Unifying Bayesian Inference and Distributed Representations for Improved Decipherment

Abstract

We introduce into Bayesian decipherment a base distribution derived from similarities of word embeddings. We use Dirichlet multinomial regression (Mimno and McCallum, 2012) to learn a mapping between ciphertext and plaintext word embeddings from *non-parallel* data. Experimental results show that the base distribution is highly beneficial to decipherment, improving state-of-the-art decipherment accuracy from 29.0% to 64.7% for Spanish/English, and from 5.1% to 11.2% for Malagasy/English.

1 Introduction

Tremendous advances in Machine Translation (MT) have been made since we began applying automatic learning techniques to learn translation rules automatically from parallel data. However, reliance on parallel data also limits the development and application of high-quality MT systems, as the amount of parallel data is far from adequate in low-density languages and domains.

In general, it is easier to obtain non-parallel monolingual data. The ability to learn translations from monolingual data can alleviate obstacles caused by insufficient parallel data. Motivated by this idea, researchers have proposed different approaches to tackle this problem. They can be largely divided into two groups.

The first group is based on the idea proposed by Rapp (1995), in which words are represented as context vectors, and two words are likely to be translations if their context vectors are similar. Initially, the vectors contained only context words. Later extensions introduced more features (Haghighi et al., 2008; Garera et al., 2009; Bergsma and Van Durme, 2011; Daumé and Jagarlamudi, 2011; Irvine and Callison-Burch, 2013b; Irvine and Callison-Burch, 2013a), and used more abstract representation such as word embeddings (Klementiev et al., 2012).

Another promising approach to solve this problem is decipherment. It has drawn significant amounts of interest in the past few years (Ravi and Knight, 2011; Nuhn et al., 2012; Dou and Knight, 2013; Ravi, 2013) and has been shown to improve end-to-end translation. Decipherment views a foreign language as a cipher for English and finds a translation table that converts foreign texts into sensible English.

Both approaches have been shown to improve quality of MT systems for domain adaptation (Daumé and Jagarlamudi, 2011; Dou and Knight, 2012; Irvine et al., 2013) and low density languages (Irvine and Callison-Burch, 2013a; Dou et al., 2014). Meanwhile, they have their own advantages and disadvantages. While context vectors can take larger context into account, it requires high quality seed lexicons to learn a mapping between two vector spaces. In contrast, decipherment does not depend on any seed lexicon, but only looks at a limited n-gram context.

In this work, we take advantage of both approaches and combine them in a joint inference process. More specifically, we extend previous work in large scale Bayesian decipherment by introducing a better base distribution derived from similarities of word embedding vectors. The main contributions of this work are:

- We propose a new framework that combines the two main approaches to finding translations from monolingual data only.
- We develop a new base-distribution technique that improves state-of-the art decipher-

ment accuracy by a factor of two for Spanish/English and Malagasy/English.

- We make our software available for future research, functioning as a kind of GIZA for non-parallel data.

2 Decipherment Model

In this section, we describe the previous decipherment framework that we build on. This framework follows Ravi and Knight (2011), who built an MT system using only non-parallel data for translating movie subtitles; Dou and Knight (2012) and Nuhn et al. (2012), who scaled decipherment to larger vocabularies; and Dou and Knight (2013), who improved decipherment accuracy with dependency relations between words.

Throughout this paper, we use f to denote target language or ciphertext tokens, and e to denote source language or plaintext tokens. Given ciphertext $\mathbf{f} : f_1...f_n$, the task of decipherment is to find a set of parameters $P(f_i|e_i)$ that convert f to sensible plaintext. The ciphertext $\mathbf{f}$ can either be full sentences (Ravi and Knight, 2011; Nuhn et al., 2012) or simply bigrams (Dou and Knight, 2013). Since using bigrams and their counts speeds up decipherment, in this work, we treat $\mathbf{f}$ as bigrams, where $\mathbf{f} = \{\mathbf{f}^n\}_{n=1}^N = \{f_1^n, f_2^n\}_{n=1}^N$.

Motivated by the idea from Weaver (1955), we model an observed cipher bigram $\mathbf{f}^n$ with the following generative story:

- First, a language model $P(\mathbf{e})$ generates a sequence of two plaintext tokens e_1^n, e_2^n with probability $P(e_1^n, e_2^n)$.
- Then, substitute e_1^n with f_1^n and e_2^n with f_2^n with probability $P(f_1^n \mid e_1^n) \cdot P(f_2^n \mid e_2^n)$.

