# Bilingual Word Spectral Clustering for Statistical Machine Translation

**Bing Zhao**[†] **Eric P. Xing**[† ‡] **Alex Waibel**[†]
[†]Language Technologies Institute
[‡]Center for Automated Learning and Discovery
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213
{bzhao,epxing,ahw}@cs.cmu.edu

## Abstract

In this paper, a variant of a spectral clustering algorithm is proposed for bilingual word clustering. The proposed algorithm generates the two sets of clusters for both languages efficiently with high semantic correlation within monolingual clusters, and high translation quality across the clusters between two languages. Each cluster level translation is considered as a bilingual concept, which generalizes words in bilingual clusters. This scheme improves the robustness for statistical machine translation models. Two HMM-based translation models are tested to use these bilingual clusters. Improved perplexity, word alignment accuracy, and translation quality are observed in our experiments.

## 1 Introduction

Statistical natural language processing usually suffers from the sparse data problem. Comparing to the available monolingual data, we have much less training data especially for statistical machine translation (SMT). For example, in language modelling, there are more than 1.7 billion words corpora available: English Gigaword by (Graff, 2003). However, for machine translation tasks, there are typically less than 10 million words of training data.

Bilingual word clustering is a process of forming corresponding word clusters suitable for machine translation. Previous work from (Wang et al., 1996) showed improvements in perplexity-oriented measures using mixture-based translation lexicon (Brown et al., 1993). A later study by (Och, 1999) showed improvements on perplexity of bilingual corpus, and word translation accuracy using a template-based translation model. Both approaches are optimizing the maximum likelihood of parallel corpus, in which a data point is a sentence pair: an English sentence and its translation in another language such as French. These algorithms are essentially the same as monolingual word clusterings (Kneser and Ney, 1993)—an iterative local search. In each iteration, a two-level loop over every possible word-cluster assignment is tested for better likelihood change. This kind of approach has two drawbacks: first it is easily to get stuck in local optima; second, the clustering of English and the other language are basically two separated optimization processes, and cluster-level translation is modelled loosely. These drawbacks make their approaches generally not very effective in improving translation models.

In this paper, we propose a variant of the spectral clustering algorithm (Ng et al., 2001) for bilingual word clustering. Given parallel corpus, first, the word's bilingual context is used directly as features - for instance, each English word is represented by its bilingual word translation candidates. Second, latent eigenstructure analysis is carried out in this bilingual feature space, which leads to clusters of words with similar translations. Essentially an affinity matrix is computed using these cross-lingual features. It is then decomposed into two sub-spaces, which are meaningful for translation tasks: the left subspace corresponds to the representation of words in English vocabulary, and the right sub-space corresponds to words in French. Each eigenvector is considered as one bilingual concept, and the bilingual clusters are considered to be its realizations in two languages. Finally, a general K-means cluster-



*Proceedings of the ACL Workshop on Building and Using Parallel Texts*, pages 25–32,
Ann Arbor, June 2005. 

ing algorithm is used to find out word clusters in the two sub-spaces.

The remainder of the paper is structured as follows: in section 2, concepts of translation models are introduced together with two extended HMMs; in section 3, our proposed bilingual word clustering algorithm is explained in detail, and the related works are analyzed; in section 4, evaluation metrics are defined and the experimental results are given; in section 5, the discussions and conclusions.

## 2 Statistical Machine Translation

The task of translation is to translate one sentence in some source language *F* into a target language *E*. For example, given a French sentence with $J$ words denoted as $f_1^J = f_1 f_2 ... f_J$, an SMT system automatically translates it into an English sentence with $I$ words denoted by $e_1^I = e_1 e_2 ... e_I$. The SMT system first proposes multiple English hypotheses in its model space. Among all the hypotheses, the system selects the one with the highest conditional probability according to Bayes's decision rule:

$$\hat{e}_1^I = \arg\max_{\{e_1^I\}} P(e_1^I|f_1^J) = \arg\max_{\{e_1^I\}} P(f_1^J|e_1^I)P(e_1^I), \quad (1)$$

where $P(f_1^J|e_1^I)$ is called *translation model*, and $P(e_1^I)$ is called *language model*. The translation model is the key component, which is the focus in this paper.

### 2.1 HMM-based Translation Model

HMM is one of the effective translation models (Vogel et al., 1996), which is easily scalable to very large training corpus.

