[TMI 99: *Proceedings of 8th International Conference on Theoretical and Methodological Issues in Machine Translation*, August 1999, Chester, UK]

Bilingual Clustering Using Monolingual Algorithms

Sergio Barrachina Juan Miguel Vilar
Departamento de Informática
Universidad Jaume I
12071 Castellón, SPAIN
{barrachi|jvilar}@inf.uji.es

Abstract

The use of bilingual word classes greatly reduces the amount of data needed for training subsequential transducers, a finite state model adequate for small to medium translation tasks. We present an automatic approach to derive these classes using traditional monolingual word clustering methods.

1 Introduction

Subsequential Transducers are a kind of finite state models that have been successfully used for small to medium translation tasks (Amengual et al. 1997b; Vilar et al. 1995). They have different advantages. As they are finite state models, it is very easy to combine them with state of the art speech recognizers. They are simple enough to be learnable from sufficiently large sets of training samples. Furthermore, they are powerful enough to capture several phenomena present in the translation of natural languages, as small differences in word order like those in many Indoeuropean language pairs.

On the other hand, the large amount of samples needed in order to produce acceptable models constitutes the biggest problem with SSTs. Fortunately, it has been shown (Amengual et al. 1997a) that a small set of well chosen categories can substantially reduce the amount of required data.

The automatic grouping of words in classes has been extensively investigated in the field of language modeling (Brown et al. 1992; Jelinek et al. 1990; Kneser & Ney 1993). The idea driving those methods is to find the best grouping of words in order to minimize the perplexity of a class-based n-gram.

We present an approach that employs traditional monolingual clustering methods to find classes that can be used in translation. The main idea is to define a new language consisting in sentences where the words are labelled with their translations. These sentences are then used for finding the clusters that will be used in training the transducers.

The rest of this paper is structured as follows: the second section introduces some basic concepts, the third section explains our approach, finally, some experiments and conclusions are presented.

2 Basic Concepts

Our aim is to automatically derive word classes that can be used for translating from one language (which we call *source* language) into another (the *target* language). We consider these classes to be sets of pairs such that the second element of the pair is the translation of the first. The objective is that the different pairs in a class be interchangeable: if in a given source sentence a word is substituted by another from the same class, the sentence can be translated by doing the corresponding substitution in the original translation.

These classes will be later integrated in Subsequential Transducers (that are our basic translation model) using the approach from (Amengual et al. 1997a).

2.1 Classes and Translation

It has been shown (Fung & Wu 1995) that the mapping between source and target language tags might not be meaningful in a translation model: it is not evident that there should be a direct correspondence between parts of speech in two different languages. The relationship that could be derived from a source language part of speech and a target language part of speech mapping is therefore not necessarily a good constraint for the translation search. Instead of mapping parts of speech, Fung & Wu (1995) investigated the mapping between words in the source language and parts of speech in the target language.

In the same direction, Och & Weber (1998) argue that word equivalence classes independently derived for two different languages are not always correlated: the class of a source language word will not always give much information about the class of the generated target language word. They propose an approach to compute bilingual correlated classes consisting in deriving word classes for the target language using a monolingual method and afterwards determining the word classes for the source language taking the other classes into account.

Also, qualitatively better classes of the source language than those constructed from monolingual data alone can be obtained if the existent mutual information clustering algorithms for monolingual data are generalized by incorporating a statistical translation model (Wang et al. 1996).

2.2 Subsequential Transducers

As we mentioned above, the basic model that we use for translations is the Subsequential Transducer (SST). A formal description of the model can be found in (Berstel 1979). Here we give only an informal explanation.

An SST is a deterministic finite state network that accepts sentences from a given input language and produces associated sentences of an output language. It is composed of states and transitions connecting them. Each transition has associated an input symbol and an output string. The condition of determinism implies that no two distinct transitions departing from a given state have the same input symbol. The processing of an input sentence begins from a distinguished state (the initial state) and proceeds by consuming input symbols one by one. Every time an input symbol is accepted, the string associated to the corresponding transition is output and a new state is reached.