Based on the above generative story, the probability of any cipher bigram $\mathbf{f^n}$ is:

$$P(\mathbf{f}^n) = \sum_{e_1 e_2} P(e_1 e_2) \prod_{i=1}^{2} P(f_i^n \mid e_i)$$

The probability of the ciphertext corpus,

$$P(\{\mathbf{f}^n\}_{n=1}^N) = \prod_{n=1}^{N} P(\mathbf{f}^n)$$

There are two sets of parameters in the model: the channel probabilities $\{P(f \mid e)\}$ and the bigram language model probabilities $\{P(e' \mid e)\}$, where f ranges over the ciphertext vocabulary and e, e' range over the plaintext vocabulary. Given a plaintext bigram language model, the training objective is to learn $P(f \mid e)$ that maximize $P(\{\mathbf{f}^n\}_{n=1}^N)$. When formulated like this, one can directly apply EM to solve the problem (Knight et al., 2006). However, EM has time complexity $O(N \cdot V_e^2)$ and space complexity $O(V_f \cdot V_e)$, where V_f, V_e are the sizes of ciphertext and plaintext vocabularies respectively, and N is the number of cipher bigrams. This makes the EM approach unable to handle long ciphertexts with large vocabulary size.

An alternative approach is Bayesian decipherment (Ravi and Knight, 2011). We assume that $P(f \mid e)$ and $P(e' \mid e)$ are drawn from a Dirichet distribution with hyper-parameters $\alpha_{f,e}$ and $\alpha_{e,e'}$, that is:

$$P(f \mid e) \sim Dirichlet(\alpha_{f,e})$$
$$P(e \mid e') \sim Dirichlet(\alpha_{e,e'}).$$

The remainder of the generative story is the same as the noisy channel model for decipherment. In the next section, we describe how we learn the hyper parameters of the Dirichlet prior. Given $\alpha_{f,e}$ and $\alpha_{e,e'}$, The joint likelihood of the complete data and the parameters,

$$\begin{aligned}
&P(\{\mathbf{f}^n, \mathbf{e}^n\}_{n=1}^N, \{P(f \mid e)\}, \{P(e \mid e')\}) \\
&= P(\{\mathbf{f}^n \mid \mathbf{e}^n\}_{n=1}^N, \{P(f \mid e)\}) \\
&P(\{\mathbf{e}^n\}_{n=1}^N, P(e \mid e')) \\
&= \prod_e \frac{\Gamma\left(\sum_f \alpha_{f,e}\right)}{\prod_f \Gamma\left(\alpha_{e,f}\right)} \prod_f P(f \mid e)^{\#(e,f)+\alpha_{e,f}-1} \\
&\prod_e \frac{\Gamma\left(\sum_{e'} \alpha_{e,e'}\right)}{\prod_{e'} \Gamma\left(\alpha_{e,e'}\right)} \prod_f P(e \mid e')^{\#(e,e')+\alpha_{e,e'}-1},
\end{aligned} \tag{1}$$

where $\#(e, f)$ and $\#(e, e')$ are the counts of the translated word pairs and plaintext bigram pairs in the complete data, and $\Gamma(\cdot)$ is the Gamma function. Unlike EM, in Bayesian decipherment, we no longer search for parameters $P(f \mid e)$ that maximize the likelihood of the observed ciphertext. Instead, we draw samples from posterior distribution of the plaintext sequences given the ciphertext. Under the above Bayesian decipherment model, it turns out that the probability of a particular cipher word f_j having a value k, given the current plaintext word e_j, and the samples for all the other ciphertext and plaintext words, $\mathbf{f}_{-j}$ and $\mathbf{e}_{-j}$, is:

$$P(f_j = k \mid e_j, \mathbf{f}_{-j}, \mathbf{e}_{-j}) = \frac{\#(k, e_j)_{-j} + \alpha_{e_j,k}}{\#(e_j)_{-j} + \sum_f \alpha_{e_j,f}}. \tag{2}$$

Where, $\#(k, e_j)_{-j}$ and $\#(e_j)_{-j}$ are the counts of the ciphertext, plaintext word pair and plaintext word in the samples excluding f_j and e_j. Similarly, the probability of a plaintext word e_j taking a value l given samples for all other plaintext words,

$$P(e_j = l \mid \mathbf{e}_{-j}) = \frac{\#(l, e_{j-1})_{-j} + \alpha_{l,e_{j-1}}}{\#(e_{j-1})_{-j} + \sum_e \alpha_{e,e_{j-1}}}. \tag{3}$$

