InProceedings of the ICSLP'96, Philadelphia, USA, October 1996.

WORD CLUSTERING WITH PARALLEL SPOKEN LANGUAGE CORPORA*

Ye-Yi Wang — yyw@cs.cmu.edu
John Lafferty — lafferty@cs.cmu.edu
Alex Waibel — waibel@cs.cmu.edu

Carnegie Mellon University

ABSTRACT

In this paper we introduce a word clustering algorithm which uses a bilingual, parallel corpus to group together words in the source and target language. Our method generalizes previous mutual information clustering algorithms for monolingual data by incorporating a statistical translation model. Preliminary experiments have shown that the algorithm can effectively employ the constraints implicit in bilingual data to extract classes which are well-suited to machine translation tasks.

1. INTRODUCTION

Language learning is a multi-modal process. Children can never learn a language by only reading a book, without any other input. Language learning is also a multi-channel process. A second language learner often uses the knowledge about his native language and the correspondence between the native language and the second language in acquiring new language ability. In this paper we investigate a method to automatically classify words by employing a parallel, bilingual corpus of text.

Word clustering and class-based language modeling provide an efficient way to reduce the number of parameters and subdue the sparse data problem. Various clustering techniques [2, 3, 4, 5] have been reported recently which use a corpus of text in a single language. This monolingual approach can sometimes lead to peculiar results since the clustering decisions are typically based on local contexts. For example, when we applied the approach of [2] to scheduling data, we found many cases like the class {`couple few lot message`}, in which the word `message` is out of place. This is due to the fact that the clustering technique is based on word bigrams, and each word in this class typically follows the word `a` and precedes the word `of` or `to` in the training corpus.

Compared with a clustering algorithm based on a single language, a clustering algorithm taking constraints from parallel corpora potentially has several attractive advantages. First, training samples in another language provide indirect evidence for a classification.

Second, constraints from both languages may help to "wash out" some biased language-specific usages, resulting in classes of better quality. In addition, the resulting classes may be better suited for statistical machine translation, which is the primary motivation for this work. Of course, there are potential disadvantages to bilingual clustering as well. For one, there will never be nearly as much parallel bilingual data as monolingual data available. A key problem, therefore, is to combine data from many sources to obtain better clustering procedures.

2. CLUSTERING WITH PARALLEL CORPORA

Several classification schemes [2, 3, 5] are based on the maximum likelihood principle, and seek to find a classification C such that $P(W \mid C)$, the class-based likelihood of W, is maximized. It was shown in [2] that maximizing the log-likelihood of a corpus with a class-based bigram is equivalent to maximizing the average mutual information $I(C_1, C_2)$ between adjacent classes in text:

$$\frac{1}{n-1} \log P(W \mid C) \approx -H(W) + I(C_1, C_2) \qquad (1)$$

where $H(W)$ is the entropy of the English corpus, which is independent of the clustering. A greedy algorithm was then introduced to find classes that maximize the average mutual information. Initially each word is assigned to a distinct class and the average mutual information between adjacent classes is computed. At each step in the algorithm, the loss in average mutual information that would result from merging each candidate pair of classes is computed, and the merge is then carried out for that pair which affects the smallest loss.

The bilingual clustering algorithm described here is based on this mutual information clustering technique. To employ

*This research was partly supported by ATR. The views and conclusions in this document are those of the authors.

the constraints from a parallel corpus, we use an alignment between pairs of sentences [1] as a "bridge" between the languages. To be concrete, suppose we have an English corpus E and its parallel German corpus G, and we want to cluster the English words appearing in E. Instead of maximizing the log-likelihood $\log P(E \mid C)$, we instead seek to maximize the joint log-likelihood of the parallel corpus:

$$\begin{aligned} & \frac{1}{n-1} \log P(E, G \mid C) \\ & = \frac{1}{n-1} (\log P(E \mid C) + \log P(G \mid E, C)) \\ & \approx -H(E) + I(C_1, C_2) + \frac{1}{n-1} \log P(G \mid E, C) \end{aligned} \quad (2)$$

where

$$P(G \mid E, C) = \sum_{i}^{L} \sum_{A} P(G_i A \mid E_i, C) \quad (3)$$

Here E_i and G_i are the ith pair of utterances in the parallel corpus, L is the number of sentences in the corpus, and A is an alignment between E_i and G_i.

