consonants. Ugaritic also does not have articles, while Hebrew nouns and adjectives take definite articles which are realized as prefixes. These differences result in significant divergence between Hebrew and Ugaritic cognates, thereby complicating the decipherment process.

4 Problem Formulation

We are given a corpus in a lost language and a non-parallel corpus in a related language from the same language family. Our primary goal is to translate words in the unknown language by mapping them to cognates in the known language. As part of this process, we induce a lower-level mapping between the letters of the two alphabets, capturing the regular phonetic correspondences found in cognates.

We make several assumptions about the writing system of the lost language. First, we assume that the writing system is alphabetic in nature. In general, this assumption can be easily validated by counting the number of symbols found in the written record. Next, we assume that the corpus has been transcribed into electronic format, where the graphemes present in the physical text have been unambiguously identified. Finally, we assume that words are explicitly separated in the text, either by white space or a special symbol.

We also make a mild assumption about the morphology of the lost language. We posit that each word consists of a stem, prefix, and suffix, where the latter two may be omitted. This assumption captures a wide range of human languages and a variety of morphological systems. While the correct morphological analysis of words in the lost language must be learned, we assume that the inventory and frequencies of prefixes and suffixes in the known language are given.

In summary, the observed input to the model consists of two elements: (i) a list of unanalyzed word types derived from a corpus in the lost language, and (ii) a morphologically analyzed lexicon in a known related language derived from a separate corpus, in our case non-parallel.

5 Model

5.1 Intuitions

Our goal is to incorporate the logic and intuition used by human decipherers in an unsupervised statistical model. To make these intuitions concrete, consider the following toy example, consisting of a lost language much like English, but written using numerals:

- 15234 *(asked)*
- 1525 *(asks)*
- 4352 *(desk)*

Analyzing the undeciphered corpus, we might first notice a pair of endings, -34, and -5, which both occur after the initial sequence 152- (and may likewise occur at the end of a variety of words in the corpus). If we know this lost language to be closely related to English, we can surmise that these two endings correspond to the English verbal suffixes *-ed* and *-s*. Using this knowledge, we can hypothesize the following character correspondences: (3 = *e*), (4 = *d*), (5 = *s*). We now know that (4252 = *des*2) and we can use our knowledge of the English lexicon to hypothesize that this word is *desk*, thereby learning the correspondence (2 = *k*). Finally, we can use similar reasoning to reveal that the initial character sequence 152- corresponds to the English verb *ask*.

As this example illustrates, human decipherment efforts proceed by discovering both character-level and morpheme-level correspondences. This interplay implicitly relies on a morphological analysis of words in the lost language, while utilizing knowledge of the known language's lexicon and morphology.

One final intuition our model should capture is the sparsity of the alphabetic correspondence between related languages. We know from comparative linguistics that the correct mapping will pre-

serve regular phonetic relationships between the two languages (as exemplified by cognates). As a result, each character in one language will map to a small number of characters in the other language (typically one, but sometimes two or three). By incorporating this structural sparsity intuition, we can allow the model to focus on on a smaller set of linguistically valid hypotheses.

Below we give an overview of our model, which is designed to capture these linguistic intuitions.

5.2 Model Structure

Our model posits that every observed word in the lost language is composed of a sequence of morphemes (prefix, stem, suffix). Furthermore we posit that each morpheme was probabilistically generated jointly with a latent counterpart in the known language.

Our goal is to find those counterparts that lead to high frequency correspondences both at the character and morpheme level. The technical challenge is that each level of correspondence (character and morpheme) can completely describe the observed data. A probabilistic mechanism based simply on one leaves no room for the other to play a role. We resolve this tension by employing a non-parametric Bayesian model: the distributions over bilingual morpheme pairs assign probability based on recurrent patterns at the morpheme level. These distributions are themselves drawn from a prior probabilistic process which favors distributions with consistent character-level correspondences.

We now give a formal description of the model (see Figure 1 for a graphical overview). There are four basic layers in the generative process:

1. **Structural sparsity**: draw a set of indicator variables $\vec{\lambda}$ corresponding to character-edit operations.