To model word-to-word translation, we introduce the mapping $j \rightarrow a_j$, which assigns a French word $f_j$ in position $j$ to a English word $e_i$ in position $i = a_j$ denoted as $e_{a_j}$. Each French word $f_j$ is an observation, and it is generated by a HMM state defined as $[e_{a_j}, a_j]$, where the alignment $a_j$ for position $j$ is considered to have a dependency on the previous alignment $a_{j-1}$. Thus the first-order HMM is defined as follows:

$$P(f_1^J|e_1^I) = \sum_{a_1^J} \prod_{j=1}^{J} P(f_j|e_{a_j}) P(a_j|a_{j-1}), \quad (2)$$

where $P(a_j|a_{j-1})$ is the transition probability. This model captures the assumption that words close in the source sentence are aligned to words close in the target sentence. An additional pseudo word of "NULL" is used as the beginning of English sentence for HMM to start with. The (Och and Ney, 2003) model includes other refinements such as special treatment of a jump to a Null word, and a uniform smoothing prior. The HMM with these refinements is used as our baseline. Motivated by the work in both (Och and Ney, 2000) and (Toutanova et al., 2002), we propose the two following simplest versions of extended HMMs to utilize bilingual word clusters.

### 2.2 Extensions to HMM with word clusters

Let $\mathsf{F}$ denote the cluster mapping $f_j \rightarrow \mathsf{F}(f_j)$, which assigns French word $f_j$ to its cluster ID $F_j = \mathsf{F}(f_j)$. Similarly $\mathsf{E}$ maps English word $e_i$ to its cluster ID of $E_i = \mathsf{E}(e_i)$. In this paper, we assume each word belongs to one cluster only.

With bilingual word clusters, we can extend the HMM model in Eqn. 1 in the following two ways:

$$P(f_1^J|e_1^I) = \sum_{a_1^J} \prod_{j=1}^{J} P(f_j|e_{a_j}) \cdot P(a_j|a_{j-1}, \mathsf{E}(e_{a_{j-1}}), \mathsf{F}(f_{j-1})), \quad (3)$$

where $\mathsf{E}(e_{a_{j-1}})$ and $\mathsf{F}(f_{j-1})$ are non overlapping word clusters $(E_{a_{j-1}}, F_{j-1})$for English and French respectively.

Another explicit way of utilizing bilingual word clusters can be considered as a two-stream HMM as follows:

$$P(f_1^J, F_1^J|e_1^I, E_1^I) = \sum_{a_1^J} \prod_{j=1}^{J} P(f_j|e_{a_j}) P(F_j|E_{a_j}) P(a_j|a_{j-1}). \quad (4)$$

This model introduces the translation of bilingual word clusters directly as an extra factor to Eqn. 2. Intuitively, the role of this factor is to boost the translation probabilities for words sharing the same concept. This is a more expressive model because it models both word and the cluster level translation equivalence. Also, compared with the model in Eqn. 3, this model is easier to train, as it uses a two-dimension table instead of a four-dimension table.

However, we do not want this $P(F_j|E_{a_j})$ to dominate the HMM transition structure, and the obser-

vation probability of $P(f_j|e_{a_j})$ during the EM iterations. Thus a uniform prior $P(F_j) = 1/|F|$ is introduced as a smoothing factor for $P(F_j|E_{a_j})$:

$$P(F_j|E_{a_j}) = \lambda P(F_j|E_{a_j}) + (1-\lambda)P(F_j), \quad (5)$$

where $|F|$ is the total number of word clusters in French (we use the same number of clusters for both languages). $\lambda$ can be chosen to get optimal performance on a development set. In our case, we fix it to be 0.5 in all our experiments.

## 3 Bilingual Word Clustering

In bilingual word clustering, the task is to build word clusters $\mathsf{F}$ and $\mathsf{E}$ to form partitions of the vocabularies of the two languages respectively. The two partitions for the vocabularies of $\mathsf{F}$ and $\mathsf{E}$ are aimed to be suitable for machine translation in the sense that the cluster/partition level translation equivalence is reliable and focused to handle data sparseness; the translation model using these clusters explains the parallel corpus $\{(f_1^J, e_1^I)\}$ better in terms of perplexity or joint likelihood.

