Automatic discovery of translation collocations from bilingual corpora[1]

Sergio Barrachina [2] and **Juan Miguel Vilar** [3]

Abstract.

We describe a method to automatically discover translation collocations from a bilingual corpus and how these improve a machine translation system. The process of inference of collocations is iterative: an alignment is used to derive an initial set of collocations, these are used in turn to improve the alignment and this new alignment is used to generate new collocations. This process is repeated until no more collocations are found. The final alignment and the set of collocations are used to train a translation model. We use a model that is based on finite state transducers and word clusters and has been modified to work with collocations in addition to single words.

We present experiments in which we show that automatic collocations improve translation quality without prior linguistic information.

1 INTRODUCTION

Traditionally, a collocation is defined as:

> "a sequence of one or more consecutive words that has characteristics of a syntactic and semantic unit, and whose exact and unambiguous meaning or connotation cannot be derived directly from the meaning or connotation of its components." [4, pag. 183]

Several linguistic criteria are used to manually identify collocations [4, pag. 184]: *non-compositionality*, the meaning of a collocation is not a straightforward composition of the meanings of its parts (e.g. *kick the bucket*); *non-substitutability*, the components of a collocation can not be substituted even if, in context, they have the same or similar meaning (e.g. while *white wine* is correct, *yellow wine* is not); and *non-modifiability*, many collocations can not be freely modified with additional lexical material or through grammatical transformations (e.g. *to get a frog in one's throat* can not be changed into *to get an ugly frog in one's throat*).

Additionally, there are also are several methods to automatically find collocations from monolingual corpora: selection of collocations by frequency, selection based on mean and variance of the distance between focal and collocating words, hypothesis testing, and mutual information. Nevertheless, little research on automatically identifying collocations from two languages in order to improve machine translation has been conducted.

[1] This work has been partially funded by the European Union's TransType 2 project (n. IST-2001-32091) and the *Fundación Caixa Castelló – Bancaja*'s SIEsTA project (n. P1·1B2002-15)

[2] Dpt. of Computer Engineering and Science, Universitat Jaume I, Castellón, Spain, email: barrachi@icc.uji.es

[3] Dpt. of Programming Languages and Computer Systems, Universitat Jaume I, Castellón, Spain, email: jvilar@lsi.uji.es

Melamed [5] proposed an approach based on the use of parallel texts to find collocations. His approach exploits the idea that a collocation usually is not translated word by word in other languages. Therefore, comparing texts from two languages should reveal those word sequences that are collocations. His method identifies collocations by comparing the predictive power of two translation models that differ on whether they treat or not a word sequence as a collocation. As inducing a translation model is computationally expensive, deriving translation models for each possible collocation would be unpracticable. To overcome this limitation, Melamed did some independence assumptions that allowed simultaneous testing of several possible collocations. Using this procedure Melamed succeeded in finding collocations that are natural to each language.

Usually, the methods employed to discover collocations answer the next question: is this sequence of words a collocation? Their answer is either yes or no. However, we think that it could be convenient for machine translation purposes that a given sequence of words could be treated as a collocation in certain contexts and as individual words in others.

Hence, our aim is to find whether a sequence of words should be treated as a collocation and when. The method that we propose will not decide if a sequence of words is a collocation or not but if a sequence of words when is translated in a particular way should be considered as a collocation or not. Since we are more interested in improving translation models than on finding linguistically sound collocations, our method considers as collocations certain sequences that linguistically cannot be considered as such, but that help on the development of automatic translation systems. To emphasize this difference, we call them *translation collocations*.

One example of such a translation collocation could be the English word sequence *give us* when is translated as the Spanish word *darnos*. Clearly, this word sequence is not a proper collocation in English. Nevertheless, it could be useful to treat *give us* like a translation collocation of *darnos*. Moreover, as we have stated before, this does not imply that each time the sequence *give us* occurs we will treat it as a collocation. For example, given the next pair of sentences:

¿Nos podría dar las llaves de la habitación?

Would you mind giving us the keys to the room?,

we can see that the words *giving* and *us* should not be treated as a single entity but as a sequence of simple words (as *giving* is the translation of *dar* and *us* the translation of *nos*). Therefore, we would like to treat *giving us* like a translation collocation of *darnos*, but not of *dar* neither of *us*. In addition, *giving us* should be treated like a translation collocation of other Spanish words, e.g. *entregarnos*.