2. **Character-edit distribution**: draw a *base distribution* G_0 parameterized by weights on character-edit operations.

3. **Morpheme-pair distributions**: draw a set of distributions on bilingual morpheme pairs $G_{stm}, G_{pre|stm}, G_{suf|stm}$.

4. **Word generation**: draw pairs of cognates in the lost and known language, as well as words in the lost language with no cognate counterpart.

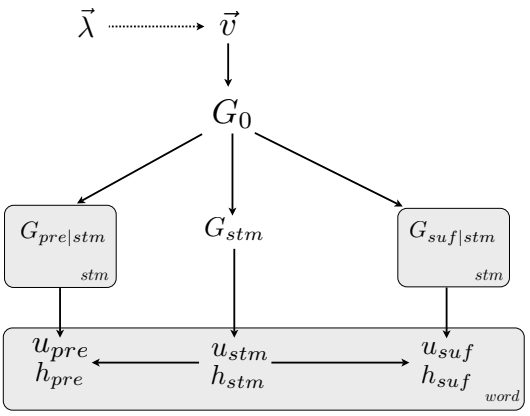

Figure 1: Plate diagram of the decipherment model. The structural sparsity indicator variables $\vec{\lambda}$ determine the values of the base distribution hyperparameters $\vec{v}$. The base distribution G_0 defines probabilities over string-pairs based solely on character-level edits. The morpheme-pair distributions $G_{stm}, G_{pre|stm}, G_{suf|stm}$ directly assign probabilities to highly frequent morpheme pairs.

We now go through each step in more detail.

Structural Sparsity The first step of the generative process provides a control on the sparsity of edit-operation probabilities, encoding the linguistic intuition that the correct character-level mappings should be sparse. The set of edit operations includes character substitutions, insertions, and deletions, as well as a special end symbol: $\{(u,h),(\epsilon,h),(u,\epsilon),END\}$ (where u and h range over characters in the lost and known languages, respectively). For each edit operation e we posit a corresponding indicator variable λ_e. The set of character substitutions with indicators set to one, $\{(u,h) : \lambda_{(u,h)} = 1\}$) conveys the set of phonetically valid correspondences. We define a joint prior over these variables to encourage sparse character mappings. This prior can be viewed as a distribution over *binary matrices* and is defined to encourage rows and columns to sum to low integer values (typically 1). More precisely, for each character u in the lost language, we count the number of mappings $c(u) = \sum_h \lambda_{(u,h)}$. We then define a set of features which count how many of these characters map to i other characters beyond some budget b_i: $f_i = \max(0, |\{u : c(u) = i\}| - b_i)$. Likewise, we define corresponding features f'_i and budgets b'_i for the characters h in the known lan-

guage. The prior over $\vec{\lambda}$ is then defined as

$$P(\vec{\lambda}) = \frac{\exp(\vec{f} \cdot \vec{w} + \vec{f}' \cdot \vec{w})}{Z} \quad (1)$$

where the feature weight vector $\vec{w}$ is set to encourage sparse mappings, and Z is a corresponding normalizing constant, which we never need compute. We set $\vec{w}$ so that each character must map to at least one other character, and so that mappings to more than one other character are discouraged [2]

Character-edit Distribution The next step in the generative process is drawing a base distribution G_0 over character edit sequences (each of which yields a bilingual pair of morphemes). This distribution is parameterized by a set of weights $\vec{\phi}$ on edit operations, where the weights over substitutions, insertions, and deletions each individually sum to one. In addition, G_0 provides a fixed distribution q over the *number* of insertions and deletions occurring in any single edit sequence. Probabilities over edit sequences (and consequently on bilingual morpheme pairs) are then defined according to G_0 as:

$$P(\vec{e}) = \prod_i \phi_{e_i} \cdot q\left(\#_{ins}(\vec{e}), \#_{del}(\vec{e})\right)$$

We observe that the average Ugaritic word is over two letters longer than the average Hebrew word. Thus, occurrences of Hebrew character insertions are *a priori* likely, and Ugaritic character deletions are very unlikely. In our experiments, we set q to disallow Ugaritic deletions, and to allow one Hebrew insertion per morpheme (with probability 0.4).