### 3.1 From Monolingual to Bilingual

To infer bilingual word clusters of $(\mathsf{F}, \mathsf{E})$, one can optimize the joint probability of the parallel corpus $\{(f_1^J, e_1^I)\}$ using the clusters as follows:

$$\begin{aligned}(\hat{\mathsf{F}}, \hat{\mathsf{E}}) &= \arg\max_{(F,E)} P(f_1^J, e_1^I|F, E) \\ &= \arg\max_{(F,E)} P(e_1^I|E)P(f_1^J|e_1^I, F, E), \quad (6)\end{aligned}$$

Eqn. 6 separates the optimization process into two parts: the monolingual part for $\mathsf{E}$, and the bilingual part for $\mathsf{F}$ given fixed $\mathsf{E}$. The monolingual part is considered as a prior probability: $P(e_1^I|\mathsf{E})$, and $\mathsf{E}$ can be inferred using corpus bigram statistics in the following equation:

$$\begin{aligned}\hat{\mathsf{E}} &= \arg\max_{\{E\}} P(e_1^I|E) \\ &= \arg\max_{\{E\}} \prod_{i=1}^{I} P(E_i|E_{i-1})P(e_i|E_i). \quad (7)\end{aligned}$$

We need to fix the number of clusters beforehand, otherwise the optimum is reached when each word is a class of its own. There exists efficient leave-one-out style algorithm (Kneser and Ney, 1993), which can automatically determine the number of clusters.

For the bilingual part $P(f_1^J|e_1^I, F, E)$, we can slightly modify the same algorithm as in (Kneser and Ney, 1993). Given the word alignment $\{a_1^J\}$ between $f_1^J$ and $e_1^I$ collected from the Viterbi path in HMM-based translation model, we can infer $\hat{\mathsf{F}}$ as follows:

$$\begin{aligned}\hat{\mathsf{F}} &= \arg\max_{\{F\}} P(f_1^J|e_1^I, F, E) \\ &= \arg\max_{\{F\}} \prod_{j=1}^{J} P(F_j|E_{a_j})P(f_j|F_j). \quad (8)\end{aligned}$$

Overall, this bilingual word clustering algorithm is essentially a two-step approach. In the first step, $\mathsf{E}$ is inferred by optimizing the monolingual likelihood of English data, and secondly $\mathsf{F}$ is inferred by optimizing the bilingual part without changing $\mathsf{E}$. In this way, the algorithm is easy to implement without much change from the monolingual correspondent.

This approach was shown to give the best results in (Och, 1999). We use it as our baseline to compare with.

### 3.2 Bilingual Word Spectral Clustering

Instead of using word alignment to bridge the parallel sentence pair, and optimize the likelihood in two separate steps, we develop an alignment-free algorithm using a variant of spectral clustering algorithm. The goal is to build high cluster-level translation quality suitable for translation modelling, and at the same time maintain high intra-cluster similarity , and low inter-cluster similarity for monolingual clusters.

#### 3.2.1 Notations

We define the vocabulary $V_F$ as the French vocabulary with a size of $|V_F|$; $V_E$ as the English vocabulary with size of $|V_E|$. A co-occurrence matrix $C_{\{F,E\}}$ is built with $|V_F|$ rows and $|V_E|$ columns; each element represents the co-occurrence counts of the corresponding French word $f_j$ and English word $e_i$. In this way, each French word forms a row vector with a dimension of $|V_E|$, and each dimensionality is a co-occurring English word. The elements in the vector are the co-occurrence counts. We can also

view each column as a vector for English word, and we'll have similar interpretations as above.

### 3.2.2 Algorithm

With $C_{\{F,E\}}$, we can infer two affinity matrixes as follows:

$$A_E = C_{\{F,E\}}^T C_{\{F,E\}}$$
$$A_F = C_{\{F,E\}} C_{\{F,E\}}^T,$$

where $A_E$ is an $|V_E| \times |V_E|$ affinity matrix for English words, with rows and columns representing English words and each element the inner product between two English words column vectors. Correspondingly, $A_F$ is an affinity matrix of size $|V_F| \times |V_F|$ for French words with similar definitions. Both $A_E$ and $A_F$ are *symmetric* and *non-negative*. Now we can compute the eigenstructure for both $A_E$ and $A_F$. In fact, the eigen vectors of the two are correspondingly the right and left sub-spaces of the original co-occurrence matrix of $C_{\{F,E\}}$ respectively. This can be computed using singular value decomposition (SVD): $C_{\{F,E\}} = USV^T$, $A_E = VS^2V^T$, and $A_F = US^2U^T$, where $U$ is the left sub-space, and $V$ the right sub-space of the co-occurrence matrix $C_{\{F,E\}}$. $S$ is a diagonal matrix, with the singular values ranked from large to small along the diagonal. Obviously, the left sub-space $U$ is the eigenstructure for $A_F$; the right sub-space $V$ is the eigenstructure for $A_E$.

