Learning Bilingual Lexicons from Monolingual Corpora

Aria Haghighi, Percy Liang, Taylor Berg-Kirkpatrick and **Dan Klein**
Computer Science Division, University of California at Berkeley
{aria42,pliang,tberg,klein}@cs.berkeley.edu

Abstract

We present a method for learning bilingual translation lexicons from monolingual corpora. Word types in each language are characterized by purely monolingual features, such as context counts and orthographic substrings. Translations are induced using a generative model based on canonical correlation analysis, which explains the monolingual lexicons in terms of latent matchings. We show that high-precision lexicons can be learned in a variety of language pairs and from a range of corpus types.

1 Introduction

Current statistical machine translation systems use parallel corpora to induce translation correspondences, whether those correspondences be at the level of phrases (Koehn, 2004), treelets (Galley et al., 2006), or simply single words (Brown et al., 1994). Although parallel text is plentiful for some language pairs such as English-Chinese or English-Arabic, it is scarce or even non-existent for most others, such as English-Hindi or French-Japanese. Moreover, parallel text could be scarce for a language pair even if monolingual data is readily available for both languages.

In this paper, we consider the problem of learning translations from monolingual sources alone. This task, though clearly more difficult than the standard parallel text approach, can operate on language pairs and in domains where standard approaches cannot. We take as input two monolingual corpora and perhaps some seed translations, and we produce as output a *bilingual lexicon*, defined as a list of word pairs deemed to be word-level translations. Precision and recall are then measured over these bilingual lexicons. This setting has been considered before, most notably in Koehn and Knight (2002) and Fung (1995), but the current paper is the first to use a probabilistic model and present results across a variety of language pairs and data conditions.

In our method, we represent each language as a *monolingual lexicon* (see figure 2): a list of word types characterized by monolingual feature vectors, such as context counts, orthographic substrings, and so on (section 5). We define a generative model over (1) a source lexicon, (2) a target lexicon, and (3) a matching between them (section 2). Our model is based on *canonical correlation analysis* (CCA)[1] and explains matched word pairs via vectors in a common latent space. Inference in the model is done using an EM-style algorithm (section 3).

Somewhat surprisingly, we show that it is possible to learn or extend a translation lexicon using monolingual corpora alone, in a variety of languages and using a variety of corpora, even in the absence of orthographic features. As might be expected, the task is harder when no seed lexicon is provided, when the languages are strongly divergent, or when the monolingual corpora are from different domains. Nonetheless, even in the more difficult cases, a sizable set of high-precision translations can be extracted. As an example of the performance of the system, in English-Spanish induction with our best feature set, using corpora derived from topically similar but non-parallel sources, the system obtains 89.0% precision at 33% recall.

[1] See Hardoon et al. (2003) for an overview.

Proceedings of ACL-08: HLT, pages 771–779,
Columbus, Ohio, USA, June 2008.

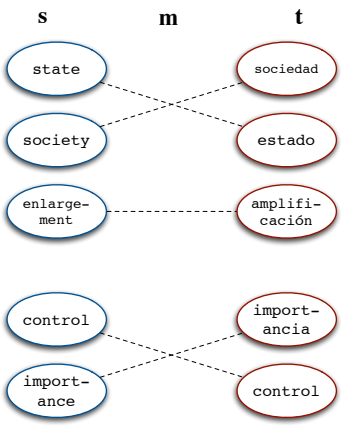

Figure 1: Bilingual lexicon induction: source word types $\mathbf{s}$ are listed on the left and target word types $\mathbf{t}$ on the right. Dashed lines between nodes indicate translation pairs which are in the matching $\mathbf{m}$.

2 Bilingual Lexicon Induction

As input, we are given a monolingual corpus S (a sequence of word tokens) in a *source* language and a monolingual corpus T in a *target* language. Let $\mathbf{s} = (s_1, \ldots, s_{n_S})$ denote n_S word types appearing in the source language, and $\mathbf{t} = (t_1, \ldots, t_{n_T})$ denote word types in the target language. Based on S and T, our goal is to output a matching $\mathbf{m}$ between $\mathbf{s}$ and $\mathbf{t}$. We represent $\mathbf{m}$ as a set of integer pairs so that $(i, j) \in \mathbf{m}$ if and only if s_i is matched with t_j.