The prior on the base distribution G_0 is a Dirichlet distribution with hyperparameters $\vec{v}$, i.e., $\vec{\phi} \sim Dirichlet(\vec{v})$. Each value v_e thus corresponds to a character edit operation e. Crucially, the value of each v_e depends deterministically on its corresponding indicator variable:

$$v_e = \begin{cases} 1 & \text{if } \lambda_e = 0, \\ K & \text{if } \lambda_e = 1. \end{cases}$$

where K is some constant value > 1.[3] The overall effect is that when $\lambda_e = 0$, the marginal prior density of the corresponding edit weight ϕ_e spikes at 0. When $\lambda_e = 1$, the corresponding marginal prior density remains relatively flat and unconstrained. See (Ishwaran and Rao, 2005) for a similar application of "spike-and-slab" priors in the regression scenario.

Morpheme-pair Distributions Next we draw a series of distributions which *directly* assign probability to morpheme pairs. The previously drawn base distribution G_0 along with a fixed concentration parameter α define a *Dirichlet process* (Antoniak, 1974): $DP(G_0, \alpha)$, which provides probabilities over morpheme-pair distributions. The resulting distributions are likely to be skewed in favor of a few frequently occurring morpheme-pairs, while remaining sensitive to the character-level probabilities of the base distribution.

Our model distinguishes between three types of morphemes: prefixes, stems, and suffixes. As a result, we model each morpheme type as arising from distinct Dirichlet processes, that share a single base distribution:

$$\begin{aligned} G_{stm} &\sim DP(G_0, \alpha_{stm}) \\ G_{pre|stm} &\sim DP(G_0, \alpha_{pre}) \\ G_{suf|stm} &\sim DP(G_0, \alpha_{suf}) \end{aligned}$$

We model prefix and suffix distributions as conditionally dependent on the part-of-speech of the stem morpheme-pair. This choice capture the linguistic fact that different parts-of-speech bear distinct affix frequencies. Thus, while we draw a single distribution G_{stm}, we maintain separate distributions $G_{pre|stm}$ and $G_{suf|stm}$ for each possible stem part-of-speech.

Word Generation Once the morpheme-pair distributions have been drawn, actual word pairs may now be generated. First the model draws a boolean variable c_i to determine whether word i in the lost language has a cognate in the known language, according to some prior $P(c_i)$. If $c_i = 1$, then a cognate word pair (u, h) is produced:

$$\begin{aligned} (u_{stm}, h_{stm}) &\sim G_{stm} \\ (u_{pre}, h_{pre}) &\sim G_{pre|stm} \\ (u_{suf}, h_{suf}) &\sim G_{suf|stm} \\ u &= u_{pre} u_{stm} u_{suf} \\ h &= h_{pre} h_{stm} h_{suf} \end{aligned}$$

Otherwise, a lone word u is generated, according a uniform character-level language model.

[2] We set $w_0 = -\infty, w_1 = 0, w_2 = -50, w_{>2} = -\infty$, with budgets $b'_2 = 7, b'_3 = 1$ (otherwise zero), reflecting the knowledge that there are eight more Ugaritic than Hebrew letters.

[3] Set to 50 in our experiments.

In summary, this model structure captures both character and lexical level correspondences, while utilizing morphological knowledge of the known language. An additional feature of this multi-layered model structure is that each distribution over morpheme pairs is derived from the single character-level base distribution G_0. As a result, any character-level mappings learned from one type of morphological correspondence will be propagated to all other morpheme distributions. Finally, the character-level mappings discovered by the model are encouraged to obey linguistically motivated structural sparsity constraints.