By choosing the top $K$ singular values (the square root of the eigen values for both $A_E$ and $A_F$), the sub-spaces will be reduced to: $U_{|V_F| \times K}$ and $V_{|V_E| \times K}$ respectively. Based on these subspaces, we can carry out K-means or other clustering algorithms to infer word clusters for both languages. Our algorithm goes as follows:

- Initialize bilingual co-occurrence matrix $C_{\{F,E\}}$ with rows representing French words, and columns English words. $C_{ji}$ is the co-occurrence raw counts of French word $f_j$ and English word $e_i$;
- Form the affinity matrix $A_E = C_{\{F,E\}}^T C_{\{F,E\}}$ and $A_F = C_{\{F,E\}}^T C_{\{F,E\}}$. Kernels can also be applied here such as $A_E = \exp(\frac{C_{\{F,E\}} C_{\{F,E\}}^T}{\sigma^2})$ for English words. Set $A_{Eii} = 0$ and $A_{Fii} = 0$, and normalize each row to be unit length;
- Compute the eigen structure of the normalized matrix $A_E$, and find the $k$ largest eigen vectors: $v_1, v_2, ..., v_k$; Similarly, find the $k$ largest eigen vectors of $A_F$: $u_1, u_2, ..., u_k$;
- Stack the $k$ eigenvectors of $v_1, v_2, ..., v_k$ in the columns of $Y_E$, and stack the eigenvectors $u_1, u_2, ..., u_k$ in the columns for $Y_F$; Normalize rows of both $Y_E$ and $Y_F$ to have unit length. $Y_E$ is size of $|V_E| \times k$ and $Y_F$ is size of $|V_F| \times k$;
- Treat each row of $Y_E$ as a point in $R^{|V_E| \times k}$, and cluster them into $K$ English word clusters using K-means. Treat each row of $Y_F$ as a point in $R^{|V_F| \times k}$, and cluster them into $K$ French word clusters.
- Finally, assign original word $e_i$ to cluster $E_k$ if row i of the matrix $Y_E$ is clustered as $E_k$; similar assignments are for French words.

Here $A_E$ and $A_F$ are affinity matrixes of pair-wise inner products between the monolingual words. The more similar the two words, the larger the value. In our implementations, we did not apply a kernel function like the algorithm in (Ng et al., 2001). But the kernel function such as the exponential function mentioned above can be applied here to control how rapidly the similarity falls, using some carefully chosen scaling parameter.

### 3.2.3 Related Clustering Algorithms

The above algorithm is very close to the variants of a big family of the spectral clustering algorithms introduced in (Meila and Shi, 2000) and studied in (Ng et al., 2001). Spectral clustering refers to a class of techniques which rely on the eigenstructure of a similarity matrix to partition points into disjoint clusters with high intra-cluster similarity and low inter-cluster similarity. It's shown to be computing the $k$-way normalized cut: $K - trY^T D^{-\frac{1}{2}} A D^{-\frac{1}{2}} Y$ for any matrix $Y \in R^{M \times N}$. $A$ is the affinity matrix, and $Y$ in our algorithm corresponds to the subspaces of $U$ and $V$.

Experimentally, it has been observed that using more eigenvectors and directly computing a $k$-way partitioning usually gives better performance. In our implementations, we used the top 500 eigen vectors to construct the subspaces of $U$ and $V$ for K-means clustering.

### 3.2.4 K-means

The K-means here can be considered as a post-processing step in our proposed bilingual word clustering. For initial centroids, we first compute the *center* of the whole data set. The farthest centroid from the center is then chosen to be the first initial centroid; and after that, the other K-1 centroids are chosen one by one to well separate all the previous chosen centroids.

The stopping criterion is: if the maximal change of the clusters' centroids is less than the threshold of 1e-3 between two iterations, the clustering algorithm then stops.