2.1 Generative Model

We propose the following generative model over matchings $\mathbf{m}$ and word types $(\mathbf{s}, \mathbf{t})$, which we call *matching canonical correlation analysis (MCCA)*.

MCCA model

$\mathbf{m} \sim$ MATCHING-PRIOR [matching $\mathbf{m}$]
For each matched edge $(i, j) \in \mathbf{m}$:
  $z_{i,j} \sim \mathcal{N}(0, I_d)$ [latent concept]
  $f_S(s_i) \sim \mathcal{N}(W_S z_{i,j}, \Psi_S)$ [source features]
  $f_T(t_i) \sim \mathcal{N}(W_T z_{i,j}, \Psi_T)$ [target features]
For each unmatched source word type i:
  $f_S(s_i) \sim \mathcal{N}(0, \sigma^2 I_{d_S})$ [source features]
For each unmatched target word type j:
  $f_T(t_j) \sim \mathcal{N}(0, \sigma^2 I_{d_T})$ [target features]

First, we generate a matching $\mathbf{m} \in \mathcal{M}$, where $\mathcal{M}$ is the set of matchings in which each word type is matched to at most one other word type.[2] We take MATCHING-PRIOR to be uniform over $\mathcal{M}$.[3]

Then, for each matched pair of word types $(i, j) \in \mathbf{m}$, we need to generate the observed feature vectors of the source and target word types, $f_S(s_i) \in \mathbb{R}^{d_S}$ and $f_T(t_j) \in \mathbb{R}^{d_T}$. The feature vector of each word type is computed from the appropriate monolingual corpus and summarizes the word's monolingual characteristics; see section 5 for details and figure 2 for an illustration. Since s_i and t_j are translations of each other, we expect $f_S(s_i)$ and $f_T(t_j)$ to be connected somehow by the generative process. In our model, they are related through a vector $z_{i,j} \in \mathbb{R}^d$ representing the shared, language-independent concept.

Specifically, to generate the feature vectors, we first generate a random concept $z_{i,j} \sim \mathcal{N}(0, I_d)$, where I_d is the $d \times d$ identity matrix. The source feature vector $f_S(s_i)$ is drawn from a multivariate Gaussian with mean $W_S z_{i,j}$ and covariance Ψ_S, where W_S is a $d_S \times d$ matrix which transforms the language-independent concept $z_{i,j}$ into a language-dependent vector in the source space. The arbitrary covariance parameter $\Psi_S \succeq 0$ explains the source-specific variations which are not captured by W_S; it does not play an explicit role in inference. The target $f_T(t_j)$ is generated analogously using W_T and Ψ_T, conditionally independent of the source given $z_{i,j}$ (see figure 2). For each of the remaining unmatched source word types s_i which have not yet been generated, we draw the word type features from a baseline normal distribution with variance $\sigma^2 I_{d_S}$, with hyperparameter $\sigma^2 \gg 0$; unmatched target words are similarly generated.

If two word types are truly translations, it will be better to relate their feature vectors through the latent space than to explain them independently via the baseline distribution. However, if a source word type is not a translation of any of the target word types, we can just generate it independently without requiring it to participate in the matching.

[2] Our choice of $\mathcal{M}$ permits unmatched word types, but does not allow words to have multiple translations. This setting facilitates comparison to previous work and admits simpler models.

[3] However, non-uniform priors could encode useful information, such as rank similarities.