Automatically generated translation systems can benefit from the knowledge of translation collocations (sequences of words in a language that act as if they were a single concept in the other language).

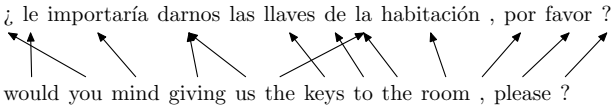


Figure 1. A pair of aligned sentences.

In order to automatically find them, we propose a new method, which is an extension of one of I. Dan Melamed's [5].

In the next section we present the notation we have used in the rest of the paper. In Section 3 we show how a valid translation collocation can be discovered by means of a simple translation model. In Section 4 we present the algorithm used to automatically discover the translation collocations from a bilingual corpus. In Section 5 we describe the machine translation system in which we have integrated the translation collocations. In Section 6 we show the experimental results. Finally, in Section 7 we present the conclusions.

2 A BIT OF NOTATION

For identifying these translation collocations we will use a bilingual corpus, $\mathcal{C}_{xy}$. This bilingual corpus is formed by pairs of sentences in two languages. Each sentence pair has a sentence in a *source* language and its translation in a *target* language. We have used x and y to denote words in source and target languages, respectively. The corpus can also be seen as two monolingual corpora, one comprising the input sentences, $\mathcal{C}_x$ and one comprising the output sentences, $\mathcal{C}_y$. To represent a word sequence we use $\bar{y}$ or $y_1 \dots y_n$, when we want to emphasize the individual words.

To express the relations between the words in a pair of sentences we have used the concept of alignment as defined by Brown et al. [3]. Basically, an alignment relates each target word with a source word or with no source word. In the latter case, we will say the target word is aligned with the empty word, λ. This way, a source word can be aligned with more than one target word, but a target word can be aligned at most with a source word.

3 IDENTIFICATION OF TRANSLATION COLLOCATIONS

To identify the translation collocations as defined in the previous section we propose a *simple translation model* (STM). The model will be greatly influenced by those sequences that are treated as collocations. This way, we will be able to use the *goodness* of the model in order to identify valid translation collocations.

We define STM, $\mathrm{P}(x, \bar{y})$, as a translation model in which each source word produces a word sequence in the target language. These sequences are those aligned with the same source word and need not be consecutive in the target sentence. They would nevertheless retain their original order. This way, two word sequences with the same words but in different order are considered different word sequences.

STM allows that a source word be aligned with no target word. In this case, we will say that this word is aligned with the empty word, λ. In addition, each source sentence is supposed to have an empty word that is aligned with a target word sequence: those target words that are not aligned with any source word.

Note that this model is not intended to be used for actual translation, it does not predict the order of the words in the target sentence.

For an example, consider the alignment shown in Figure 1. This alignment has been obtained after training the IBM model 2 [3]. It generates the following pairs of source word and target word sequences:

x	$\bar{y}$	x	$\bar{y}$
λ	λ	de	to
¿	you	la	the the
le	would	habitación	room
importaría	mind	,	λ
darnos	giving us	por	,
las	λ	favor	please
llaves	keys	?	?

where the pair (λ,λ) represents that the empty word in the source sentence has generated a target word sequence formed by an empty word (i.e. in this example all the words in the target sentence have been aligned with a source word). The next pair connects the source word "¿" with the word sequence *you*. The next one, *le* with *would*. And so on. It can also be seen that the source words *las* and "," that are not aligned with any target word produce the pairs (*las*, λ) and ("," , λ). Moreover, the pair (*la*, *the the*) shows that non consecutive words can be part of a word sequence (although this example is due to an erroneous alignment).

We estimate $\mathrm{P}(x, \bar{y})$ using a bilingual corpus. First, we automatically align all the sentences in the corpus and use the counts between pairs of aligned words to estimate its parameters. Using the maximum-likelihood estimate, $\mathrm{P}(x, \bar{y})$ can be computed as:

$$\mathrm{P}(x, \bar{y}) = \frac{\mathrm{n}(x, \bar{y})}{N}, \tag{1}$$

where $\mathrm{n}(x, \bar{y})$ is the number of times the word sequence $\bar{y}$ has been aligned with the source word x, and N is the total number of alignments between target word sequences and source sentences: due to the way the model has been defined, N is given by the total number of source words seen (plus an empty word per sentence).