This process continues on until the whole input is processed; then, additional output may be produced from the last state reached in the analysis of the input.

The use of the empty string as output allows the SSTs to deal with differences in word order. For example, *habitación doble* is translated into 'double room' in two steps: after seeing *habitación* the empty string is produced; after seeing *doble* the words 'double room' are produced. Structural mismatches are handled similarly: for a sentence like *me llamo Antonio* the SST produces the empty string after seeing *me;* then after seeing *llamo* it produces 'my name is'; and finally *Antonio* produces 'Antonio'.

A distinctive advantage of SSTs is the fact that they can be efficiently learned from unambiguous[1] training sets of input-output examples. This can be done by means of OSTIA and similar algorithms (Vilar 1998). These algorithms basically work in three steps:

1. A finite state prefix tree acceptor is built from the input sentences. Then, empty strings are assigned as output substrings to the transitions of this tree, while every output sentence is associated as a whole to the state reached by the corresponding input string.

2. The longest common prefixes of the output strings are recursively moved, level by level, from the leaf states of the tree towards the root.

3. Starting from the root state, all pairs of states are orderly considered, level by level, and they are merged if merging is *acceptable:* the resulting transducer is subsequential; it is not in contradiction with the training set; and other additional constraints are fulfilled.

The constraints mentioned in the last step can take two forms: finite state models for the source and target language; or bilingual dictionaries together with word alignments. For the first case, the version of OSTIA called OSTIA-DR has to be used and for the second case the appropriate version is OMEGA (which can also take into account models for the source and target languages).

3 An Extension of Traditional Clustering Algorithms

Several methods for monolingual clustering have been described in the literature (Brown et al. 1992; Jelinek et al. 1990; Kneser & Ney 1993), where every word is assigned to a class seeking to minimize the perplexity of a class-based *n*-gram model. We propose to extend these methods so they can be used in machine translation. To achieve this, we have developed an algorithm *(e-cluster)* that clusters words in both source and target languages in classes that are meaningful to machine translation.

In contrast to classical monolingual clustering methods, *e-cluster* permits that a word belongs to more than one class. In monolingual clustering, a word could only belong to one class, because when using *n*-gram language models there is no easy way to determine when one word could belong to different classes. However, when doing translation, the information of the source (target) sentences can be used to distinguish

[1] The training set contains no sentence with two different translations.

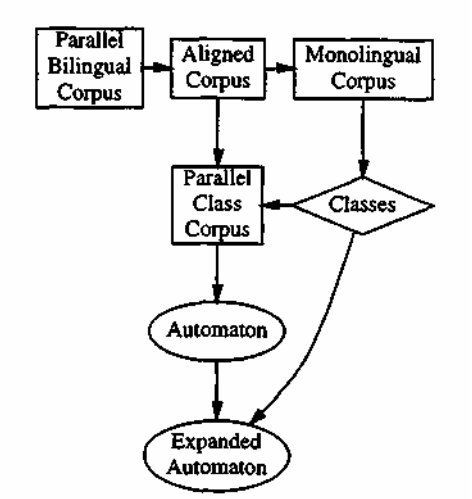

Figure 1: Interrelation of the data produced by the *e-cluster* algorithm.

different meanings of the same target (source) word. As a result, one word in the target (source) language can be in more than one class depending on the several translations of it.

The *e-cluster* algorithm comprises the following steps:

- The training corpus is automatically aligned by an statistical method.
- From the alignments, a new monolingual corpus is generated by labelling each output word with the associated input word.
- The words of this new corpus are clustered using a traditional algorithm.
- The original corpus is labelled according to these clusters.
- An automaton is derived from this labelled corpus and the corresponding classes are expanded.

The interrelation between the different steps and the data produced can be seen in Figure 1.

3.1 Aligning the Corpus

Following (Brown et al. 1993) it is possible to find for each target sentence an alignment that relates every target word with the source word that most probably generated it. These are the alignments that we will use when creating the extended monolingual corpus.