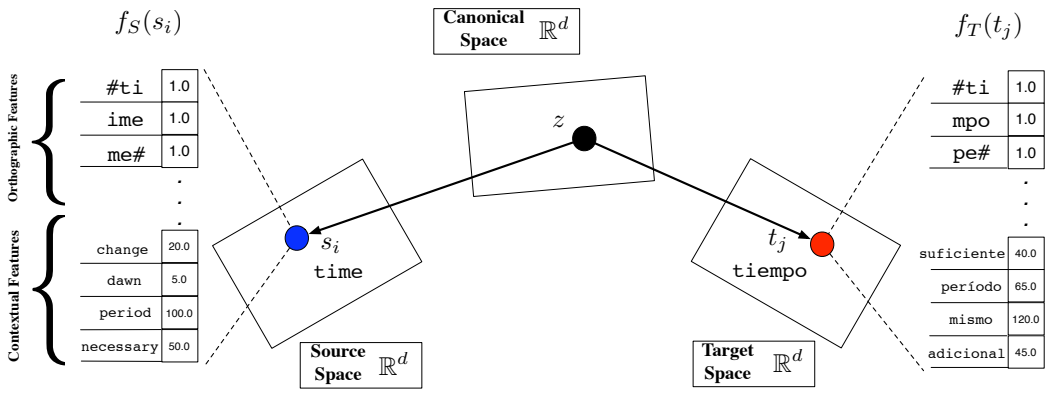

Figure 2: Illustration of our MCCA model. Each latent concept $z_{i,j}$ originates in the canonical space. The observed word vectors in the source and target spaces are generated independently given this concept.

3 Inference

Given our probabilistic model, we would like to maximize the log-likelihood of the observed data $(\mathbf{s}, \mathbf{t})$:

$$\ell(\theta) = \log p(\mathbf{s}, \mathbf{t}; \theta) = \log \sum_{\mathbf{m}} p(\mathbf{m}, \mathbf{s}, \mathbf{t}; \theta)$$

with respect to the model parameters $\theta = (W_S, W_T, \Psi_S, \Psi_T)$.

We use the hard (Viterbi) EM algorithm as a starting point, but due to modeling and computational considerations, we make several important modifications, which we describe later. The general form of our algorithm is as follows:

> Summary of learning algorithm
>
> **E-step**: Find the maximum weighted (partial) bipartite matching $\mathbf{m} \in \mathcal{M}$
> **M-step**: Find the best parameters θ by performing canonical correlation analysis (CCA)

M-step Given a matching $\mathbf{m}$, the M-step optimizes $\log p(\mathbf{m}, \mathbf{s}, \mathbf{t}; \theta)$ with respect to θ, which can be rewritten as

$$\max_{\theta} \sum_{(i,j) \in \mathbf{m}} \log p(s_i, t_j; \theta). \quad (1)$$

This objective corresponds exactly to maximizing the likelihood of the probabilistic CCA model presented in Bach and Jordan (2006), which proved that the maximum likelihood estimate can be computed by *canonical correlation analysis* (CCA). Intuitively, CCA finds d-dimensional subspaces $U_S \in \mathbb{R}^{d_S \times d}$ of the source and $U_T \in \mathbb{R}^{d_T \times d}$ of the target such that the components of the projections $U_S^\top f_S(s_i)$ and $U_T^\top f_T(t_j)$ are maximally correlated.[4]

U_S and U_T can be found by solving an eigenvalue problem (see Hardoon et al. (2003) for details). Then the maximum likelihood estimates are as follows: $W_S = C_{SS} U_S P^{1/2}$, $W_T = C_{TT} U_T P^{1/2}$, $\Psi_S = C_{SS} - W_S W_S^\top$, and $\Psi_T = C_{TT} - W_T W_T^\top$, where P is a $d \times d$ diagonal matrix of the canonical correlations, $C_{SS} = \frac{1}{|\mathbf{m}|} \sum_{(i,j) \in \mathbf{m}} f_S(s_i) f_S(s_i)^\top$ is the empirical covariance matrix in the source domain, and C_{TT} is defined analogously.

E-step To perform a conventional E-step, we would need to compute the posterior over all matchings, which is #P-complete (Valiant, 1979). On the other hand, hard EM only requires us to compute the best matching under the current model:[5]

$$\mathbf{m} = \operatorname*{argmax}_{\mathbf{m}'} \log p(\mathbf{m}', \mathbf{s}, \mathbf{t}; \theta). \quad (2)$$

We cast this optimization as a maximum weighted bipartite matching problem as follows. Define the edge weight between source word type i and target word type j to be

$$w_{i,j} = \log p(s_i, t_j; \theta) - \log p(s_i; \theta) - \log p(t_j; \theta), \quad (3)$$

[4] Since d_S and d_T can be quite large in practice and often greater than $|\mathbf{m}|$, we use Cholesky decomposition to re-represent the feature vectors as $|\mathbf{m}|$-dimensional vectors with the same dot products, which is all that CCA depends on.