6 Inference

For each word u_i in our undeciphered language we predict a morphological segmentation $(u_{pre}u_{stm}u_{suf})_i$ and corresponding cognate in the known language $(h_{pre}h_{stm}h_{suf})_i$. Ideally we would like to predict the analysis with highest marginal probability under our model given the observed undeciphered corpus and related language lexicon. In order to do so, we need to integrate out all the other latent variables in our model. As these integrals are intractable to compute exactly, we resort to the standard Monte Carlo approximation. We collect samples of the variables over which we wish to marginalize but for which we cannot compute closed-form integrals. We then approximate the marginal probabilities for undeciphered word u_i by summing over all the samples, and predicting the analysis with highest probability.

In our sampling algorithm, we avoid sampling the base distribution G_0 and the derived morpheme-pair distributions (G_{stm} etc.), instead using analytical closed forms. We explicitly sample the sparsity indicator variables $\vec{\lambda}$, the cognate indicator variables c_i, and latent word analyses (segmentations and Hebrew counterparts). To do so tractably, we use Gibbs sampling to draw each latent variable conditioned on our current sample of the others. Although the samples are no longer independent, they form a Markov chain whose stationary distribution is the true joint distribution defined by the model (Geman and Geman, 1984).

6.1 Sampling Word Analyses

For each undeciphered word, we need to sample a morphological segmentation $(u_{pre}, u_{stm}, u_{suf})_i$ along with latent morphemes in the known language $(h_{pre}, h_{stm}, h_{suf})_i$. More precisely, we need to sample three character-edit sequences $\vec{e}_{pre}, \vec{e}_{stm}, \vec{e}_{suf}$ which together yield the observed word u_i.

We break this into two sampling steps. First we sample the morphological segmentation of u_i, along with the part-of-speech pos of the latent stem cognate. To do so, we enumerate each possible segmentation and part-of-speech and calculate its joint conditional probability (for notational clarity, we leave implicit the conditioning on the other samples in the corpus):

$$P(u_{pre}, u_{stm}, u_{suf}, pos) = \sum_{\vec{e}_{stm}} P(\vec{e}_{stm}) \sum_{\vec{e}_{pre}} P(\vec{e}_{pre}|pos) \sum_{\vec{e}_{suf}} P(\vec{e}_{suf}|pos) \quad (2)$$

where the summations over character-edit sequences are restricted to those which yield the segmentation $(u_{pre}, u_{stm}, u_{suf})$ and a latent cognate with part-of-speech pos.

For a particular stem edit-sequence $\vec{e}_{stm}$, we compute its conditional probability in closed form according to a Chinese Restaurant Process (Antoniak, 1974). To do so, we use counts from the other sampled word analyses: $count_{stm}(\vec{e}_{stm})$ gives the number of times that the entire edit-sequence $\vec{e}_{stm}$ has been observed:

$$P(\vec{e}_{stm}) \propto \frac{count_{stm}(\vec{e}_{stm}) + \alpha \prod_i p(e_i)}{n + \alpha}$$

where n is the number of other word analyses sampled, and α is a fixed concentration parameter. The product $\prod_i p(e_i)$ gives the probability of $\vec{e}_{stm}$ according to the *base distribution* G_0. Since the parameters of G_0 are left unsampled, we use the marginalized form:

$$p(e) = \frac{v_e + count(e)}{\sum_{e'} v_{e'} + k} \quad (3)$$

where $count(e)$ is the number of times that character-edit e appears in distinct edit-sequences (across prefixes, stems, and suffixes), and k is the sum of these counts across all character-edits. Recall that v_e is a hyperparameter for the Dirichlet prior on G_0 and depends on the value of the corresponding indicator variable λ_e.

Once the segmentation $(u_{pre}, u_{stm}, u_{suf})$ and part-of-speech pos have been sampled, we proceed to sample the actual edit-sequences (and thus

latent morphemes counterparts). Now, instead of summing over the values in Equation 2, we instead sample from them.