Since we have large amounts of plaintext data, we can train a high-quality dependency-bigram language model, $P_{LM}(e \mid e')$ and use it to guide our samples and learn a better posterior distribution. For that, we define $\alpha_{e,e'} = \alpha P_{LM}(e \mid e')$, and set α to be very high. The probability of a plaintext word (Equation 3) is now

$$P(e_j = l \mid \mathbf{e}_{-j}) \approx P_{LM}(l \mid e_{j-1}). \tag{4}$$

To sample from the posterior, we iterate over the observed ciphertext bigram tokens and use equations 2 and 4 to sample a plaintext token with probability

$$\begin{aligned} &P(e_j \mid \mathbf{e}_{-j}, \mathbf{f}) \propto P_{LM}(e_j \mid e_{j-1}) \\ &P_{LM}(e_{j+1} \mid e_j) P(f_j \mid e_j, \mathbf{f}_{-j}, \mathbf{e}_{-j}). \end{aligned} \tag{5}$$

In previous work (Dou and Knight, 2012), the authors use symmetric priors over the channel probabilities, where $\alpha_{e,f} = \alpha \frac{1}{V_f}$, and they set α to 1. Symmetric priors over word translation probabilities are a poor choice, as one would not a-priori expect plaintext words and ciphertext words to cooccur with equal frequency. Bayesian inference is a powerful framework that allows us to inject useful prior information into the sampling process, a feature that we would like to use. In the next section, we will describe how we model and learn better priors using distributional properties of words. In subsequent sections, we show significant improvements over the baseline by learning better priors.

3 Base Distribution with Cross-Lingual Word Similarities

As shown in the previous section, the base distribution in Bayesian decipherment is given independent of the inference process. A better base distribution can improve decipherment accuracy. Ideally, we should assign higher base distribution probabilities to word pairs that are similar.

One straightforward way is to consider orthographic similarities. This works for closely related languages, e.g., the English word "new" is translated as "neu" in German and "nueva" in Spanish. However, this fails when two languages are not closely related, e.g., Chinese/English. Previous work aims to discover translations from comparable data based on word context similarities. This is based on the assumption that words appearing in similar contexts have similar meanings. The approach straightforwardly discovers monolingual synonyms. However, when it comes to finding translations, one challenge is to draw a mapping between the different context spaces of the two languages. In previous work, the mapping is usually learned from a seed lexicon.

There has been much recent work in learning distributional vectors (embeddings) for words. The most popular among these is learned by the skip-gram and continuous-bag-of-words models (Mikolov et al., 2013a). In Mikolov et al. (2013b), the authors are able to successfully learn word translations using *linear transformations* between the source and target word vector-spaces. However, unlike our learning setting, their approach relied on large amounts of translation pairs learned from *parallel* data to train their linear transformations. Inspired by these approaches, we aim to exploit high-quality monolingual word embeddings to help learn better posterior distributions in unsupervised decipherment, without any parallel data.

In the previous section, we incorporated our pre-trained language model in $\alpha_{e,e'}$ to steer our sampling. In the same vein, we model $\alpha_{e,f}$ using pre-trained word embeddings, enabling us to improve our estimate of the posterior distribution. In Mimno and McCallum (2012), the authors develop topic models where the base distribution over topics is a log-linear model of observed document features, which permits learning better priors over topic distributions for each document. Similarly, we introduce a latent cross-lingual linear mapping M and define:

$$\alpha_{f,e} = \exp\{v_e^T M v_f\}, \tag{6}$$

where v_e and v_f are the pre-trained plaintext word and ciphertext word embeddings. M is the similarity matrix between the two embedding spaces. $\alpha_{f,e}$ can be thought of as the affinity of a plaintext word to be mapped to a ciphertext word. Rewriting the channel part of the joint likelihood in equation 1,

$$P(\{\mathbf{f}^n \mid \mathbf{e}^n\}_{n=1}^N, \{P(f \mid e)\}) = \prod_e \frac{\Gamma\left(\sum_f \exp\{v_e^T M v_f\}\right)}{\prod_f \Gamma\left(\exp\{v_e^T M v_f\}\right)} \prod_f P(f \mid e)^{\#(e,f)+\exp\{v_e^T M v_f\}-1}$$