## 4 Experiments

To test our algorithm, we applied it to the TIDES Chinese-English small data track evaluation test set. After preprocessing, such as English tokenization, Chinese word segmentation, and parallel sentence splitting, there are in total 4172 parallel sentence pairs for training. We manually labeled word alignments for 627 test sentence pairs randomly sampled from the dry-run test data in 2001, which has four human translations for each Chinese sentence. The preprocessing for the test data is different from the above, as it is designed for humans to label word alignments correctly by removing ambiguities from tokenization and word segmentation as much as possible. The data statistics are shown in Table 1.

| | | English | Chinese |
|---|---|---|---|
| Train | Sent. Pairs | 4172 | |
| | Words | 133598 | 105331 |
| | Voc Size | 8359 | 7984 |
| Test | Sent. Pairs | 627 | |
| | Words | 25500 | 19726 |
| | Voc Size | 4084 | 4827 |
| | Unseen Voc Size | 1278 | 1888 |
| | Alignment Links | 14769 | |

Table 1: Training and Test data statistics

### 4.1 Building Co-occurrence Matrix

Bilingual word co-occurrence counts are collected from the training data for constructing the matrix of $C_{\{F,E\}}$. Raw counts are collected without word alignment between the parallel sentences. Practically, we can use word alignment as used in (Och, 1999). Given an initial word alignment inferred by HMM, the counts are collected from the aligned word pair. If the counts are L-1 normalized, then the co-occurrence matrix is essentially the bilingual word-to-word translation lexicon such as $P(f_j|e_{a_j})$. We can remove very small entries ($P(f|e) \leq 1e^{-7}$), so that the matrix of $C_{\{F,E\}}$ is more sparse for eigen-structure computation. The proposed algorithm is then carried out to generate the bilingual word clusters for both English and Chinese.

Figure 1 shows the ranked Eigen values for the co-occurrence matrix of $C_{\{F,E\}}$.

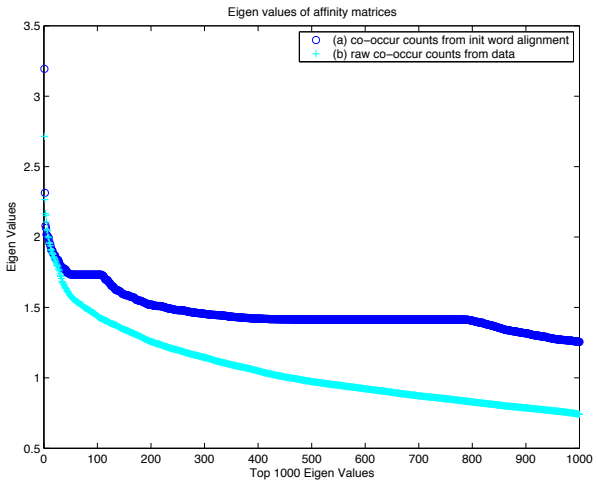


Figure 1: Top-1000 Eigen Values of Co-occurrence Matrix

It is clear, that using the initial HMM word alignment for co-occurrence matrix makes a difference. The top Eigen value using word alignment in plot $a$. (the deep blue curve) is 3.1946. The two plateaus indicate how many top $K$ eigen vectors to choose to reduce the feature space. The first one indicates that K is in the range of 50 to 120, and the second plateau indicates K is in the range of 500 to 800. Plot $b$. is inferred from the raw co-occurrence counts with the top eigen value of 2.7148. There is no clear plateau, which indicates that the feature space is less structured than the one built with initial word alignment.

We find 500 top eigen vectors are good enough for bilingual clustering in terms of efficiency and effectiveness.

### 4.2 Clustering Results

Clusters built via the two described methods are compared. The first method *bil1* is the two-step optimization approach: first optimizing the monolingual clusters for target language (English), and afterwards optimizing clusters for the source language (Chinese). The second method *bil2* is our proposed algorithm to compute the eigenstructure of the co-occurrence matrix, which builds the left and right subspaces, and finds clusters in such spaces. Top 500 eigen vectors are used to construct these subspaces. For both methods, 1000 clusters are inferred for English and Chinese respectively. The number of clusters is chosen in a way that the final word alignment accuracy was optimal. Table 2 provides the clustering examples using the two algorithms.