6.2 Sampling Sparsity Indicators

Recall that each sparsity indicator λ_e determines the value of the corresponding hyperparameter v_e of the Dirichlet prior for the character-edit base distribution G_0. In addition, we have an unnormalized joint prior $P(\vec{\lambda}) = \frac{g(\vec{\lambda})}{Z}$ which encourages a sparse setting of these variables. To sample a particular λ_e, we consider the set $\vec{\lambda}$ in which $\lambda_e = 0$ and $\vec{\lambda}'$ in which $\lambda_e = 1$. We then compute:

$$P(\vec{\lambda}) \propto g(\vec{\lambda}) \cdot \frac{v_e^{[count(e)]}}{\sum_{e'} v_{e'}^{[k]}}$$

where k is the sum of counts for all edit operations, and the notation $a^{[b]}$ indicates the ascending factorial. Likewise, we can compute a probability for $\vec{\lambda}'$ with corresponding values v'_e.

6.3 Sampling Cognate Indicators

Finally, for each word u_i, we sample a corresponding indicator variable c_i. To do so, we calculate Equation 2 for all possible segmentations and parts-of-speech and sum the resulting values to obtain the conditional likelihood $P(u_i|c_i = 1)$. We also calculate $P(u_i|c_i = 0)$ using a uniform unigram character-level language model (and thus depends only on the number of characters in u_i). We then sample from among the two values:

$$P(u_i|c_i = 1) \cdot P(c_i = 1)$$
$$P(u_i|c_i = 0) \cdot P(c_i = 0)$$

6.4 High-level Resampling

Besides the individual sampling steps detailed above, we also consider several larger sampling moves in order to speed convergence. For example, for each type of edit-sequence $\vec{e}$ which has been sampled (and may now occur many times throughout the data), we consider a single joint move to another edit-sequence $\vec{e'}$ (both of which yield the same lost language morpheme u). The details are much the same as above, and as before the set of possible edit-sequences is limited by the string u and the known language lexicon.

We also resample groups of the sparsity indicator variables $\vec{\lambda}$ in tandem, to allow a more rapid exploration of the probability space. For each character u, we block sample the entire set $\{\lambda_{(u,h)}\}_h$, and likewise for each character h.

6.5 Implementation Details

Many of the steps detailed above involve the consideration of all possible edit-sequences consistent with (i) a particular undeciphered word u_i and (ii) the entire lexicon of words in the known language (or some subset of words with a particular part-of-speech). In particular, we need to both sample from and sum over this space of possibilities repeatedly. Doing so by simple enumeration would needlessly repeat many sub-computations. Instead we use finite-state acceptors to compactly represent both the entire Hebrew lexicon as well as potential Hebrew word forms for each Ugaritic word. By intersecting two such FSAs and minimizing the result we can efficiently represent all potential Hebrew words for a particular Ugaritic word. We weight the edges in the FSA according to the base distribution probabilities (in Equation 3 above). Although these intersected acceptors have to be constantly reweighted to reflect changing probabilities, their topologies need only be computed once. One weighted correctly, marginals and samples can be computed using dynamic programming.

Even with a large number of sampling rounds, it is difficult to fully explore the latent variable space for complex unsupervised models. Thus a clever initialization is usually required to start the sampler in a high probability region. We initialize our model with the results of the HMM-based baseline (see section 8), and rule out character substitutions with probability < 0.05 according to the baseline.

7 Experiments

7.1 Corpus and Annotations

We apply our model to the ancient Ugaritic language (see Section 3 for background). Our undeciphered corpus consists of an electronic transcription of the Ugaritic tablets (Cunchillos et al., 2002). This corpus contains 7,386 unique word types. As our known language corpus, we use the Hebrew Bible, which is both geographically and temporally close to Ugaritic. To extract a Hebrew morphological lexicon we assume the existence of manual morphological and part-of-speech annotations (Groves and Lowery, 2006). We divide Hebrew stems into four main part-of-speech categories each with a distinct affix profile: Noun, Verb, Pronoun, and Particle. For each part-of-speech category, we determine the set of allowable affixes using the annotated Bible corpus.