Integrating out the channel probabilities, the complete data log-likelihood of the observed ciphertext bigrams and the sampled plaintext bigrams,

$$P(\{\mathbf{f}^n \mid \mathbf{e}^n\} = \prod_e \frac{\Gamma\left(\sum_f \exp\{v_e^T M v_f\}\right)}{\prod_f \Gamma\left(\exp\{v_e^T M v_f\}\right)} \prod_e \frac{\prod_f \Gamma\left(\exp\{v_e^T M v_f\} + \#(e,f)\right)}{\Gamma\left(\sum_f \exp\{v_e^T M v_f\} + \#(e)\right)}.$$

We also add a $L2$ regularization penalty on the elements of M. The derivative of $\log P(\{\mathbf{f}^n \mid \mathbf{e}^n\} - \frac{\lambda}{2}\sum_{i,j} M_{i,j}^2$, where λ is the regularization weight, with respect to M,

$$\frac{\partial \log P(\{\mathbf{f}^n \mid \mathbf{e}^n\} - \frac{\lambda}{2}\sum_{i,j} M_{i,j}^2}{\partial M} = \sum_e \sum_f \exp\{v_e^T M v_f\} v_e v_f^T (\Psi\left(\sum_{f'} \exp\{v_e^T M v_{f'}\}\right) - \Psi\left(\sum_{f'} \exp\{v_e^T M v_{f'}\} + \#(e)\right) + + \Psi\left(\exp\{v_e^T M v_f\} + \#(e,f)\right) - \Psi\left(exp\{v_e^T M v_f\}\right) - \lambda M,$$

where we use

$$\frac{\partial \exp\{v_e^T M v_f\}}{\partial M} = \exp\{v_e^T M v_f\} \frac{\partial v_e^T M v_f}{\partial M} = \exp\{v_e^T M v_f\} v_e v_f^T.$$

$\Psi(\cdot)$ is the Digamma function, the derivative of $\log \Gamma(\cdot)$. Again, following Mimno and McCallum (2012), we train the similarity matrix M with stochastic EM. In the E-step, we sample plaintext words for the observed ciphertext using equation 5 and in the M-step, we learn M that maximizes $\log P(\{\mathbf{f}^n \mid \mathbf{e}^n\})$ with stochastic gradient descent. After learning M, we can set

$$\alpha_{e,f} = \sum_{f'} \exp\{v_e^T M v_{f'}\} \frac{\exp\{v_e^T M v_f\}}{\sum_{f'} \exp\{v_e^T M v_{f'}\}} = \alpha_e m_{e,f}, \quad (7)$$

where α_e is the concentration parameter and $m_{e,f}$ is an element of the base measure $\mathbf{m}_e$ for plaintext word e. In practice, we find that α_e can be very large, overwhelming the counts from sampling when we only have a few ciphertext bigrams. Therefore, we use $\mathbf{m}_e$ and set α_e proportional to the data size.

4 Deciphering Spanish Gigaword

In this section, we describe our data and experimental conditions for deciphering Spanish into English.

4.1 Data

In our Spanish/English decipherment experiments, we use half of the Gigaword corpus as monolingual data, and a small amount of parallel data from Europarl *only for evaluation*. We keep only the 10k most frequent word types for both languages and replace all other word types with "UNK". We also exclude sentences longer than 40 tokens, which significantly slow down our parser. After preprocessing, the size of data for each language is shown in Table 1. While we use all the monolingual data shown in Table 1 to learn word embeddings, we only parse the AFP (Agence France-Presse) section of the Gigaword corpus to extract cipher dependency bigrams and build a plaintext language model. We also use GIZA (Och and Ney, 2003) to align Europarl parallel data to build a dictionary for evaluating our decipherment.

4.2 Systems

We implement a baseline system based on the work described in Dou and Knight (2013). The baseline system carries out decipherment on dependency bigrams. Therefore, we use the Bohnet

	Spanish	English
Training	992 million (Gigaword)	940 million (Gigaword)
Evaluation	1.1 million (Europarl)	1.0 million (Europarl)

Table 1: Size of data in tokens used in Spanish/English decipherment experiment

parser (Bohnet, 2010) to parse the AFP section of both Spanish and English versions of the Gigaword corpus. Since not all dependency relations are shared across the two languages, we do not extract all dependency bigrams. Instead, we only use bigrams with dependency relations from the following list:

- Verb / Subject
- Verb / Object
- Preposition / Object
- Noun / Noun-Modifier

We denote the system that uses our new method as **DMRE** (Dirichlet Multinomial Regression with Embedings). The system is the same as the baseline except that it uses a base distribution derived from word embeddings similarities. Word embeddings are learned using word2vec (Mikolov et al., 2013a).