We can think of each pair $(x, \bar{y})$ as values from a pair of random variables X and $\bar{Y}$. These random variables represent possible source words and target word sequences, respectively. We can measure the *goodness* of the model using the cross-entropy of X and $\bar{Y}$:

$$\mathrm{H}(X, \bar{Y}) = -\sum_{x \in X} \sum_{\bar{y} \in \bar{Y}} \mathrm{P}(x, \bar{y}) \cdot \log \mathrm{P}(x, \bar{y}). \tag{2}$$

The lower the cross-entropy, the better the estimation of $\mathrm{P}(x, \bar{y})$. This estimation depends on the counts of the alignments between source words and target word sequences. To evaluate if a given word (sub)sequence, $y_1 \dots y_n$, is a translation collocation of some source word x, it suffices to train two translation models: the *base* model that does not treat $y_1 \dots y_n$ as a translation collocation of x; and the *modified* model, which takes $y_1 \dots y_n$ as a translation collocation of x. The modified model forces all the words of each occurrence of the sequence $y_1 \dots y_n$, when at least one of them was aligned with x, to be aligned with the same source word. Therefore, the words $y_1 \dots y_n$ will be aligned in the modified model with the same source word in each sentence in which some of them, but not necessarily all, were initially aligned with x. If the cross-entropy of the modified model is lower than that of the base model, we will consider $y_1 \dots y_n$ as a translation collocation of x.

For example, suppose that we want to check whether *giving us* is a valid translation collocation of *darnos*. First of all, we generate the alignment of the bilingual corpus. Then, we estimate the base translation model $\mathrm{P}(x, \bar{y})$ and compute its cross-entropy $\mathrm{H}(X, \bar{Y})$. After

this, we modify the training corpus replacing the sequence *giving us* each time either *giving* or *us* are aligned to *darnos* with a single new word: *giving_us*. We generate an alignment for this modified corpus so that *giving* and *us*, in those sentences in which they were replaced by *giving_us*, will be aligned with the same source word. Then, we estimate the modified translation model $\mathrm{P}'(x,\bar{y})$ and compute its cross-entropy $\mathrm{H}'(X,\bar{Y})$. Finally, if $\mathrm{H}'(X,\bar{Y}) \leq \mathrm{H}(X,\bar{Y})$ then we consider *giving us* like a translation collocation of *darnos*.

It should be pointed out that the new *joined* words (i.e. *giving_us* in the preceding example) are used only to obtain the new alignments; when the modified translation model is estimated these joined words are split again into their components. For example, if the joined word *giving_us* has been aligned with *darnos*, then the pair (*darnos, giving us*) will be part of the translation model, not (*darnos, giving_us*).

It is easy to observe that the process described so far to generate a modified translation model only changes the distribution of $\bar{Y}$. The distribution of X remains the same. Therefore, to validate a translation collocation, as $\mathrm{H}(X)$ is constant between models and $\mathrm{H}(X,\bar{Y}) = \mathrm{H}(\bar{Y}|X) + \mathrm{H}(X)$, it suffices to evaluate the variation of $\mathrm{H}(\bar{Y}|X)$ between models. $\mathrm{H}(\bar{Y}|X)$ is defined as:

$$\mathrm{H}(\bar{Y}|X) = -\sum_{x\in X}\sum_{\bar{y}\in\bar{Y}} \mathrm{P}(x,\bar{y}) \cdot \log \mathrm{P}(\bar{y}|x), \qquad (3)$$

where $\mathrm{P}(\bar{y}|x)$ is estimated as $\frac{\mathrm{n}(x,\bar{y})}{\mathrm{n}(x)}$.

The generation and evaluation of each new model is a costly operation. Therefore, it is desirable to evaluate more than one translation collocation in each turn. Furthermore, as the number of translation collocations that must be evaluated is large, it also would be desirable to estimate the improvement of $\mathrm{H}(\bar{Y}|X)$ due to each candidate translation collocation in order to select which translation collocations should be evaluated first. A last question that we would like to address is which translation collocations should be evaluated. All the possible word sequences? Is there any way to restrict the number of word sequences to be tried? Our proposals to these three issues are in the next subsections.