	Words		Morphemes	
	type	token	type	token
Baseline	28.82%	46.00%	N/A	N/A
Our Model	60.42%	66.71%	75.07%	81.25%
No Sparsity	46.08%	54.01%	69.48%	76.10%

Table 1: Accuracy of cognate translations, measured with respect to complete word-forms and morphemes, for the HMM-based substitution cipher baseline, our complete model, and our model without the structural sparsity priors. Note that the baseline does not provide per-morpheme results, as it does not predict morpheme boundaries.

To evaluate the output of our model, we annotated the words in the Ugaritic lexicon with the corresponding Hebrew cognates found in the standard reference dictionary (del Olo Lete and Sanmartín, 2004). In addition, manual morphological segmentation was carried out with the guidance of a standard Ugaritic grammar (Schniedewind and Hunt, 2007). Although Ugaritic is an inflectional rather than agglutinative language, in its written form (which lacks vowels) words can easily be segmented (e.g. *wyplṭn* becomes *wy-plṭ-n*).

Overall, we identified Hebrew cognates for 2,155 word forms, covering almost 1/3 of the Ugaritic vocabulary.[4]

8 Evaluation Tasks and Results

We evaluate our model on four separate decipherment tasks: (i) Learning alphabetic mappings, (ii) translating cognates, (iii) identifying cognates, and (iv) morphological segmentation.

As a baseline for the first three of these tasks (learning alphabetic mappings and translating and identifying cognates), we adapt the HMM-based method of Knight et al. (2006) for learning letter substitution ciphers. In its original setting, this model was used to map written texts to spoken language, under the assumption that each character was emitted from a hidden phonemic state. In our adaptation, we assume instead that each Ugaritic character was generated by a hidden Hebrew letter. Hebrew character trigram transition probabilities are estimated using the Hebrew Bible, and Hebrew to Ugaritic character emission probabilities are learned using EM. Finally, the highest probability sequence of latent Hebrew letters is predicted for each Ugaritic word-form, using Viterbi decoding.

Alphabetic Mapping The first essential step towards successful decipherment is recovering the mapping between the symbols of the lost language and the alphabet of a known language. As a gold standard for this comparison, we use the well-established relationship between the Ugaritic and Hebrew alphabets (Hetzron, 1997). This mapping is not one-to-one but is generally quite sparse. Of the 30 Ugaritic symbols, 28 map predominantly to a single Hebrew letter, and the remaining two map to two different letters. As the Hebrew alphabet contains only 22 letters, six map to two distinct Ugaritic letters and two map to three distinct Ugaritic letters.

We recover our model's predicted alphabetic mappings by simply examining the sampled values of the binary indicator variables $\lambda_{u,h}$ for each Ugaritic-Hebrew letter pair (u, h). Due to our structural sparsity prior $P(\vec{\lambda})$, the predicted mappings are sparse: each Ugaritic letter maps to only a single Hebrew letter, and most Hebrew letters map to only a single Ugaritic letter. To recover alphabetic mappings from the HMM substitution cipher baseline, we predict the Hebrew letter h which maximizes the model's probability $P(h|u)$, for each Ugaritic letter u.

To evaluate these mappings, we simply count the number of Ugaritic letters that are correctly mapped to one of their Hebrew reflexes. By this measure, the baseline recovers correct mappings for 22 out of 30 Ugaritic characters (73.3%). Our model recovers correct mappings for all but one (very low frequency) Ugaritic characters, yielding 96.67% accuracy.

Cognate Decipherment We compare the decipherment accuracy for Ugaritic words that have corresponding Hebrew cognates. We evaluate our model's predictions on each distinct Ugaritic word-form at both the type and token level. As Table 1 shows, our method correctly translates over 60% of all distinct Ugaritic word-forms with Hebrew cognates and over 71% of the individual morphemes that compose them, outperforming the baseline by significant margins. Accuracy improves when the frequency of the word-forms is taken into account (token-level evaluation), indicating that the model is able to decipher frequent words more accurately than infre-

[4] We are confident that a large majority of Ugaritic words with known Hebrew cognates were thus identified. The remaining Ugaritic words include many personal and geographic names, words with cognates in other Semitic languages, and words whose etymology is uncertain.