For all the systems, language models are built using the SRILM toolkit (Stolcke, 2002). We use the modified Kneser-Ney (Kneser and Ney, 1995) algorithm for smoothing.

4.3 Sampling Procedure

Motivated by the previous work, we use multiple random restarts and an iterative sampling process to improve decipherment (Dou and Knight, 2012). As shown in Figure 1, we start a few sampling processes each with a different random sample. Then results from different runs are combined to initiate the next sampling iteration. The details of the sampling procedure are listed below:

1. Extract dependency bigrams from parsing outputs and collect their counts.
2. Keep bigrams whose counts are greater than a threshold t. Then start N different randomly seeded and initialized sampling processes. Perform sampling.
3. At the end of sampling, extract word translation pairs (f, e) from the final sample. Estimate translation probabilities $P(e|f)$ for each pair. Then construct a translation table by keeping translation pairs (f, e) seen in more than one decipherment and use the average $P(e|f)$ as the new translation probability.
4. Start N different sampling processes again. Initialize the first samples with the translation pairs obtained from the previous step (for each dependency bigram f_1, f_2, find an English sequence e_1, e_2, whose $P(e_1|f_1) \cdot P(e_2|f_2) \cdot P(e_1, e_2)$is the highest). Initialize similarity matrix M with one learned by previous sampling process whose posterior probability is highest. Go to the third step, repeat until it converges.
5. Lower the threshold t to include more bigrams into the sampling process. Go to the second step, and repeat until $t = 1$.

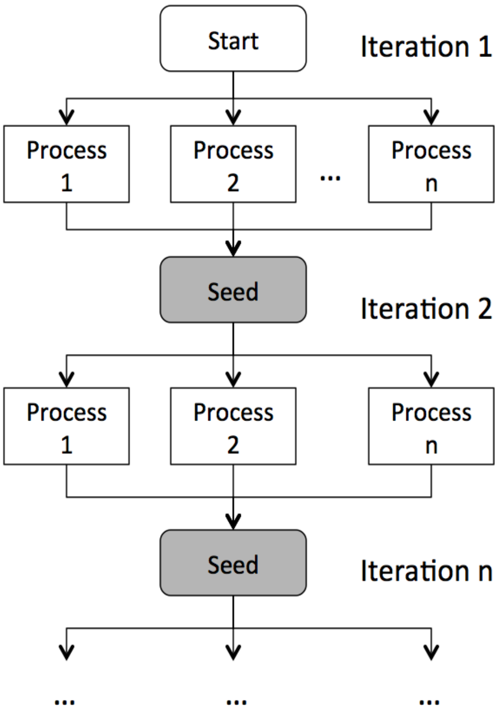

Figure 1: Iterative sampling procedures

The sampling process consists of sampling and learning of similarity matrix M. The sampling process creates training examples for learning M, and the new M is used to update the base distribution for sampling. In our Spanish/English decipherment experiments, we use 10 different random starts. As pointed out in section 3, setting α_e to it's theoretical value (equation 7) gives poor results as it can be quite large. In experiments, we set α_e to a small value for the smaller data sets and increase it as more ciphtertext becomes available.

	Malagasy	English
Training	16 million (Web)	1.2 billion (Gigaword and Web)
Evaluation	2.0 million (GlobalVoices)	1.8 million (GlobalVoices)

Table 2: Size of data in tokens used in Malagasy/English decipherment experiment. GlobalVoices is a parallel corpus.

We find that using the learned base distribution always improves decipherment accuracy, however, certain ranges are better for a given data size. We find that α_e values of $1, 2$, and 5 for ciphertexts with 100k, 1 million, and 10 million tokens respectively works well for decipherment.

5 Deciphering Malagasy

Despite spoken in Africa, Malagasy has its root in Asia, and belongs to the Malayo-Polynesian branch of the Austronesian language family. Malagasy and English have very different word order (VOS versus SVO). Generally, Malagasy is a typical head-initial language: Determiners precede nouns, while other modifiers and relative clauses follow nouns (e.g. ny "the" ankizilahy "boy" kely "little"). The significant differences in word order pose great challenges for both parsing and decipherment.