3.1 Simultaneous evaluation of translation collocations

Several translation collocations can be simultaneously evaluated from their individual contribution to the improvement of $\mathrm{H}(\bar{Y}|X)$. To simplify the evaluation of these contributions we restrict ourselves to translation collocations involving just two words. We can build larger translation collocations treating previously accepted word sequences as if they were a single word.

To evaluate the contribution of the candidate translation collocation y_1y_2 as the translation of x_c to the improvement of $\mathrm{H}(\bar{Y}|X)$ we make the following assumption:

Assumption 1 *If a modified translation model is generated in which the target words y_1 and y_2 are treated like a translation collocation of x_c, the distribution of $\mathrm{P}(x,\bar{y})$ will only be modified for those x, besides x_c, which were originally aligned with y_1 or y_2 when y_1 and y_2 occurred together and any of y_1 and y_2 were aligned with x_c.*

If c is the translation collocation y_1y_2 of x_c and X_c is the subset of X that has x_c and those x that were aligned with y_1 or y_2 when they occurred together and any of them were aligned with x_c; we make use of assumption 1 to evaluate the improvement of $\mathrm{H}(\bar{Y}|X)$ due to c as:

$$\begin{aligned}\Delta\mathrm{H}(\bar{Y}|X) \;&=\; \mathrm{H}'(\bar{Y}|X) - \mathrm{H}(\bar{Y}|X)\\ &\simeq\; -\sum_{x\in X_c}\sum_{\bar{y}\in\bar{Y}} \mathrm{P}'(x,\bar{y}) \log \mathrm{P}'(\bar{y}|x)\\ &\quad + \sum_{x\in X_c}\sum_{\bar{y}\in\bar{Y}} \mathrm{P}(x,\bar{y}) \log \mathrm{P}(\bar{y}|x). \qquad (4)\end{aligned}$$

Mutual exclusion condition. We can conclude from (4) that in order to simultaneously evaluate n translation collocations, $c_1 \ldots c_n$, we should at least guarantee that the X_c are pairwise disjoint. If we do not respect this condition we will not be able to know the individual contribution of each translation collocation to $\Delta\mathrm{H}(\bar{Y}|X)$. It should be pointed out that from a practical point of view the empty word should be excluded from this exclusion condition.

3.2 Estimation of the contribution of each translation collocation

The mutual exclusion condition limits the number of translation collocations that can be simultaneously evaluated. Therefore, it is desirable to estimate the contribution of each translation collocation. This estimation can be used to select which candidate translation collocations should be evaluated first. Moreover, if the estimation of a candidate translation gives a negative result we could reject in advance this translation collocation.

If we express (4) as a function of $\mathrm{n}(x,\bar{y})$, the increment of $\mathrm{H}(\bar{Y}|X)$ due to the translation collocation c is:

$$\begin{aligned}\Delta\mathrm{H}(\bar{Y}|X) \simeq &-\sum_{x\in X_c}\sum_{\bar{y}\in\bar{Y}} \frac{\mathrm{n}'(x,\bar{y})}{N'} \log \frac{\mathrm{n}'(x,\bar{y})}{\mathrm{n}'(x)}\\ &+\sum_{x\in X_c}\sum_{\bar{y}\in\bar{Y}} \frac{\mathrm{n}(x,\bar{y})}{N} \log \frac{\mathrm{n}(x,\bar{y})}{\mathrm{n}(x)}. \qquad (5)\end{aligned}$$

To estimate this increment we should estimate $\mathrm{n}'(x)$, N' and $\mathrm{n}'(x,\bar{y})$. The number of times a source word occurs in a source word target and word sequence pair does not depend on whether some target words have been joined. Therefore, $\mathrm{n}'(x)$ is equal to $\mathrm{n}(x)$. On the other hand, $N' = N$ since in each sentence, each source word, including the empty word, will always be aligned with a target word sequence (which could be the empty word). This means that only $\mathrm{n}'(x,\bar{y})$ has to be estimated.