[5] If we wanted softer estimates, we could use the agreement-based learning framework of Liang et al. (2008) to combine two tractable models.

which can be loosely viewed as a pointwise mutual information quantity. We can check that the objective $\log p(\mathbf{m}, \mathbf{s}, \mathbf{t}; \theta)$ is equal to the weight of a matching plus some constant C:

$$\log p(\mathbf{m}, \mathbf{s}, \mathbf{t}; \theta) = \sum_{(i,j) \in \mathbf{m}} w_{i,j} + C. \quad (4)$$

To find the optimal partial matching, edges with weight $w_{i,j} < 0$ are set to zero in the graph and the optimal full matching is computed in $O((n_S + n_T)^3)$ time using the Hungarian algorithm (Kuhn, 1955). If a zero edge is present in the solution, we remove the involved word types from the matching.[6]

Bootstrapping Recall that the E-step produces a partial matching of the word types. If too few word types are matched, learning will not progress quickly; if too many are matched, the model will be swamped with noise. We found that it was helpful to explicitly control the number of edges. Thus, we adopt a bootstrapping-style approach that only permits high confidence edges at first, and then slowly permits more over time. In particular, we compute the optimal full matching, but only retain the highest weighted edges. As we run EM, we gradually increase the number of edges to retain.

In our context, bootstrapping has a similar motivation to the annealing approach of Smith and Eisner (2006), which also tries to alter the space of hidden outputs in the E-step over time to facilitate learning in the M-step, though of course the use of bootstrapping in general is quite widespread (Yarowsky, 1995).

4 Experimental Setup

In section 5, we present developmental experiments in English-Spanish lexicon induction; experiments are presented for other languages in section 6. In this section, we describe the data and experimental methodology used throughout this work.

4.1 Data

Each experiment requires a source and target monolingual corpus. We use the following corpora:

- EN-ES-W: 3,851 Wikipedia articles with both English and Spanish bodies (generally not direct translations).
- EN-ES-P: 1st 100k sentences of text from the parallel English and Spanish Europarl corpus (Koehn, 2005).
- EN-ES(FR)-D: English: 1st 50k sentences of Europarl; Spanish (French): 2nd 50k sentences of Europarl.[7]
- EN-CH-D: English: 1st 50k sentences of Xinhua parallel news corpora;[8] Chinese: 2nd 50k sentences.
- EN-AR-D: English: 1st 50k sentences of 1994 proceedings of UN parallel corpora;[9] Arabic: 2nd 50k sentences.
- EN-ES-G: English: 100k sentences of English Gigaword; Spanish: 100k sentences of Spanish Gigaword.[10]

Note that even when corpora are derived from parallel sources, no explicit use is ever made of document or sentence-level alignments. In particular, our method is robust to permutations of the sentences in the corpora.

4.2 Lexicon

Each experiment requires a lexicon for evaluation. Following Koehn and Knight (2002), we consider lexicons over only noun word types, although this is not a fundamental limitation of our model. We consider a word type to be a noun if its most common tag is a noun in our monolingual corpus.[11] For

[6] Empirically, we obtained much better efficiency and even increased accuracy by replacing these marginal likelihood weights with a simple proxy, the distances between the words' mean latent concepts:

$$w_{i,j} = A - ||z_i^* - z_j^*||_2, \quad (5)$$

where A is a thresholding constant, $z_i^* = \mathbb{E}(z_{i,j} \mid f_S(s_i)) = P^{1/2} U_S^\top f_S(s_i)$, and z_j^* is defined analogously. The increased accuracy may not be an accident: whether two words are translations is perhaps better characterized directly by how close their latent concepts are, whereas log-probability is more sensitive to perturbations in the source and target spaces.

[7] Note that the although the corpora here are derived from a parallel corpus, there are no parallel sentences.

[8] LDC catalog # 2002E18.

[9] LDC catalog # 2004E13.

[10] These corpora contain no parallel sentences.

[11] We use the Tree Tagger (Schmid, 1994) for all POS tagging except for Arabic, where we use the tagger described in Diab et al. (2004).