When the training material is scarce, the alignments obtained can be erroneous, leading to errors in later phases of the algorithm. On the other hand, as it is not necessary to classify each word, we decided to filter out the alignments in order to focus in the most promising ones. One method to accomplish this, as mentioned in (Och & Weber 1998), is based on the observation that automatic word alignment techniques do not in general lead to the same results when applied from target to source language

and the other way round. As we need the relations of the sentences to be symmetric —there should be the same word alignment in one direction and in the other— we consider only those words that are linked in both directions. This also increases our confidence in the alignments. We have called the alignment produced by this method the *cross alignment* of the bilingual parallel corpus.

As a running example for the rest of the paper we will use the following sentence pair:

(1) *por favor , tengo reservada una habitación .*
'I have booked a room .'

A direct alignment is:

(2) *por favor , tengo reservada una habitación .*
'I (4) have (4) booked (5) a (6) room (7) . (8)'

The numbers between parentheses show the position in the source sentence where the aligned source word is (i.e. the word *booked* is aligned with the fifth word in the source sentence: *reservada).* An inverse alignment is:

(3) *por favor (3) , (4) tengo (2) reservada (3) una habitación (5) . (6)*
'I have booked a room .'

The corresponding cross alignment is:

(4) *por favor , tengo reservada una habitación .*
'I have (4) booked (5) a room (7) . (8)'

Note that this crossing has different features. For instance, a source word will be aligned with at most one target word. Also, not all the words in the target sentence have to be aligned (in this example a could be aligned with *una* but, as this alignment did not appear in one of the directions, it was discarded).

3.2 Creation of a Monolingual Corpus

Making use of the cross alignments obtained, the target vocabulary is extended with target words labelled with the source words they are aligned to. We call *e-words* (extended words) to the pairs formed by a target word and the corresponding source word.

This labelling of the target words with source words allows a differentiation between the same target word when produced as the translation of different source words. An *e-word* then, contains more information about its meaning or about the context in which it is likely to appear, than the original target word.

A new corpus (we call it *e-corpus)* based on the target language sentences is generated by substituting each word with the corresponding *e-word.* The result of this process is the generation of a monolingual corpus whose sentences are formed by words in the same order that the original target sentences.

The vocabulary of this *e-corpus* will be in general greater than the original target language vocabulary. It is important to note that the new vocabulary will contain not only the newly generated *e-words,* but also original target language words that weren't aligned in some of the sentences where they appeared.

The example (4) produces:

(5) 'I [have,tengo] [booked,reservada] a [room,habitacion] [.,.]'

As it can be seen, the new generated *e-corpus* is formed by original words in the target language (English) and the new defined *e-words*. While in this example the word *a* is present instead of the *e-word* [*a,una*], in other sentences of the *e-corpus,* the *e-word [a,una]* could appear.

3.3 Clustering the *E-words*

We cluster *e-words* into word equivalence classes so that we obtain classes that group words in both the source and target languages. For doing this, we have used the clustering technique described in (Kneser & Ney 1993) and refined later in (Martin et al. 1995).

The basic idea of the algorithm is to begin with an initial classification that is modified in order to improve the perplexity of the class model. The modifications consist in taking each word out from its current class and moving it to the class that minimizes the new perplexity.

We have used the following initial classification. The most frequent *e-words* are each in a different class: if we use N for the number of classes, there will be $N - 1$ classes with these most frequent *e-words*. The rest of the *e-words* are grouped together in a single class. This ensures that the total number of movable classes is exactly N. Finally, each word in the *e-corpus* that is not an *e-word* becomes a single class considered to be non-movable. This way, these words are used in order to compute the perplexity of the model but they are not moved to any class.

The perplexity measure is simply the corresponding to the *e-corpus* when it is generated by the bigram class model induced by the current classification. This measure can be efficiently updated following (Martin et al. 1995).

As stopping criterion, we have decided to fix the number of iterations of the algorithm. Obviously, if in a given iteration no *e-word* is moved, the process stops.