We can initially assign each word to a separate class, and incrementally merge classes using a greedy search algorithm. At the k-th step in the algorithm, the decrease in likelihood (2) resulting from a merge of classes c_1 and c_2 can be expressed as a sum of two terms: $L_k(c_1, c_2)$, the loss of average mutual information between adjacent classes, and $D_k(c_1, c_2)$, the change in the likelihood of the German corpus when c_1 and c_2 are merged. With clever bookkeeping, one can efficiently find the smallest $L_k(c_1, c_2)$ in time $O(V^2)$, where V is the lexicon size [2]. In the following section we describe a method to efficiently calculate $D_k(c_1, c_2)$ using a class-based translation model.

3. IMPLEMENTATION AND COMPLEXITY

To model the change in likelihood of the German corpus, we employ a slight modification of "Model 1" used in the IBM statistical machine translation system. This model probabilistically generates the German corpus from the English corpus using a simple alignment between pairs of words:

$$P(G_i, A \mid E_i) = \frac{\epsilon}{(|E_i| + 1)^{|G_i|}} \prod_{j=1}^{|G_i|} t(g_j \mid e_{a_j}). \quad (4)$$

Equation (4) can be interpreted by imagining that the German sentence G_i has a fixed probability for its length $|G_i|$, and a position j in G_i is aligned to any position in its English translation E_i with equal likelihood $(|E_i| + 1)^{-|G_i|}$. The German word at position j, g_j, is generated from the English word e_{a_j} at its aligned position with the *translation probability* $t(g_j \mid e_{a_j})$. The EM algorithm can be used to estimate the parameters t in this alignment model.

By "tying" the translations probabilities so that $t(g_j \mid e_{a_j}) = t(g_j \mid c_{a_j})$, where c_i is the class of English word e_i, the model can be expressed as

$$\begin{aligned} & P(G_i \mid E_i, C) \\ & = \sum_{A} P(G_i, A \mid E_i, C) \\ & = \frac{\epsilon}{(|E_i| + 1)^{|G_i|}} \sum_{a_1=0}^{|E_i|} \cdots \sum_{a_{|G_i|}=0}^{|E_i|} \prod_{j=1}^{|G_i|} t(g_j \mid c_{a_j}) \\ & = \frac{\epsilon}{(|E_i| + 1)^{|G_i|}} \prod_{j=1}^{|G_i|} \sum_{k=0}^{|E_i|} t(g_j \mid c_k) . \end{aligned} \quad (5)$$

Therefore,

$$\begin{aligned} & D_k(c_1, c_2) \\ & = \sum_{i}^{L} \log P(G_i \mid E_i, C(c_1 + c_2)) - \sum_{i} \log P(G_i \mid E_i, C) \\ & = \sum_{i}^{L} \sum_{j=1}^{|G_i|} \log \left(\sum_{k=0}^{|E_i|} t'(g_i \mid c'_{e_k}) \Big/ \sum_{k=0}^{|E_i|} t(g_i \mid c_{e_k}) \right) \end{aligned} \quad (6)$$

where C is the classification before the merge of c_1 and c_2, $C(c_1 + c_2)$ is the classification after the merge, c_e is the class of e in C, c'_e is the class of e in $C(c_1 + c_2)$, and t' is the new translation probability after the merge of c_1 and c_2.

Although (6) provides a way to calculate the likelihood change of the second language corpus, it is not practical for implementation. To estimate the likelihood change of the German corpus after a merge, we would in principle need to know the new parameters t'. Since these are determined by EM training, and since all of the parameters could be affected by a single merge, the bookkeeping method that works for monolingual clustering is not applicable in the bilingual case.

To reduce the computational demands, we have made the following approximating assumptions:

1. The merge of classes c_1 and c_2 will not affect the translation probabilities for classes other than c_1 and c_2. That is, $t(g \mid c)$ will remain unchanged, for $c \neq c_1, c_2$. For the merged class $c1 + c2$, the translation probability can be estimated without re-training:

$$\begin{aligned} & t(g \mid c1 + c2) \\ & \approx t(g \mid c1) P(c1 \mid c1 + c2) + t(g \mid c2) P(c2 \mid c1 + c2) \\ & = \frac{t(g \mid c1) P(c1) + t(g \mid c2) P(c2)}{P(c1) + P(c2)} \end{aligned} \quad (7)$$

2. The translation probabilities will not change significantly for at least M merges.
3. The best potential merge pair c_1, c_2 is within the top N merge candidates with lowest $L_k(c_1, c_2)$ discovered by the monolingual clustering technique.