5.1 Data

Table 2 lists the sizes of monolingual and parallel data used in this experiment, released by Dou et al. (2014). The monolingual data in Malagasy contains news text collected from Madagascar websites. The English monolingual data contains Gigaword and an additional 300 million tokens of African news. Parallel data (used for evaluation only) is collected from GlobalVoices, a multilingual news website, where volunteers translate news into different languages.

5.2 Systems

The baseline system is the same as the baseline used in Spanish/English decipherment experiments. We use data provided in previous work (Dou et al., 2014) to build a Malagasy dependency parser. For English, we use the Turbo parser, trained on the Penn Treebank (Martins et al., 2013).

Because the Malagasy parser does not predict dependency relation types, we use the following head-child part-of-speech (POS) tag patterns to select a subset of dependency bigrams for decipherment:

- Verb / Noun
- Verb / Proper Noun
- Verb / Personal Pronoun
- Preposition / Noun
- Preposision / Proper Noun
- Noun / Adjective
- Noun / Determiner
- Noun / Verb Particle
- Noun / Verb Noun
- Noun / Cardinal
- Noun / Noun

5.3 Sampling Procedure

We use the same sampling protocol designed for Spanish/English decipherment. We double the number of random starts to 20. Further more, compared with Spanish/English decipherment, we find the base distribution plays a more important role in achieving higher decipherment accuracy for Malagasy/English. Therefore, we set α to 10, 50, and 200 when deciphering 100k, 1 million, and 20 million token ciphtertexts, respectively.

6 Results

In this section, we first compare decipherment accuracy of the baseline with our new approach. Then, we evaluate the quality of the base distribution through visualization.

We use top-5 type accuracy as our evaluation metric for decipherment. Given a word type f in Spanish, we find top-5 translation pairs (f, e) ranked by $P(e|f)$ from the learned decipherent translation table. If any pair (f, e) can also be found in a gold translation lexicon T_{gold}, we treat the word type f as correctly deciphered. Let $|C|$ be the number of word types correctly deciphered, and $|V|$ be the total number of word types evaluated. We define type accuracy as $\frac{|C|}{|V|}$.

To create T_{gold}, we use GIZA to align a small amount of Spanish/English parallel text (1 million tokens for each language), and use the lexicon derived from the alignment as our gold translation lexicon. T_{gold} contains a subset of 4233 word types in the 5k most frequent word types, and 7479 word types in the top 10k frequent word types. We decipher the 10k most frequent Span-

	Spanish/English				Malagasy/English			
Top	5k		10k		5k		10k	
System	Baseline	DMRE	Baseline	DMRE	Baseline	DMRE	Baseline	DMRE
100k	1.9	12.4	1.1	7.1	1.2	2.7	0.6	1.4
1 million	7.3	37.7	4.2	23.6	2.5	5.8	1.3	3.2
10 million	29.0	64.7	15.9	43.7	5.4	11.2	3.0	6.9

Table 3: Spanish/English, Malagasy/English decipherment top-5 accuracy (%) of 5k and 10k most frequent word types

ish word types to the 10k most frequent English word types, and evaluate decipherment accuracy on both the 5k most frequent word types as well as the full 10k word types.

We evaluate accuracy for the 5k and 10k most frequent word types for each language pair, and present them in Table 3.

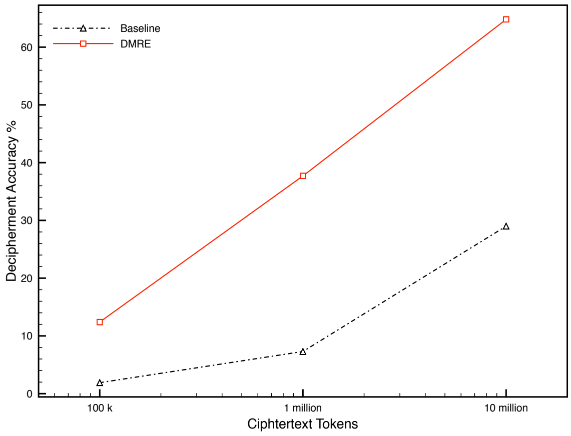

Figure 2: Learning curves of top-5 accuracy evaluated on 5k most frequent word types for Spanish/English decipherment.

We also present the learning curves of decipherment accuracy for the 5k most frequent word types. Figure 2 compares the baseline with **DMRE** in deciphering Spanish into English. Performance of the baseline is in line with previous work (Dou and Knight, 2013). (The accuracy reported here is higher as we evaluate top-5 accuracy for each word type.) With 100k tokens of Spanish text, the baseline achieves 1.9% accuracy, while **DMRE** reaches 12.4% accuracy, improving the baseline by over 6 times. The improvement holds consistently throughout the experiment. In the end, the baseline achieves 29.0% accuracy, while **DMRE** reaches 64.7% accuracy, over 2 times higher.