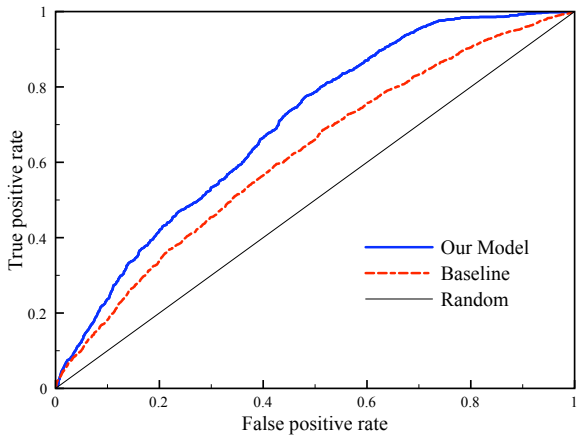

Figure 2: ROC curve for cognate identification.

quent words. We also measure the average Levenshtein distance between predicted and actual cognate word-forms. On average, our model's predictions lie 0.52 edit operations from the true cognate, whereas the baseline's predictions average a distance of 1.26 edit operations.

Finally, we evaluated the performance of our model when the structural sparsity constraints are not used. As Table 1 shows, performance degrades significantly in the absence of these priors, indicating the importance of modeling the sparsity of character mappings.

Cognate identification We evaluate our model's ability to identify cognates using the sampled indicator variables c_i. As before, we compare our performance against the HMM substitution cipher baseline. To produce baseline cognate identification predictions, we calculate the probability of each latent Hebrew letter sequence predicted by the HMM, and compare it to a uniform character-level Ugaritic language model (as done by our model, to avoid automatically assigning higher cognate probability to shorter Ugaritic words). For both our model and the baseline, we can vary the threshold for cognate identification by raising or lowering the cognate prior $P(c_i)$. As the prior is set higher, we detect more true cognates, but the false positive rate increases as well.

Figure 2 shows the ROC curve obtained by varying this prior both for our model and the baseline. At all operating points, our model outperforms the baseline, and both models always predict better than chance. In practice for our model, we use a high cognate prior, thus only ruling out those Ugaritic word-forms which are very unlikely to have Hebrew cognates.

	precision	recall	f-measure
Morfessor	88.87%	67.48%	76.71%
Our Model	86.62%	90.53%	88.53%

Table 2: Morphological segmentation accuracy for a standard unsupervised baseline and our model.

Morphological segmentation Finally, we evaluate the accuracy of our model's morphological segmentation for Ugaritic words. As a baseline for this comparison, we use Morfessor Categories-MAP (Creutz and Lagus, 2007). As Table 2 shows, our model provides a significant boost in performance, especially for recall. This result is consistent with previous work showing that morphological annotations can be projected to new languages lacking annotation (Yarowsky et al., 2000; Snyder and Barzilay, 2008), but generalizes those results to the case where parallel data is unavailable.

9 Conclusion and Future Work

In this paper we proposed a method for the automatic decipherment of lost languages. The key strength of our model lies in its ability to incorporate a range of linguistic intuitions in a statistical framework.

We hope to address several issues in future work. Our model fails to take into account the known *frequency* of Hebrew words and morphemes. In fact, the most common error is incorrectly translating the masculine plural suffix (*-m*) as the third person plural possessive suffix (*-m*) rather than the correct and much more common plural suffix (*-ym*). Also, even with the correct alphabetic mapping, many words can only be deciphered by examining their literary context. Our model currently operates purely on the vocabulary level and thus fails to take this contextual information into account. Finally, we intend to explore our model's predictive power when the family of the lost language is unknown.[5]

[5]The authors acknowledge the support of the NSF (CAREER grant IIS-0448168, grant IIS-0835445, and grant IIS-0835652) and the Microsoft Research New Faculty Fellowship. Thanks to Michael Collins, Tommi Jaakkola, and the MIT NLP group for their suggestions and comments. Any opinions, findings, conclusions, or recommendations expressed in this paper are those of the authors, and do not necessarily reflect the views of the funding organizations.