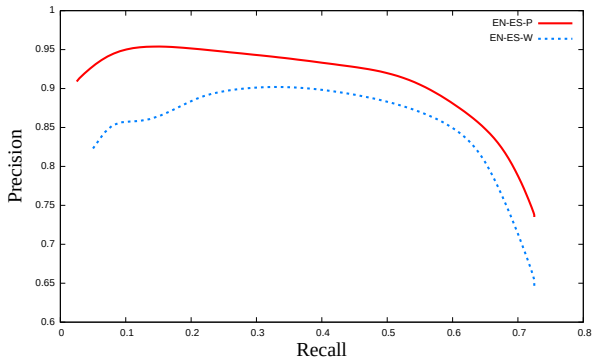

Figure 3: Example precision/recall curve of our system on EN-ES-P and EN-ES-W settings. See section 6.1.

all languages pairs except English-Arabic, we extract evaluation lexicons from the *Wiktionary* online dictionary. As we discuss in section 7, our extracted lexicons have low coverage, particularly for proper nouns, and thus all performance measures are (sometimes substantially) pessimistic. For English-Arabic, we extract a lexicon from 100k parallel sentences of UN parallel corpora by running the HMM intersected alignment model (Liang et al., 2008), adding (s, t) to the lexicon if s was aligned to t at least three times and more than any other word.

Also, as in Koehn and Knight (2002), we make use of a *seed lexicon*, which consists of a small, and perhaps incorrect, set of initial translation pairs. We used two methods to derive a seed lexicon. The first is to use the evaluation lexicon $\mathcal{L}_e$ and select the hundred most common noun word types in the source corpus which have translations in $\mathcal{L}_e$. The second method is to heuristically induce, where applicable, a seed lexicon using edit distance, as is done in Koehn and Knight (2002). Section 6.2 compares the performance of these two methods.

4.3 Evaluation

We evaluate a proposed lexicon $\mathcal{L}_p$ against the evaluation lexicon $\mathcal{L}_e$ using the F_1 measure in the standard fashion; precision is given by the number of proposed translations contained in the evaluation lexicon, and recall is given by the fraction of possible translation pairs proposed.[12] Since our model naturally produces lexicons in which each entry is associated with a weight based on the model, we can give a full precision/recall curve (see figure 3). We summarize these curves with both the best F_1 over all possible thresholds and various precisions p_x at recalls x. All reported numbers exclude evaluation on the seed lexicon entries, regardless of how those seeds are derived or whether they are correct.

[12] We should note that precision is not penalized for (s, t) if s does not have a translation in $\mathcal{L}_e$, and recall is not penalized for failing to recover multiple translations of s.

Setting	$p_{0.1}$	$p_{0.25}$	$p_{0.33}$	$p_{0.50}$	Best-F_1
EDITDIST	58.6	62.6	61.1	—-	47.4
ORTHO	76.0	81.3	80.1	52.3	55.0
CONTEXT	**91.1**	81.3	80.2	65.3	58.0
MCCA	87.2	**89.7**	**89.0**	**89.7**	**72.0**

Table 1: Performance of EDITDIST and our model with various features sets on EN-ES-W. See section 5.

In all experiments, unless noted otherwise, we used a seed of size 100 obtained from $\mathcal{L}_e$ and considered lexicons between the top $n = 2,000$ most frequent source and target noun word types which were not in the seed lexicon; each system proposed an already-ranked one-to-one translation lexicon amongst these n words. Where applicable, we compare against the EDITDIST baseline, which solves a maximum bipartite matching problem where edge weights are normalized edit distances. We will use MCCA (for *matching CCA*) to denote our model using the optimal feature set (see section 5.3).

5 Features

In this section, we explore feature representations of word types in our model. Recall that $f_S(\cdot)$ and $f_T(\cdot)$ map source and target word types to vectors in $\mathbb{R}^{d_S}$ and $\mathbb{R}^{d_T}$, respectively (see section 2). The features used in each representation are defined identically and derived only from the appropriate monolingual corpora. For a concrete example of a word type to feature vector mapping, see figure 2.

5.1 Orthographic Features

For closely related languages, such as English and Spanish, translation pairs often share many *orthographic* features. One direct way to capture orthographic similarity between word pairs is edit distance. Running EDITDIST (see section 4.3) on EN-

ES-W yielded 61.1 $p_{0.33}$, but precision quickly degrades for higher recall levels (see EDITDIST in table 1). Nevertheless, when available, orthographic clues are strong indicators of translation pairs.