In a more formal way, the algorithm is as follows:

```
set up initial mapping;
compute initial training set perplexity;
do
    for each e-word e in vocabulary
        remove e from its class;
        for all movable classes c
            compute the perplexity if e were moved to c
        assign e to the class with the best perplexity
until a stopping criterion is met
```

After this, we purge from each class those *e-words* that have the same source word, leaving only that *e-word* which has been seen more times in the corpus. Two benefits are likely to be achieved from this: the most probable alignments are left while those with a poorer evidence are filtered out and, at the same time, the ambiguity in the output of the future automata is avoided.

3.4 Labelling the Original Corpus

The original parallel bilingual corpus is rewritten substituting each word in the source and target sentences with a label that identifies the class where it belongs and the original alignments are restored. The new corpus has therefore source words, target words, and class labels.

Our example becomes:

(6) *por favor , C36 C44 una C1 C0*
'I (4) C36 (4) C44 (5) a (6) C1 (7) C0 (8)'

Note that the alignment that we have used here is the one in (2).

3.5 Deriving the SST and Expanding the Classes

Using the new class-based corpus with the original alignments, an SST is trained with an OSTIA-like algorithm.

Also, trivial automata are built for the classes. The classes in the original SST are substituted by these automata using the approach from (Amengual et al. 1997a).

4 Experiments

The experiments on this section have been carried out on the bilingual Spanish-English *Traveler Task Corpus* (Amengual et al. 1997a). This corpus aims at covering usual sentences that can be needed in typical scenarios by a traveler visiting a foreign country whose language he or she does not speak. Clearly, for these situations a word to word translation is infeasible even for the simplest sentences (i.e. in (4) the translation *please* of the source words *por favor* is omitted in the target sentence).

The vocabulary size of this corpus is 686 Spanish words and 513 English words; the sentences average length is 9.7 and 9.9 for Spanish and English respectively.

For the experiments we have used:

- The IBM model 2 (Brown et al. 1993) with smoothing techniques for computing the alignments: the cross alignments to obtain the *e-words,* and the plain IBM model 2 alignments to derive the SSTs.
- The OMEGA (Vilar 1998) algorithm for training the SSTs.
- Error Correcting Parsing (Amengual et al. 1997b) for translating the sentences.

Table 1: Some of the classes automatically derived when generating 125 classes for 10,000 samples.

Class	Members of the class
C5	[pardon,cómo] [when,cuándo] [where,dónde] [please,favor] [why,qué] [who,quién] [pardon,dice]
C10	[forest,bosque] [town,ciudad] [bill,cuenta] [bill,factura] [mountain,montaña] [river,río] [station,estación]
C20	[five,cincuenta] [four,cuarenta] [nine,noventa] [eight,ochenta] [six,sesenta] [seven,setenta] [three,treinta] [one,uno]

To evaluate the *e-cluster* algorithm, we have trained SSTs with 1,000 to 10,000 training pairs, using 25 to 200 classes. For test purposes 3,000 additional independent pairs were extracted from the corpus. The translation of the test sentences produced by the transducers were compared with the reference translations to compute the word error rates (WERs).

The classes listed in Table 1 are part of a typical result of the *e-cluster* algorithm. As it can be seen, the categories can be either syntactic (e.g. C5) or semantic (e.g. C10), this is because our methods don't try to do a semantical analysis of the sentences and the SSTs treat both kind of categories in the same manner.

If we look at the members of the class C20 in Table 1, we can see that numbers in English and in Spanish have been grouped in it. It is worth noting that most of the translation of English units correspond to Spanish tens, this is due to the different manner in which the room numbers are designated in both languages, like in:

(7) *habitación treinta y cuatro*
'room number three four'

This differentiation of the English numbers when they represent units, tens and so on, that could be useful to machine translation in this context, would have not been achieved from clustering both languages independently. As we have already mention, a source or target word may belong to more than one class if the corresponding translations can be used to distinguish the different meanings.

Just for the record, the classes C0, C1, C34, and C44 from our example sentence have the following members C0={ [.,.]}, C1={[room,habitación] }, C34={[have,tengo]}, and C44={ [made,hecha] , [booked,reservada] }. They are so small because the corresponding words are very frequent in the corpus.