References

C. E. Antoniak. 1974. Mixtures of Dirichlet processes with applications to bayesian nonparametric problems. *The Annals of Statistics*, 2:1152–1174, November.

Alexandre Bouchard, Percy Liang, Thomas Griffiths, and Dan Klein. 2007. A probabilistic approach to diachronic phonology. In *Proceedings of EMNLP*, pages 887–896.

Mathias Creutz and Krista Lagus. 2007. Unsupervised models for morpheme segmentation and morphology learning. *ACM Transactions on Speech and Language Processing*, 4(1).

Jesus-Luis Cunchillos, Juan-Pablo Vita, and Jose-Ángel Zamora. 2002. Ugaritic data bank. CD-ROM.

Gregoria del Olo Lete and Joaquín Sanmartín. 2004. *A Dictionary of the Ugaritic Language in the Alphabetic Tradition*. Number 67 in Handbook of Oriental Studies. Section 1 The Near and Middle East. Brill.

Pascale Fung and Kathleen McKeown. 1997. Finding terminology translations from non-parallel corpora. In *Proceedings of the Annual Workshop on Very Large Corpora*, pages 192–202.

S. Geman and D. Geman. 1984. Stochastic relaxation, gibbs distributions and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12:609–628.

Alan Groves and Kirk Lowery, editors. 2006. *The Westminster Hebrew Bible Morphology Database*. Westminster Hebrew Institute, Philadelphia, PA, USA.

Jacques B. M. Guy. 1994. An algorithm for identifying cognates in bilingual wordlists and its applicability to machine translation. *Journal of Quantitative Linguistics*, 1(1):35–42.

Aria Haghighi, Percy Liang, Taylor Berg-Kirkpatrick, and Dan Klein. 2008. Learning bilingual lexicons from monolingual corpora. In *Proceedings of the ACL/HLT*, pages 771–779.

Robert Hetzron, editor. 1997. *The Semitic Languages*. Routledge.

H. Ishwaran and J.S. Rao. 2005. Spike and slab variable selection: frequentist and Bayesian strategies. *The Annals of Statistics*, 33(2):730–773.

Kevin Knight and Richard Sproat. 2009. Writing systems, transliteration and decipherment. NAACL Tutorial.

K. Knight and K. Yamada. 1999. A computational approach to deciphering unknown scripts. In *ACL Workshop on Unsupervised Learning in Natural Language Processing*.

Kevin Knight, Anish Nair, Nishit Rathod, and Kenji Yamada. 2006. Unsupervised analysis for decipherment problems. In *Proceedings of the COLING/ACL*, pages 499–506.

Philipp Koehn and Kevin Knight. 2002. Learning a translation lexicon from monolingual corpora. In *Proceedings of the ACL-02 workshop on Unsupervised lexical acquisition*, pages 9–16.

Grzegorz Kondrak. 2001. Identifying cognates by phonetic and semantic similarity. In *Proceeding of NAACL*, pages 1–8.

Grzegorz Kondrak. 2009. Identification of cognates and recurrent sound correspondences in word lists. *Traitement Automatique des Langues*, 50(2):201–235.

John B. Lowe and Martine Mazaudon. 1994. The reconstruction engine: a computer implementation of the comparative method. *Computational Linguistics*, 20(3):381–417.

Reinhard Rapp. 1999. Automatic identification of word translations from unrelated english and german corpora. In *Proceedings of the ACL*, pages 519–526.

Andrew Robinson. 2002. *Lost Languages: The Enigma of the World's Undeciphered Scripts*. McGraw-Hill.

William M. Schniedewind and Joel H. Hunt. 2007. *A Primer on Ugaritic: Language, Culture and Literature*. Cambridge University Press.

Mark S. Smith, editor. 1955. *Untold Stories: The Bible and Ugaritic Studies in the Twentieth Century*. Hendrickson Publishers.

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

Wilfred Watson and Nicolas Wyatt, editors. 1999. *Handbook of Ugaritic Studies*. Brill.

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.