Figure 3 compares the baseline with our new approach in deciphering Malagasy into English. With 100k tokens of data, the baseline achieves 1.2% accuracy, and **DMRE** improves it to 2.4%. We observe consistent improvement throughout the experiment. In the end, the baseline accuracy obtains 5.8% accuracy, and **DMRE** improves it to 11.2%.

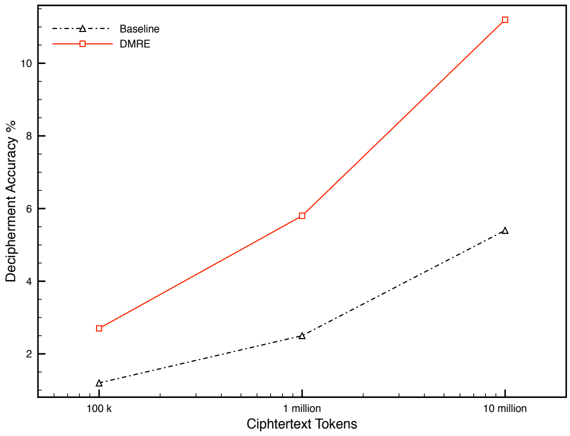

Figure 3: Learning curves of top-5 accuracy evaluated on 5k most frequent word types for Malagasy/English decipherment.

Overall, we achieve large consistent gains across both language pairs. We hypothesize the gain comes from a better base distribution that considers larger context information. This helps prevent the language model driving deicpherment to a wrong direction.

Since our learned transformation matrix M significantly improves decipherment accuracy, it's likely that it is *translation preserving*, that is, plaintext words are transformed from their native vector space to points in the ciphertext such that translations are close to each other. To visualize this effect, we take the $5k$ most frequent plaintext words and transform them into new embeddings in the ciphertext embedding space $v_{e'} = v_e^T M$, where M is learned from 10 million Spanish bigram data. We then project the $5k$ most frequent ciphertext words and the projected plaintext words from the joint embedding space into a $2-$dimensional space using t-sne (Van der Maaten and Hinton, 2008).

In Figure 4, we see an instance of a recurring phenomenon, where translation pairs are very close and sometimes even overlap each other, for example (judge, jueces), (secret, secretos). The word “magistrado” does not appear in our evaluation set. However, it is placed close to its possible translations. Thus, our approach is capable of learning word translations that cannot be discovered from limited parallel data.

We often also see *translation clusters*, where translations of groups of words are close to each other. For example, in Figure 5, we can see that time expressions in Spanish are quite close to their translations in English. Although better quality translation visualizations (Mikolov et al., 2013b) have been presented in previous work, they exploit large amounts of parallel data to learn the mapping between source and target words, while our transformation is learned on *non-parallel* data.

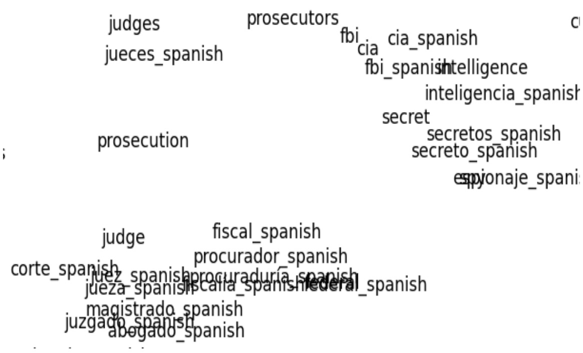

Figure 4: Translation pairs are often close and sometimes overlap each other. Words in spanish have been appended with _spanish

These results show that our approach can achieve high translation accuracy and discover novel word translations from non-parallel data.

7 Conclusion and Future Work

We propose a new framework that simultaneously performs decipherment and learns a cross-lingual mapping of word embeddings. Our method is both theoretically appealing and practically powerful. The mapping is used to give decipherment a better base distribution.