We can represent orthographic features of a word type w by assigning a feature to each substring of length ≤ 3. Note that MCCA can learn regular orthographic correspondences between source and target words, which is something edit distance cannot capture (see table 5). Indeed, running our MCCA model with only orthographic features on EN-ES-W, labeled ORTHO in table 1, yielded 80.1 $p_{0.33}$, a 31% error-reduction over EDITDIST in $p_{0.33}$.

5.2 Context Features

While orthographic features are clearly effective for historically related language pairs, they are more limited for other language pairs, where we need to appeal to other clues. One non-orthographic clue that word types s and t form a translation pair is that there is a strong correlation between the source words used with s and the target words used with t. To capture this information, we define *context* features for each word type w, consisting of counts of nouns which occur within a window of size 4 around w. Consider the translation pair (`time`, `tiempo`) illustrated in figure 2. As we become more confident about other translation pairs which have active `period` and `periodico` context features, we learn that translation pairs tend to jointly generate these features, which leads us to believe that `time` and `tiempo` might be generated by a common underlying concept vector (see section 2).[13]

Using context features alone on EN-ES-W, our MCCA model (labeled CONTEXT in table 1) yielded a 80.2 $p_{0.33}$. It is perhaps surprising that context features alone, without orthographic information, can yield a best-F_1 comparable to EDITDIST.

5.3 Combining Features

We can of course combine context and orthographic features. Doing so yielded 89.03 $p_{0.33}$ (labeled MCCA in table 1); this represents a 46.4% error reduction in $p_{0.33}$ over the EDITDIST baseline. For the remainder of this work, we will use MCCA to refer to our model using both orthographic and context features.

[13] It is important to emphasize, however, that our current model does not directly relate a word type's role as a participant in the matching to that word's role as a context feature.

(a) Corpus Variation

Setting	$p_{0.1}$	$p_{0.25}$	$p_{0.33}$	$p_{0.50}$	Best-F_1
EN-ES-G	75.0	71.2	68.3	—-	49.0
EN-ES-W	87.2	89.7	89.0	89.7	72.0
EN-ES-D	91.4	94.3	92.3	89.7	63.7
EN-ES-P	97.3	94.8	93.8	92.9	77.0

(b) Seed Lexicon Variation

Corpus	$p_{0.1}$	$p_{0.25}$	$p_{0.33}$	$p_{0.50}$	Best-F_1
EDITDIST	58.6	62.6	61.1	—	47.4
MCCA	91.4	94.3	92.3	89.7	63.7
MCCA-AUTO	91.2	90.5	91.8	77.5	61.7

(c) Language Variation

Languages	$p_{0.1}$	$p_{0.25}$	$p_{0.33}$	$p_{0.50}$	Best-F_1
EN-ES	91.4	94.3	92.3	89.7	63.7
EN-FR	94.5	89.1	88.3	78.6	61.9
EN-CH	60.1	39.3	26.8	—-	30.8
EN-AR	70.0	50.0	31.1	—-	33.1

Table 2: (a) varying type of corpora used on system performance (section 6.1), (b) using a heuristically chosen seed compared to one taken from the evaluation lexicon (section 6.2), (c) a variety of language pairs (see section 6.3).

6 Experiments

In this section we examine how system performance varies when crucial elements are altered.

6.1 Corpus Variation

There are many sources from which we can derive monolingual corpora, and MCCA performance depends on the degree of similarity between corpora. We explored the following levels of relationships between corpora, roughly in order of closest to most distant:

- **Same Sentences**: EN-ES-P
- **Non-Parallel Similar Content**: EN-ES-W
- **Distinct Sentences, Same Domain**: EN-ES-D
- **Unrelated Corpora**: EN-ES-G

Our results for all conditions are presented in table 2(a). The predominant trend is that system performance degraded when the corpora diverged in

content, presumably due to context features becoming less informative. However, it is notable that even in the most extreme case of disjoint corpora from different time periods and topics (e.g. EN-ES-G), we are still able to recover lexicons of reasonable accuracy.