| settings | cluster examples |
|---|---|
| mono-$E_1$ | entirely,mainly,merely |
| mono-$E_2$ | 10th,13th,14th,16th,17th,18th,19th 20th,21st,23rd,24th,26th |
| mono-$E_3$ | drink,anglophobia,carota,giant,gymnasium |
| bil1-$C_3$ | 冲, 淡, 呼, 画, 啤酒, 热带, 水 |
| bil2-$E_1$ | alcoholic cognac distilled drink scotch spirits whiskey |
| bil2-$C_1$ | 白酒, 酒, 盲, 幕后, 涅, 日耳曼, 三星, 适, 苏格兰, 童, 威士忌, 蒸馏 |
| bil2-$E_2$ | evrec harmony luxury people sedan sedans tour tourism tourist toward travel |
| bil2-$C_2$ | 产业经济, 导游, 贯彻, 疾驶, 家境, 轿车, 旅行, 旅游, 人, 人民, 世人 |

Table 2: Bilingual Cluster Examples

The monolingual word clusters often contain words with similar syntax functions. This happens with esp. frequent words (eg. mono-$E_1$ and mono-$E_2$). The algorithm tends to put rare words such as "carota, anglophobia" into a very big cluster (eg. mono-$E_3$). In addition, the words within these monolingual clusters rarely share similar translations such as the typical cluster of "week, month, year". This indicates that the corresponding Chinese clusters inferred by optimizing Eqn. 7 are not close in terms of translational similarity. Overall, the method of bil1 does not give us a good translational correspondence between clusters of two languages. The English cluster of mono-$E_3$ and its best aligned candidate of bil1-$C_3$ are not well correlated either.

Our proposed bilingual cluster algorithm bil2 generates the clusters with stronger semantic meaning within a cluster. The cluster of bil2-$E_1$ relates to the concept of "wine" in English. The monolingual word clustering tends to scatter those words into several big noisy clusters. This cluster also has a good translational correspondent in bil2-$C_1$ in Chinese. The clusters of bil2-$E_2$ and bil2-$C_2$ are also correlated very well. We noticed that the Chinese clusters are slightly more noisy than their English corresponding ones. This comes from the noise in the parallel corpus, and sometimes from ambiguities of the word segmentation in the preprocessing steps.

To measure the quality of the bilingual clusters, we can use the following two kind of metrics:

- Average $\epsilon$-mirror (Wang et al., 1996): The $\epsilon$-mirror of a class $E_i$ is the set of clusters in Chinese which have a translation probability greater than $\epsilon$. In our case, $\epsilon$ is 0.05, the same value used in (Och, 1999).

- Perplexity: The perplexity is defined as proportional to the negative log likelihood of the HMM model Viterbi alignment path for each sentence pair. We use the bilingual word clusters in two extended HMM models, and measure the perplexities of the unseen test data after seven forward-backward training iterations. The two perplexities are defined as $PP1 = \exp(-\sum_{j=1}^{J} \log(P(f_j|e_{a_j})P(a_j|a_{j-1}, E_{a_{j-1}}, F_{j-1}))/J)$ and $PP2 = \exp(-J^{-1}\sum_{j=1}^{J}\log(P(f_j|e_{a_j})P(a_j|a_{j-1})P(F_{j-1}|E_{a_{j-1}})))$ for the two extended HMM models in Eqn 3 and 4.

Both metrics measure the extent to which the translation probability is spread out. The smaller the better. The following table summarizes the results on $\epsilon$-mirror and perplexity using different methods on the unseen test data.

| algorithms | $\epsilon$-mirror | HMM-1 Perp | HMM-2 Perp |
|---|---|---|---|
| baseline | - | 1717.82 | |
| bil1 | 3.97 | 1810.55 | 352.28 |
| bil2 | 2.54 | 1610.86 | 343.64 |

The baseline uses no word clusters. bil1 and bil2 are defined as above. It is clear that our proposed method gives overall lower perplexity: 1611 from the baseline of 1717 using the extended HMM-1. If we use HMM-2, the perplexity goes down even more using bilingual clusters: 352.28 using bil1, and 343.64 using bil2. As stated, the four-dimensional

table of $P(a_j|a_{j-1}, E(e_{a_{j-1}}), F(f_{j-1}))$ is easily subject to overfitting, and usually gives worse perplexities.