The following assumptions are used to estimate $\mathrm{n}'(x,\bar{y})$:

Assumption 2 *In the modified model, only those sequences $\bar{y}$ that had either y_1 or y_2 but not both of them will be modified. Moreover, these sequences will be modified in the following manner. If $\bar{y}$ was originally aligned with x_c then either y_1 or y_2 will be added to $\bar{y}$ (the one that was not in $\bar{y}$). On the other hand, if $\bar{y}$ was not originally aligned with x_c then either y_1 or y_2 will be removed from $\bar{y}$ (the one that was in $\bar{y}$).*

Assumption 3 *The new word sequences $\bar{y}$ will remain aligned with the same source word x, with which the word sequences they come from were aligned.*

Let $\hat{\mathrm{n}}(x,\bar{y})$ be the estimation of $\mathrm{n}(x,\bar{y})$. Using assumptions 1 and 3, it is easy to develop an algorithm to obtain $\hat{\mathrm{n}}(x,\bar{y})$ for a given translation collocation, c. Once $\hat{\mathrm{n}}(x,\bar{y})$ has been computed, we estimate the increment of $\mathrm{H}(\bar{Y}|X)$ due to the translation collocation c as:

$$\hat{\Delta}\mathrm{H}(\bar{Y}|X) = -\sum_{x\in X_c}\sum_{\bar{y}\in\bar{Y}} \frac{\hat{\mathrm{n}}(x,\bar{y})}{N} \log \frac{\hat{\mathrm{n}}(x,\bar{y})}{\mathrm{n}(x)}$$

$$+ \sum_{x \in X_c} \sum_{\bar{y} \in \bar{Y}} \frac{\mathrm{n}(x, \bar{y})}{N} \log \frac{\mathrm{n}(x, \bar{y})}{\mathrm{n}(x)}. \qquad (6)$$

3.3 Selection of candidate collocations

If we test each possible sequence of target words as a candidate translation collocation of any source word, we would have to test a vast number of collocations. In order to limit the number of tests, we have restricted ourselves to those word pairs $y_1 y_2$ that have been simultaneously aligned at least once with the same source word. This narrows the search to those pairs of words that the alignment model suggests as possible translation collocations.

4 ALGORITHM TO AUTOMATICALLY DISCOVER TRANSLATION COLLOCATIONS

So far, we have shown how to obtain a candidate translation collocation and how to estimate and evaluate it. The following algorithm shows how to use these methods to automatically discover the translation collocations present in a bilingual corpus.

1. Initialize a *list of not valid collocations* and a *list of valid collocations*.
2. Obtain a base STM, $\mathrm{P}(x, \bar{y})$, from $\mathcal{C}_{xy}$.
3. Compute the $\hat{\Delta}\mathrm{H}(\bar{Y}|X)$ for those pairs of words $y_1 y_2$ in $\mathcal{C}_y$ and the words x in $\mathcal{C}_x$ that are not in the *list of not valid collocations* such that y_1 and y_2 are simultaneously aligned at least in one sentence to x.
4. Produce a *list of candidate collocations* with those collocations such that $\hat{\Delta}\mathrm{H}(\bar{Y}|X) \leq 0$. Order the list by increasing values of $\hat{\Delta}\mathrm{H}(\bar{Y}|X)$.
5. Remove from the *list of candidate collocations* those c that do not satisfy the mutual exclusion condition with any previous c' in the list.
6. Generate $\mathcal{C}'_{xy}$ by substituting the target word sequence of each candidate translation collocation by a single joined word.
7. Obtain the modified STM, $\mathrm{P}'(x, \bar{y})$, from $\mathcal{C}'_{xy}$.
8. Compute the contribution to $\Delta\mathrm{H}(\bar{Y}|X)$ for each translation collocation.
9. Add to the *list of valid collocations* those candidates c such that $\Delta\mathrm{H}(\bar{Y}|X)|_c \leq 0$. Add the rest of candidates to the *list of not valid collocations*.
10. Substitute in $\mathcal{C}_{xy}$ the target word sequence of each translation collocation in the *list of valid collocations* by a joined word.
11. If there are candidate collocations that have not been evaluated yet, return to step 2.

This algorithm obtains a list of valid translation collocations and a new bilingual corpus in which each target word sequence that was part of a valid translation collocation has been replaced by a new joined word. For example, if the sequence *giving us* has been identified as a valid translation collocation of some source words, each time that *giving us* appeared in the original corpus aligned with any of these words it will be replaced in the new corpus by *giving_us*.

The way it has been presented, this algorithm obtains translation collocations in the target part of the corpus, but it can be easily modified to obtain translation collocations in both parts.