6.2 Seed Lexicon Variation

All of our experiments so far have exploited a small seed lexicon which has been derived from the evaluation lexicon (see section 4.3). In order to explore system robustness to heuristically chosen seed lexicons, we automatically extracted a seed lexicon similarly to Koehn and Knight (2002): we ran EDIT-DIST on EN-ES-D and took the top 100 most confident translation pairs. Using this automatically derived seed lexicon, we ran our system on EN-ES-D as before, evaluating on the top 2,000 noun word types not included in the automatic lexicon.[14] Using the automated seed lexicon, and still evaluating against our Wiktionary lexicon, MCCA-AUTO yielded 91.8 $p_{0.33}$ (see table 2(b)), indicating that our system can produce lexicons of comparable accuracy with a heuristically chosen seed. We should note that this performance represents no knowledge given to the system in the form of gold seed lexicon entries.

6.3 Language Variation

We also explored how system performance varies for language pairs other than English-Spanish. On English-French, for the disjoint EN-FR-D corpus (described in section 4.1), MCCA yielded 88.3 $p_{0.33}$ (see table 2(c) for more performance measures). This verified that our model can work for another closely related language-pair on which no model development was performed.

One concern is how our system performs on language pairs where orthographic features are less applicable. Results on disjoint English-Chinese and English-Arabic are given as EN-CH-D and EN-AR in table 2(c), both using only context features. In these cases, MCCA yielded much lower precisions of 26.8 and 31.0 $p_{0.33}$, respectively. For both languages, performance degraded compared to EN-ES-D and EN-FR-D, presumably due in part to the lack of orthographic features. However, MCCA still achieved surprising precision at lower recall levels. For instance, at $p_{0.1}$, MCCA yielded 60.1 and 70.0 on Chinese and Arabic, respectively. Figure 3 shows the highest-confidence outputs in several languages.

[14] Note that the 2,000 words evaluated here were not identical to the words tested on when the seed lexicon is derived from the evaluation lexicon.

(a) English-Spanish

Rank	Source	Target	Correct
1.	education	educación	Y
2.	pacto	pact	Y
3.	stability	estabilidad	Y
6.	corruption	corrupción	Y
7.	tourism	turismo	Y
9.	organisation	organización	Y
10.	convenience	conveniencia	Y
11.	syria	siria	Y
12.	cooperation	cooperación	Y
14.	culture	cultura	Y
21.	protocol	protocolo	Y
23.	north	norte	Y
24.	health	salud	Y
25.	action	reacción	N

(b) English-French

Rank	Source	Target	Correct
3.	xenophobia	xénophobie	Y
4.	corruption	corruption	Y
5.	subsidiarity	subsidiarité	Y
6.	programme	programme-cadre	N
8.	traceability	traçabilité	Y

(c) English-Chinese

Rank	Source	Target	Correct
1.	prices	价格	Y
2.	network	网络	Y
3.	population	人口	Y
4.	reporter	孙	N
5.	oil	石油	Y

Table 3: Sample output from our (a) Spanish, (b) French, and (c) Chinese systems. We present the highest confidence system predictions, where the only editing done is to ignore predictions which consist of identical source and target words.

6.4 Comparison To Previous Work

There has been previous work in extracting translation pairs from non-parallel corpora (Rapp, 1995; Fung, 1995; Koehn and Knight, 2002), but generally not in as extreme a setting as the one considered here. Due to unavailability of data and specificity in experimental conditions and evaluations, it is not possible to perform exact comparisons. How-

(a) Example Non-Cognate Pairs

health	salud
traceability	rastreabilidad
youth	juventud
report	informe
advantages	ventajas

(b) Interesting Incorrect Pairs

liberal	partido
Kirkhope	Gorsel
action	reacción
Albanians	Bosnia
a.m.	horas
Netherlands	Bretaña

Table 4: System analysis on EN-ES-W: (a) non-cognate pairs proposed by our system, (b) hand-selected representative errors.

(a) Orthographic Feature

Source Feat.	Closest Target Feats.	Example Translation
#st	#es, est	(statue, estatua)
ty#	ad#, d#	(felicity, felicidad)
ogy	gía, gí	(geology, geología)

(b) Context Feature

Source Feat.	Closest Context Features
party	partido, izquierda
democrat	socialistas, demócratas
beijing	pekín, kioto

Table 5: Hand selected examples of source and target features which are close in canonical space: (a) orthographic feature correspondences, (b) context features.

ever, we attempted to run an experiment as similar as possible in setup to Koehn and Knight (2002), using English Gigaword and German Europarl. In this setting, our MCCA system yielded 61.7% accuracy on the 186 most confident predictions compared to 39% reported in Koehn and Knight (2002).