The WER evolution with the number of training pairs for different number of classes is represented in Figure 2. If the number of classes is small (in comparison with the vocabulary size) the SSTs produced are poor, but when the number of classes is greater than a minimum (around 75) the WER using classes is lower than without using them.

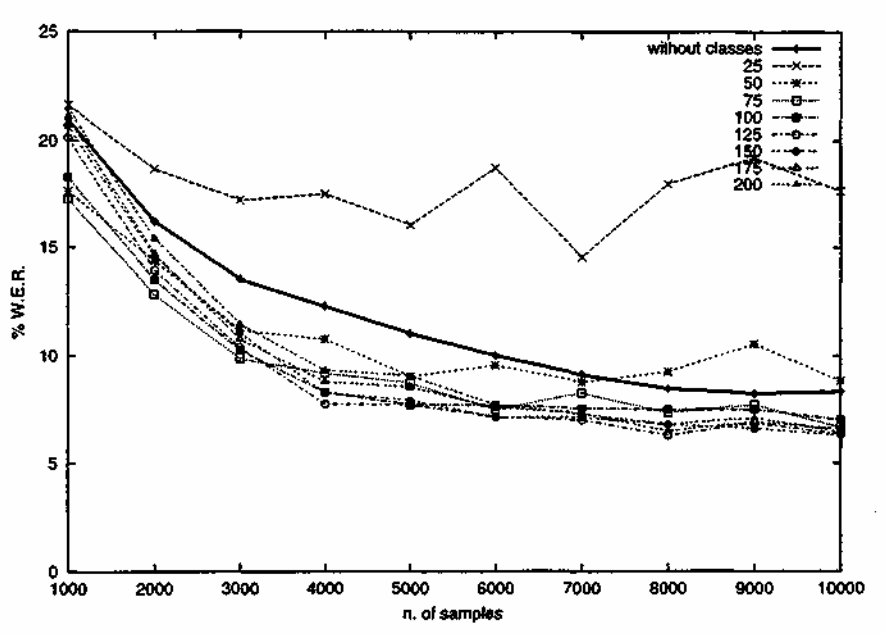

Figure 2: Translation WER for different number of classes and different training sizes.

Although the number of classes has to be manually fixed for the algorithm, we can see in Figure 3 that the WER remains more or less at the same level when the number of classes varies in a wide area, so this manual fixing is not critical.

5 Conclusions and Future Work

Automatic methods can be used for automatically deriving word classes that are useful for translations. These methods can be obtained as extensions to other word clustering algorithms that only consider one language. These extensions add further information to the words so that the obtained classes are more representative of the task at hand and they can be successfully used in order to build translation systems.

We plan to extend the method in order to allow the clustering of units larger than a word (compounds) and to use these classes with other approaches to automatic translation.

6 Acknowledgements

This work has been partially supported by the European Union under the ESPRIT Project number 30268 (EuTrans) and by the Spanish C.I.C.Y.T. project number TIC-97-0745-CO2 (ExTra).

References

Amengual, J. C., J. M. Benedí, F. Casacuberta, A. Castaño, A. Castellanos, D. Llorens, A. Marzal, F. Prat, E. Vidal & J. M. Vilar: 1997a, 'Using categories in the EuTrans system', in *Proceedings of the Spoken Language Translation Workshop,* ACL and European Network in Language and Speech, Madrid (Spain), pp. 44-53.