Table 1. The EUTRANS-I Corpus.

Data		Spanish	English
Training text	Sentence pairs	10,000	
	Different sentence pairs	6,813	
	Running words	132,198	134,922
	Vocabulary	686	513
	Bigram Test-Set Perplexity	8.6	6.3
Test	Sentence pairs (all different)	2,996	
	Running words	35,023	35,590

5 THE MACHINE TRANSLATION SYSTEM

As translation system, we have used one very similar to that presented by Barrachina and Vilar in [2]. The basic idea is to find a set of bilingual categories in the training corpus. These categories are used to define a new corpus that is employed to train a finite state transducer using OMEGA. Then, elementary transducers for the categories are expanded producing the final model.

Our approach introduces the translation collocations in the process of learning the categories. As the original categories included only single words, we expect that the use of collocations has great impact on the quality of the categories found. By the workings of OMEGA, the impact of collocations over the training of the finite state model itself is expected to be less important, as will be confirmed in the experiments.

6 EXPERIMENTAL RESULTS

We have tested our approach experimentally on the EUTRANS-I corpus [1]. This is a Spanish to English translation task involving common sentences given in the front desk of an hotel (see Table 1).

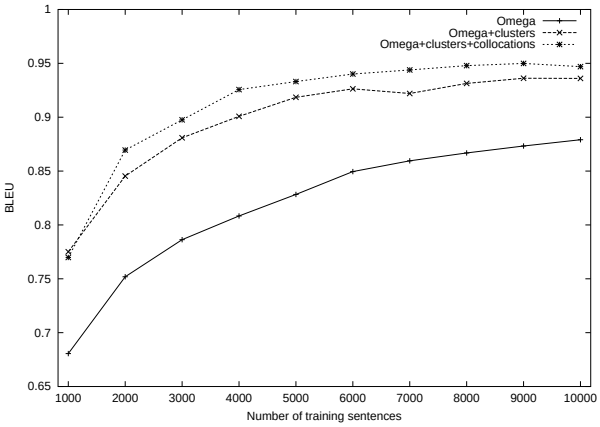
The quality of the translations have been measured by two scores:

BLEU (*Bilingual Evaluation Understudy*) This measure evaluates (from 0 to 1) the agreement in the n-grams between the reference and the proposed translation. The higher the agreement, the better. [6]

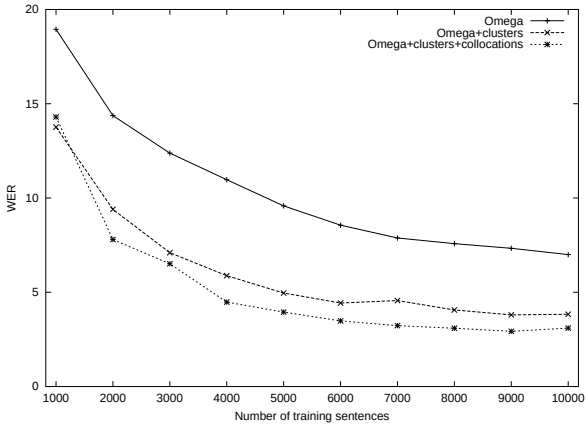
WER (*Word error rate*) This measure counts the number of edition operations (insertions, deletions and substitutions) needed to transform the proposed translation in the reference. The number is normalized by sentence length and expressed as a percentage. The lower the error rate, the better.

In order to evaluate the influence of the automatically discovered translation collocations (see Table 2) on the clustering process different models using from 1,000 to 10,000 training pairs were trained. For each number of training pairs three models were produced: one using only OMEGA, one using OMEGA and clustering, and finally one using OMEGA, collocations and clustering. The differences between OMEGA and OMEGA with collocations were negligible and will not be presented. As it can be seen in Figure 2, the use of collocations prior to the clustering greatly improves the machine translation results. Moreover, the results using collocations and 6,000 training pairs are nearly equivalent to those using 10,000 pairs but no collocations. In fact, they are slightly better (0.940 BLEU and 3.48 WER with collocations vs. 0.936 BLEU and 3.83 WER with no collocations).

Figure 3 shows the influence of collocations for different number of classes for 10,000 training pairs. Clearly, the collocations improve the results for any number of clusters. It should be noted that, as expected, the optimal number of clusters depends on whether collocations are used or not.