Average $\epsilon$-mirror for the two-step bilingual clustering algorithm is 3.97, and for spectral clustering algorithm is 2.54. This means our proposed algorithm generates more focused clusters of translational equivalence. Figure 2 shows the histogram for the cluster pairs $(F_j, E_i)$, of which the cluster level translation probabilities $P(F_j|E_i) \in [0.05, 1]$. The interval $[0.05, 1]$ is divided into 10 bins, with first bin $[0.05, 0.1]$, and 9 bins divides$[0.1, 1]$ equally. The percentage for clusters pairs with $P(F_j|E_i)$ falling in each bin is drawn.

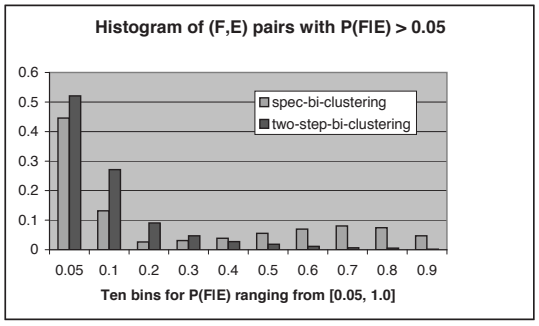


Figure 2: Histogram of cluster pairs $(F_j, E_i)$

Our algorithm generates much better aligned cluster pairs than the two-step optimization algorithm. There are 120 cluster pairs aligned with $P(F_j|E_i) \geq 0.9$ using clusters from our algorithm, while there are only 8 such cluster pairs using the two-step approach. Figure 3 compares the $\epsilon$-mirror at different numbers of clusters using the two approaches. Our algorithm has a much better $\epsilon$-mirror than the two-step approach over different number of clusters.

Overall, the extended HMM-2 is better than HMM-1 in terms of perplexity, and is easier to train.

### 4.3 Applications in Word Alignment

We also applied our bilingual word clustering in a word alignment setting. The training data is the TIDES small data track. The word alignments are manually labeled for 627 sentences sampled from the dryrun test data in 2001. In this manually aligned data, we include one-to-one, one-to-many, and many-to-many word alignments. Figure 4 summarizes the word alignment accuracy for different

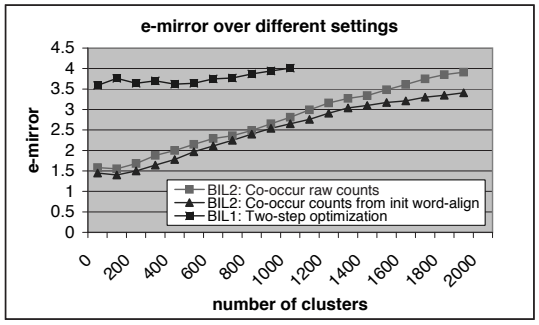


Figure 3: $\epsilon$-mirror with different settings

methods. The baseline is the standard HMM translation model defined in Eqn. 2; the HMM1 is defined in Eqn 3, and HMM2 is defined in Eqn 4. The algorithm is applying our proposed bilingual word clustering algorithm to infer 1000 clusters for both languages. As expected, Figure 4 shows that using

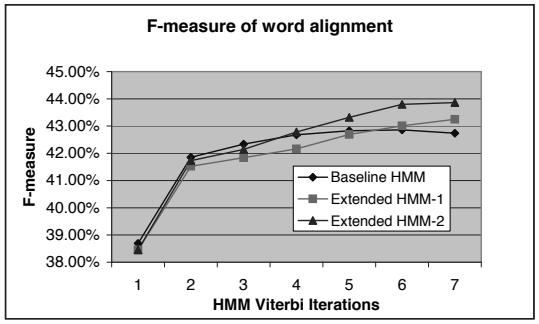


Figure 4: Word Alignment Over Iterations

word clusters is helpful for word alignment. HMM2 gives the best performance in terms of F-measure of word alignment. One quarter of the words in the test vocabulary are unseen as shown in Table 1. These unseen words related alignment links (4778 out of 14769) will be left unaligned by translation models. Thus the oracle (best possible) recall we could get is 67.65%. Our standard t-test showed that significant interval is 0.82% at the 95% confidence level. The improvement at the last iteration of HMM is marginally significant.