7 Analysis

We have presented a novel generative model for bilingual lexicon induction and presented results under a variety of data conditions (section 6.1) and languages (section 6.3) showing that our system can produce accurate lexicons even in highly adverse conditions. In this section, we broadly characterize and analyze the behavior of our system.

We manually examined the top 100 errors in the English-Spanish lexicon produced by our system on EN-ES-W. Of the top 100 errors: 21 were correct translations not contained in the Wiktionary lexicon (e.g. pintura to painting), 4 were purely morphological errors (e.g. airport to aeropuertos), 30 were semantically related (e.g. basketball to béisbol), 15 were words with strong orthographic similarities (e.g. coast to costas), and 30 were difficult to categorize and fell into none of these categories. Since many of our 'errors' actually represent valid translation pairs not contained in our extracted dictionary, we supplemented our evaluation lexicon with one automatically derived from 100k sentences of parallel Europarl data. We ran the intersected HMM word-alignment model (Liang et al., 2008) and added (s, t) to the lexicon if s was aligned to t at least three times and more than any other word. Evaluating against the union of these lexicons yielded 98.0 $p_{0.33}$, a significant improvement over the 92.3 using only the Wiktionary lexicon. Of the true errors, the most common arose from semantically related words which had strong context feature correlations (see table 4(b)).

We also explored the relationships our model learns between features of different languages. We projected each source and target feature into the shared canonical space, and for each projected source feature we examined the closest projected target features. In table 5(a), we present some of the orthographic feature relationships learned by our system. Many of these relationships correspond to phonological and morphological regularities such as the English suffix ing mapping to the Spanish suffix gía. In table 5(b), we present context feature correspondences. Here, the broad trend is for words which are either translations or semantically related across languages to be close in canonical space.

8 Conclusion

We have presented a generative model for bilingual lexicon induction based on probabilistic CCA. Our experiments show that high-precision translations can be mined without any access to parallel corpora. It remains to be seen how such lexicons can be best utilized, but they invite new approaches to the statistical translation of resource-poor languages.

References

Francis R. Bach and Michael I. Jordan. 2006. A probabilistic interpretation of canonical correlation analysis. Technical report, University of California, Berkeley.

Peter F. Brown, Stephen Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1994. The mathematic of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

Mona Diab, Kadri Hacioglu, and Daniel Jurafsky. 2004. Automatic tagging of arabic text: From raw text to base phrase chunks. In *HLT-NAACL*.

Pascale Fung. 1995. Compiling bilingual lexicon entries from a non-parallel english-chinese corpus. In *Third Annual Workshop on Very Large Corpora*.

Michel Galley, Jonathan Graehl, Kevin Knight, Daniel Marcu, Steve DeNeefe, Wei Wang, and Ignacio Thayer. 2006. Scalable inference and training of context-rich syntactic translation models. In *COLING-ACL*.

David R. Hardoon, Sandor Szedmak, and John Shawe-Taylor. 2003. Canonical correlation analysis an overview with application to learning methods. Technical Report CSD-TR-03-02, Royal Holloway University of London.

Philipp Koehn and Kevin Knight. 2002. Learning a translation lexicon from monolingual corpora. In *Proceedings of ACL Workshop on Unsupervised Lexical Acquisition*.

P. Koehn. 2004. Pharaoh: A beam search decoder for phrase-based statistical machine translation models. In *Proceedings of AMTA 2004*.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *MT Summit*.

H. W. Kuhn. 1955. The Hungarian method for the assignment problem. *Naval Research Logistic Quarterly*.

P. Liang, D. Klein, and M. I. Jordan. 2008. Agreement-based learning. In *NIPS*.

Reinhard Rapp. 1995. Identifying word translation in non-parallel texts. In *ACL*.

Helmut Schmid. 1994. Probabilistic part-of-speech tagging using decision trees. In *International Conference on New Methods in Language Processing*.

N. Smith and J. Eisner. 2006. Annealing structural bias in multilingual weighted grammar induction. In *ACL*.

L. G. Valiant. 1979. The complexity of computing the permanent. *Theoretical Computer Science*, 8:189–201.

D. Yarowsky. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In *ACL*.