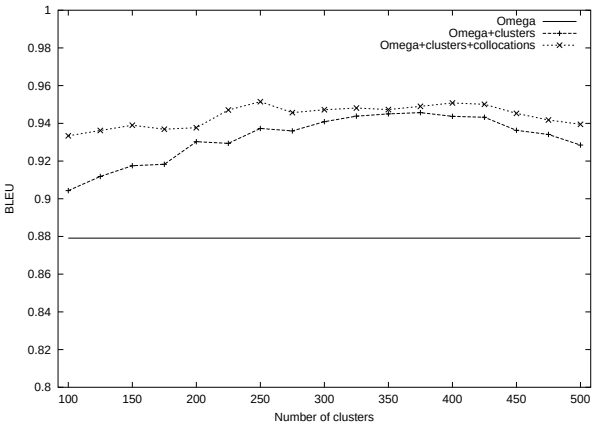


a) BLEU

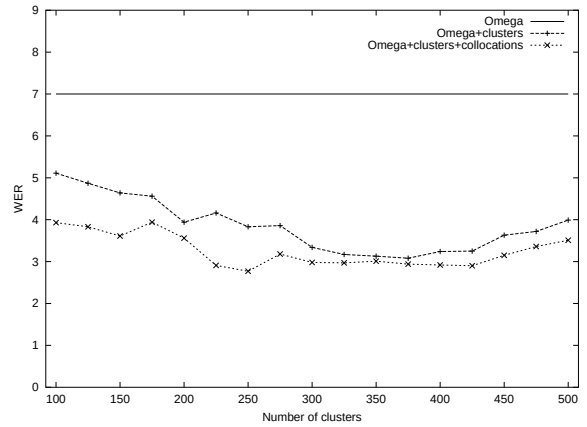


b) WER

Figure 2. a) BLEU and b) WER obtained with Omega, Omega with clusters and Omega with clusters and collocations using from 1,000 to 10,000 training sentences.



a) BLEU



b) WER

Figure 3. a) BLEU and b) WER obtained with Omega, Omega with clusters and Omega with clusters and collocations for 10,000 training sentences using from 100 to 500 clusters.

Table 2. Some automatically discovered translation collocations.

adiós ← good bye	al ← in the
bájeme ← send down	bajara ← to send down
bien ← all right	bosque ← of the forest
cuarto ← quarter past	cuatrocientos ← number four oh oh
pídanos ← could you ask for	querríamos ← we would like

7 CONCLUSIONS

We have presented a method that automatically discovers collocations that can improve translation systems. This method relies on a simple translation model in order to assess the goodness of the collocations. Also, different approximations and estimates are used to reduce the number of collocations tested to practical numbers.

The experiments presented show that the collocations have a large impact on the quality of the models obtained when using them.

REFERENCES

[1] J. C. Amengual, J. M. Benedí, F. Casacuberta, A. Castano, A. Castellanos, D. Llorens, A. Marzal, F. Prat, E. Vidal, and J. M. Vilar, 'Using categories in the EuTrans system', in *Proceedings of the Spoken Language Translation Workshop, ACL and European Network in Language and Speech*, pp. 44–53, Madrid, España, (1997).

[2] Sergio Barrachina and Juan Miguel Vilar, 'Automatically deriving categories for translation', in *Proceedings of the European Conference on Speech Communication and Technology (EuroSpeech'99)*, pp. 2415–2418, Budapest (Hungary), (September 1999).

[3] Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer, 'The mathematics of machine translation: Parameter estimation', *Computational Linguistics*, **19**(2), 263–312, (June 1993).

[4] Christopher D. Manning and Hinrich Schütze, *Foundations of Statistical Natural Language Processing*, MIT Press, Cambridge, Massachusetts, 1 edn., 2000.

[5] I. Dan Melamed, 'Automatic Discovery of Non-Compositional Compounds in Parallel Data', in *the Second Conference on Empirical Methods in Natural Language Processing*, ed., Claire Cardie y Ralph Weischedel, pp. 97–108, Somerset, New Jersey, (1997). Association for Computational Linguistics.

[6] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu, 'Bleu: a Method for Automatic Evaluation of Machine Translation', Technical report, IBM Research Division, (17 September 2001).