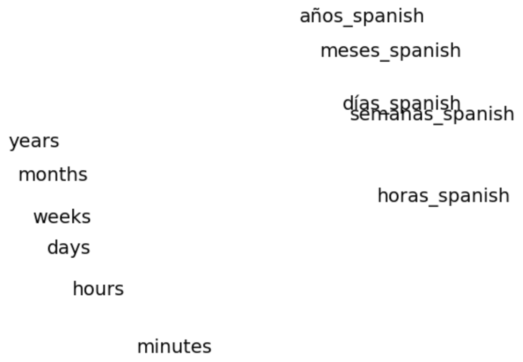

Figure 5: Semantic groups of word-translations appear close to each other.

Experimental results show that our new algorithm improves state-of-the-art decipherment accuracy significantly: from 29% to 64.7% for Spanish/English, and 5.1% to 11.2% for Malagasy/English. This improvement could lead to further advances in using monolingual data to improve end-to-end MT.

In the future, we will work on making the new method scale to much larger vocabulary sizes, and apply it to improve MT systems.

Acknowledgments

References

Shane Bergsma and Benjamin Van Durme. 2011. Learning bilingual lexicons using the visual similarity of labeled web images. In *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence - Volume Volume Three*. AAAI Press.

Bernd Bohnet. 2010. Top accuracy and fast dependency parsing is not a contradiction. In *Proceedings of the 23rd International Conference on Computational Linguistics*. Coling.

Hal Daumé, III and Jagadeesh Jagarlamudi. 2011. Domain adaptation for machine translation by mining unseen words. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics.

Qing Dou and Kevin Knight. 2012. Large scale decipherment for out-of-domain machine translation. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Association for Computational Linguistics.

Qing Dou and Kevin Knight. 2013. Dependency-based decipherment for resource-limited machine translation. In *Proceedings of the 2013 Conference* *on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Qing Dou, Ashish Vaswani, and Kevin Knight. 2014. Beyond parallel data: Joint word alignment and decipherment improves machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Nikesh Garera, Chris Callison-Burch, and David Yarowsky. 2009. Improving translation lexicon induction from monolingual corpora via dependency contexts and part-of-speech equivalences. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*. Association for Computational Linguistics.

Aria Haghighi, Percy Liang, Taylor Berg-Kirkpatrick, and Dan Klein. 2008. Learning bilingual lexicons from monolingual corpora. In *Proceedings of ACL-08: HLT*. Association for Computational Linguistics.

Ann Irvine and Chris Callison-Burch. 2013a. Combining bilingual and comparable corpora for low resource machine translation. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*. Association for Computational Linguistics, August.

Ann Irvine and Chris Callison-Burch. 2013b. Supervised bilingual lexicon induction with multiple monolingual signals. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics.

Ann Irvine, Chris Quirk, and Hal Daume III. 2013. Monolingual marginal matching for translation model adaptation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Alexandre Klementiev, Ivan Titov, and Binod Bhattarai. 2012. Inducing crosslingual distributed representations of words. In *Proceedings of COLING 2012*. The COLING 2012 Organizing Committee.

Reinhard Kneser and Hermann Ney. 1995. Improved backing-off for m-gram language modeling. *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*.

Kevin Knight, Anish Nair, Nishit Rathod, and Kenji Yamada. 2006. Unsupervised analysis for decipherment problems. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*. Association for Computational Linguistics.

Andre Martins, Miguel Almeida, and Noah A. Smith. 2013. Turning on the turbo: Fast third-order non-projective turbo parsers. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Tomas Mikolov, Quoc V Le, and Ilya Sutskever. 2013b. Exploiting similarities among languages for machine translation. *arXiv preprint arXiv:1309.4168*.

David Mimno and Andrew McCallum. 2012. Topic models conditioned on arbitrary features with dirichlet-multinomial regression. *arXiv preprint arXiv:1206.3278*.

Malte Nuhn, Arne Mauser, and Hermann Ney. 2012. Deciphering foreign language by combining language models and context vectors. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers - Volume 1*. Association for Computational Linguistics.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*.

Reinhard Rapp. 1995. Identifying word translations in non-parallel texts. In *Proceedings of the 33rd annual meeting on Association for Computational Linguistics*. Association for Computational Linguistics.

Sujith Ravi and Kevin Knight. 2011. Deciphering foreign language. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics.

Sujith Ravi. 2013. Scalable decipherment for machine translation via hash sampling. In *Proceedings of the 51th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Andreas Stolcke. 2002. SRILM - an extensible language modeling toolkit. In *Proceedings of the International Conference on Spoken Language Processing*.

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(2579-2605):85.

Warren Weaver, 1955. *Translation (1949). Reproduced in W.N. Locke, A.D. Booth (eds.).* MIT Press.