With approximation (1), we do not need to retrain the parameters for each potential merge. Similarly, with approximation (2), we can avoid reestimating the parameters after each merge that is actually carried out. With approximation (3), we need to calculate $D_k(c_1, c_2)$ for only N pairs. Figure 1 illustrates the average percentage of agreement between the Viterbi alignments of the parallel corpus with the approximated parameters and with the reestimated parameters, as a function of the number of merging steps. It shows that approximations (1) and (2) are reasonable up to $M = 5$.

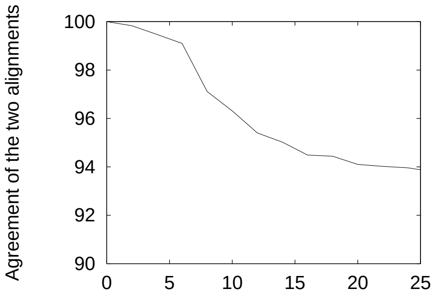

Figure 1: The average agreement of the Viterbi alignments of the parallel corpus with the approximated parameters and the re-trained parameters.

With these simplifying assumptions, we obtain the following algorithm:

Algorithm 3.1 *(Bilingual Clustering)*

1. *Initialization: assign a distinct class to each word e. Compute $L_V(c_1, c_2)$ and the other variables used in monolingual clustering for all pairs of English classes c_1, c_2.*
2. *Alignment: Train the parameters $t(g\,|\,c)$ of the class-based translation model using the EM algorithm.*
3. *Repeat the following:*
   (a) *With the monolingual clustering technique, find the N pairs c_1, c_2 having the smallest $L_k(c_1, c_2)$.*
   (b) *For each pair c_1, c_2 of the N merge candidates, compute $D_k(c_1, c_2)$. Re-score the pair c_1, c_2 with $L_k(c_1, c_2) + D_k(c_1, c_2)/(n-1)$, where n is the number of words in the English corpus.*
   (c) *Merge the pair c_1, c_2 having the lowest score.*
   (d) *Increase* **no-reestimation-count** *by 1.*
   (e) *If* **no-reestimation-count**$> M$*, reestimate the translation probabilities according to the EM algorithm, and set* **no-reestimation-count** *to 0.*

If we were to use a more complicated translation model, the above algorithm could be efficiently adapted by only collecting counts and evaluating changes in likelihood by summing over a small neighborhood of the Viterbi alignment, and by assuming that this alignment is fixed for M iterations of the algorithm. When there is a large monolingual corpus available in addition to the parallel corpus, we can use the monolingual corpus to select a pool of merge candidates, and then use the bilingual constraints to select the best pair.

4. TWO LANGUAGES ARE MORE INFORMATIVE THAN ONE

We carried out some experiments with the bilingual clustering algorithm presented above. The corpus used was the English/German scheduling data for the Janus project [6], containing about 1500 parallel utterances (39K English words and 41K German words), with a lexicon size of around 1,300 words for English and 1,800 for German. The words that occur fewer than 5 times in the corpus did not participate in the mutual information clustering procedure; they were assigned to a class according to simple heuristics. When no heuristic applied, they were assigned to a separate class. As an example of the heuristics, we put every low frequency word that is an element of a name list into one class. Other heuristics are mostly morphological, such as grouping all low frequency English words ending with `-ble` together.