### 4.4 Applications in Phrase-based Translations

Our pilot word alignment on unseen data showed improvements. However, we find it more effective in our phrase extraction, in which three key scores

are computed: phrase level fertilities, distortions, and lexicon scores. These scores are used in a local greedy search to extract phrase pairs (Zhao and Vogel, 2005). This phrase extraction is more sensitive to the differences in $P(f_j|e_i)$ than the HMM Viterbi word aligner.

The evaluation conditions are defined in NIST 2003 Small track. Around 247K test set (919 Chinese sentences) specific phrase pairs are extracted with up to 7-gram in source phrase. A trigram language model is trained using Gigaword XinHua news part. With a monotone phrase-based decoder, the translation results are reported in Table 3. The

| Eval. | Baseline | Bil1 | Bil2 |
|---|---|---|---|
| NIST | 6.417 | 6.507 | 6.582 |
| BLEU | 0.1558 | 0.1575 | 0.1644 |

Table 3: NIST'03 C-E Small Data Track Evaluation

baseline is using the lexicon $P(f_j|e_i)$ trained from standard HMM in Eqn. 2, which gives a BLEU score of 0.1558 +/- 0.0113. Bil1 and Bil2 are using $P(f_j|e_i)$ from HMM in Eqn. 4 with 1000 bilingual word clusters inferred from the two-step algorithm and the proposed one respectively. Using the clusters from the two-step algorithm gives a BLEU score of 0.1575, which is close to the baseline. Using clusters from our algorithm, we observe more improvements with BLEU score of 0.1644 and a NIST score of 6.582.

## 5 Discussions and Conclusions

In this paper, a new approach for bilingual word clustering using eigenstructure in bilingual feature space is proposed. Eigenvectors from this feature space are considered as bilingual concepts. Bilingual clusters from the subspaces expanded by these concepts are inferred with high semantic correlations within each cluster, and high translation qualities across clusters from the two languages.

Our empirical study also showed effectiveness of using bilingual word clusters in extended HMMs for statistical machine translation. The K-means based clustering algorithm can be easily extended to do hierarchical clustering. However, extensions of translation models are needed to leverage the hierarchical clusters appropriately.

## References


P.F. Brown, Stephen A. Della Pietra, Vincent. J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. In *Computational Linguistics*, volume 19(2), pages 263–331.

David Graff. 2003. Ldc gigaword corpora: English gigaword (ldc catalog no: Ldc2003t05). In *LDC link: http://www.ldc.upenn.edu/Catalog/index.jsp*.

R. Kneser and Hermann Ney. 1993. Improved clustering techniques for class-based statistical language modelling. In *European Conference on Speech Communication and Technology*, pages 973–976.

Marina Meila and Jianbo Shi. 2000. Learning segmentation by random walks. In *Advances in Neural Information Processing Systems. (NIPS2000)*, pages 873–879.

A. Ng, M. Jordan, and Y. Weiss. 2001. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems 14: Proceedings of the 2001*.

Franz J. Och and Hermann Ney. 2000. A comparison of alignment models for statistical machine translation. In *COLING'00: The 18th Int. Conf. on Computational Linguistics*, pages 1086–1090, Saarbrucken, Germany, July.

Franz J. Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. In *Computational Linguistics*, volume 29, pages 19–51.

Franz J. Och. 1999. An efficient method for determining bilingal word classes. In *Ninth Conf. of the Europ. Chapter of the Association for Computational Linguistics (EACL'99)*, pages 71–76.

Kristina Toutanova, H. Tolga Ilhan, and Christopher D. Manning. 2002. Extensions to hmm-based statistical word alignment models. In *Proc. of the Conference on Empirical Methods in Natural Language Processing*.

S. Vogel, Hermann Ney, and C. Tillmann. 1996. Hmm based word alignment in statistical machine translation. In *Proc. The 16th Int. Conf. on Computational Lingustics, (Coling'96)*, pages 836–841.

Yeyi Wang, John Lafferty, and Alex Waibel. 1996. Word clustering with parallel spoken language corpora. In *proceedings of the 4th International Conference on Spoken Language Processing (ICSLP'96)*, pages 2364–2367.

Bing Zhao and Stephan Vogel. 2005. A generalized alignment-free phrase extraction algorithm. In *ACL 2005 Workshop: Building and Using Parallel Corpora: Data-driven Machine Translation and Beyond*, Ann Arbor, Michigan.