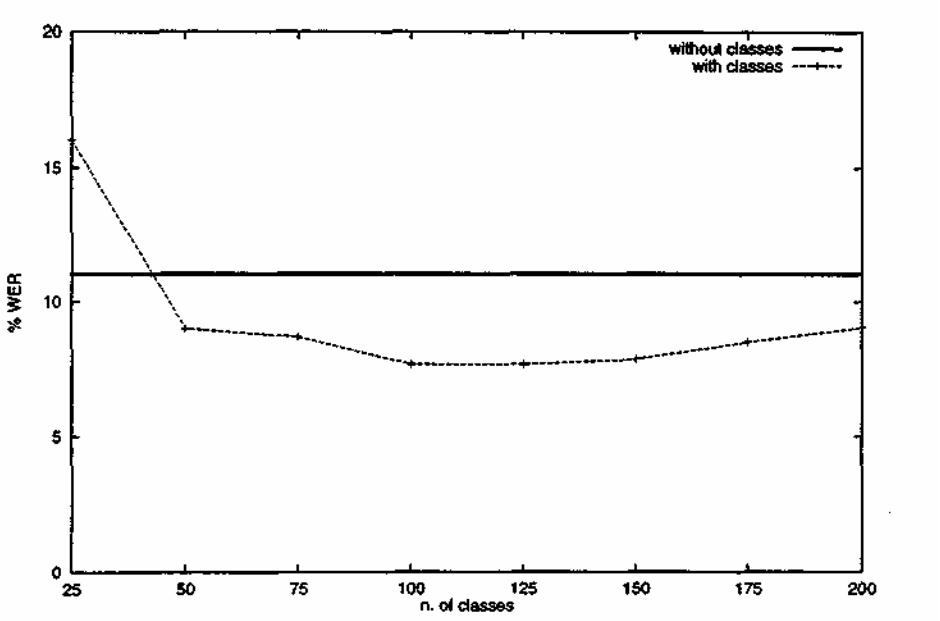

Figure 3: Translation WER for 5,000 training pairs and different number of classes.

Amengual, Juan C., José M. Benedí, Francisco Casacuberta, Asunción Castaño, Antonio Castellanos, David Llorens, Andrés Marzal, Federico Prat, Enrique Vidal & Juan M. Vilar: 1997b, 'Error correcting parsing for text-to-text machine translation using finite state models', in *Proceedings of the TMI'97,* Santa Fe, NM (USA), pp. 135-142.

Berstel, J.: 1979, *Transductions and Context-Free Languages,* Teubner.

Brown, Peter F., Stephen A. Della Pietra, Vincent J. Della Pietra & Robert L. Mercer: 1993, 'The mathematics of statistical machine translation: Parameter estimation', *Computational Linguistics,* 19(2): 263-311.

Brown, Peter F., Vincent J. Della Pietra, Peter V. deSouza, Jenifer C. Lai & Robert L. Mercer: 1992, 'Class-based *n*-gram models of natural language', *Computational Linguistics,* 18(4): 467-479.

Fung, Pascale & Dekai Wu: 1995, 'Coerced markov models for cross-lingual lexical-tag relations', in *Proceedings of the TMI'95,* Leuven (Belgium), pp. 240-255.

Jelinek, Frederick, Robert Mercer & Salim Roukos: 1990, 'Classifying words for improved statistical language models', in *Proceedings of the ICASSP'90,* Albuquerque, NM (USA), pp. 621-624.

Kneser, Reinhard & Hermann Ney: 1993, 'Improved clustering techniques for class-based statistical language modelling', in *Proceedings of the Eurospeech'93,* Berlin (Germany), pp. 973-976.

Martin, Sven, Jorg Liermann & Hermann Ney: 1995, 'Algorithms for bigram and trigram word clustering', in *Proceedings of the Eurospeech'95,* Berlin (Germany).

Och, Franz Josef & Hans Weber: 1998, 'Improving statistical natural language translations with categories and rules', in *Proceedings of the COLING'98,* Montreal (Canada).

Vilar, J. M., A. Castellanos, V. M. Jimenez, J. Oncina, H. Rulot, J. A. Sánchez & E. Vidal: 1995, 'Spoken-language machine translation in limited domains: Can it be achieved by finite-state models?' in *Proceedings of the TMI'95,* Leuven (Belgium), pp. 326-333.

Vilar, Juan Miguel: 1998, 'Aprendizaje de traductores subsecuenciales para su empleo en tareas de dominio restringido', Ph.D. thesis, Dpto. de Sistemas Informáticos y Computación. Univ. Politécnica de Valencia, Valencia (Spain).

Wang, Ye-Yi, John Lafferty & Alex Waibel: 1996, 'Word clustering with parallel spoken language corpora', in *Proceedings of the ICSLP'96,* Philadelphia, Pennsylvania, (USA).