The perplexity of the class-based bigram models trained with classes discovered using the parallel corpus is slightly lower than that for the language model with classes found with a single language corpus (35.2 vs. 36.9 for English). While this improvement is not significant, it appears that the new clustering algorithm finds classes of higher quality. Table 1 and Table 2 list some of the classes discovered by the monolingual and bilingual algorithms.

```
say +re
are unless days times
fact May January November July having department
case Wean
after around before between
or +ah+ afternoons
out fine
free clear available open
and however otherwise idea Patty through
day weekend right Mark
good perfect space nice great better away
pretty completely totally real
half m date conference cream bit
what afterwards why
couple few lot message
```

Table 1: Example Word Classes Discovered with Monolingual Mutual Information Clustering

```
are +re
January May November July fact
one noon
it early
or through
after before between
hours weeks days times
all
still had certainly may completely totally
well yeah unfortunately John Patty Mark
fine great better perfect nice
what when where
third sixteenth eleventh lounge thirtieth fifteenth
couple little bit lot half
```

Table 2: Example Word Classes Discovered with Bilingual Mutual Information Clustering

In Table 1, the "month name" class {`fact May January November July having department case Wean`} is mixed with what might be considered "noise" words, which appear because of various biased, language-specific usages of words. This is much improved in Table 2. The same effect occurs in many other classes.

How does bilingual clustering achieve this improvement? This can be explained as follows. The alignment model will assign some probability mass not only to the correct translations of the classes, but also to words that appear frequently in the same sentences with the correct translations. This spreading of the probability is less harmful if the classes contain semantically similar words. Since semantically similar words usually appear in similar contexts (in this case, sentences), although the class-based probability may reduce the probability of the correct translation of a word, it may raise the probability of other words in the context of the correct translation. This affect is minimized when words are clustered in a semantically similar manner. If a class contains words of distinct meanings, because those words generally occur in different contexts, the translation probabilities can become much more spread out over the different contexts, hence the overall sentence translation probability will be reduced significantly.

To be more precise, we define the ϵ-*mirror* of an input language class C_i as the set of all possible translations of C_i in another language having translation probability greater than ϵ:

$$C_i^\epsilon = \{s : P(s \mid C_i) > \epsilon\} \tag{8}$$

The average size of an ϵ-mirror is an indication of the extent to which the translation probability is spread out. With $\epsilon = 0.05$, the bilingual clustering has an average ϵ-mirror size of 3.46 words for the classes discovered by the mutual information clustering (i.e., classes of words with more than 5 occurrences in the corpus), while the monolingual clustering has an average size of 4.31. We also measured the conditional entropy

$$H(G \mid C_E) = -\sum_{c_E} P(c_E) \sum_{g} t(g \mid c_E) \log t(g \mid c_E) \tag{9}$$

over all classes. This measure reflects the uncertainty of the target German word given a source English class. The conditional entropy is 2.52 with the bilingual-trained classes, and 2.60 with the monolingual trained classes.

5. SUMMARY

In this paper we introduced a word clustering algorithm which takes advantage of a bilingual, parallel corpus to group together words in the source language. The method we have described extends naturally to simultaneously clustering words in both the source and target language. Our method generalizes previous mutual information clustering algorithms for monolingual data by incorporating a statistical translation model, and our preliminary experiments have indicated that the resulting classes can be qualitatively better than those constructed from monolingual data alone.

6. REFERENCES

1. P. F. Brown, S. A. Della-Pietra, V. J Della-Pietra, and R. L. Mercer. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311, 1993.
2. P. F. Brown, V. J. Della-Pietra, P. V. deSouza, J. C. Lai, and R. L. Mercer. Class-based n-gram models of natural language. *Computational Linguistics*, 18(4):467–479, 1992.
3. Reinhard Kneser and Hermann Ney. Improved clustering techniques for class-based statistical language modeling. In *The Proceedings of EUROSPEECH 93*, 1993.
4. Fernando Pereira, Naftali Tishby, and Lillian Lee. Distributational clusteering of english words. In *Proceedings of the 31th Annual Meeting of the ACL*, pages 183–190, 1993.
5. Klaus Ries, Finn Dag Buø, and Ye-Yi Wang. Improved language modeling by unsupervised acquisition of structure. In *Proceedings of the 1995 International Conference on Acoustics, Speech, and Signal Processing*, 1995.
6. B. Suhm, P.Geutner, T. Kemp, A. Lavie, L. Mayfield, A. McNair, I. Rogina, T. Schultz, T. Sloboda, W. Ward, M. Woszczyna, and A. Waibel. JANUS: Towards multilingual spoken language translation. In *Proceedings of the ARPA Speech Spoken Language Technology Workshop, Austin, TX, 1995*, 1995.