Experiments Using MAR for Aligning Corpora*

Juan Miguel Vilar
Departamento de Lenguajes y Sistemas Informáticos
Universitat Jaume I
Castellón (Spain)
jvilar@lsi.uji.es

Abstract

We present some experiments conducted within the context of one of the shared tasks of the ACL 2005 Workshop on Building and Using Parallel Texts. We have employed a new model for finding the alignments. This new model takes a recursive approach in order to find the alignments. As its computational costs are quite high, a method for splitting the training sentences in smaller parts is used.

1 Introduction

We present the experiments we conducted within the context of the shared task of the track on building and using parallel texts for languages with scarce resources of the ACL 2005 Workshop on Building and Using Parallel Texts. The aim of the task was to align the words of sentence pairs in different language pairs. We have participated using the Romanian-English corpora.

We have used a new model, the MAR (from the Spanish initials of Recursive Alignment Model) that allowed us to find structured alignments that were later transformed in a more conventional format. The basic idea of the model is that the translation of a sentence can be obtained in three steps: first, the sentence is divided in two parts; second, each part is translated separately using the same process; and third, the two translations are joined. The high computational costs associated with the training of the model made it necessary to split the training pairs in smaller parts using a simple heuristic.

Initial work with this model can be seen in (Vilar Torres, 1998). A detailed presentation can be found in (Vilar and Vidal, 2005). This model shares some similarities with the stochastic inversion transduction grammars (SITG) presented by Wu in (Wu, 1997). The main point in common is the number of possible alignments between the two models. On the other hand, the parametrizations of SITGs and the MAR are completely different. The generative process of SITGs produces simultaneously the input and output sentences and the parameters of the model refer to the rules of the nonterminals. This gives a clear symmetry to both input and output sentences. Our model clearly distinguishes an input and output sentence and the parameters are based on observable properties of the sentences (their lengths and the words composing them). Also, the idea of splitting the sentences until a simple structure is found in the Divisive Clustering presented in (Deng et al., 2004). Again, the main difference is in the probabilistic modeling of the alignments. In Divisive Clustering a uniform distribution on the alignments is assumed while MAR uses a explicit parametrization.

The rest of the paper is structured as follows: the next section gives an overview of the MAR, then we explain the task and how the corpora were split, after that, how the alignments were obtained is explained, finally the results and conclusions are presented.

*Work partially supported by Bancaixa through the project "Sistemas Inductivos, Estadísticos y Estructurales, para la Traducción Automática (SIEsTA)".

Proceedings of the ACL Workshop on Building and Using Parallel Texts, pages 95–98,
Ann Arbor, June 2005.

2 The MAR

We provide here a brief description of the model, a more detailed presentation can be found in (Vilar and Vidal, 2005). The idea is that the translation of a sentence $\bar{x}$ into a sentence $\bar{y}$ can be performed in the following steps[1]:

(a) If $\bar{x}$ is small enough, IBM's model 1 (Brown et al., 1993) is employed for the translation.

(b) If not, a cut point is selected in $\bar{x}$ yielding two parts that are independently translated applying the same procedure recursively.

(c) The two translations are concatenated either in the same order that they were produced or second first.

2.1 Model parameters

Apart from the parameters of model 1 (a stochastic dictionary and a discrete distribution of lenghts), each of the steps above defines a set of parameters. We will consider now each set in turn.

Deciding the submodel The first decision is whether to use IBM's model 1 or to apply the MAR recursively. This decision is taken on account of the length of $\bar{x}$. A table is used so that:

$$\Pr(\text{IBM} \mid \bar{x}) \approx \mathcal{M}_I(|\bar{x}|),$$
$$\Pr(\text{MAR} \mid \bar{x}) \approx \mathcal{M}_M(|\bar{x}|).$$

Clearly, for every $\bar{x}$ we have that $\Pr(\text{IBM} \mid \bar{x}) + \Pr(\text{MAR} \mid \bar{x}) = 1$.

Deciding the cut point It is assumed that the probability of cutting the input sentence at a given position b is most influenced by the words around it: x_b and x_{b+1}. We use a table $\mathcal{B}$ such that:

$$\Pr(b \mid \bar{x}) \approx \frac{\mathcal{B}(x_b, x_{b+1})}{\sum_{i=1}^{|\bar{x}|-1} \mathcal{B}(x_i, x_{i+1})}.$$

That is, a weight is assigned to each pair of words and they are normalized in order to obtaing a proper probability distribution.

[1]We use the following notational conventions. A string or sequence of words is indicated by a bar like in $\bar{x}$, individual words from the sequence carry a subindex and no bar like in x_i, substrings are indicated with the first and last position like in $\bar{x}_i^j$. Finally, when the final position of the substring is also the last of the string, a dot is used like in $\bar{x}_i^{\cdot}$

Deciding the concatenation direction The direction of the concatenation is also decided as a function of the two words adjacent to the cut point, that is:

$$\Pr(D \mid b, \bar{x}) \approx \mathcal{D}_D(x_b, x_{b+1}),$$
$$\Pr(I \mid b, \bar{x}) \approx \mathcal{D}_I(x_b, x_{b+1}),$$

where D stands for *direct* concatenation (i.e. the translation of $\bar{x}_1^b$ will precede the translation of $\bar{x}_{b+1}^{\cdot}$) and I stands for *inverse*. Clearly, $\mathcal{D}_D(x_b, x_{b+1}) + \mathcal{D}_I(x_b, x_{b+1}) = 1$ for every pair (x_b, x_{b+1}).

2.2 Final form of the model

With these parameters, the final model is:

$$\begin{aligned} p_T(\bar{y} \mid \bar{x}) = & \\ & \mathcal{M}_I(|\bar{x}|) p_I(\bar{y} \mid \bar{x}) \\ & + \mathcal{M}_M(|\bar{x}|) \sum_{b=1}^{|\bar{x}|-1} \frac{\mathcal{B}(x_b, x_{b+1})}{\sum_{i=1}^{|\bar{x}|-1} \mathcal{B}(x_i, x_{i+1})} \\ & \cdot \Bigg(\mathcal{D}_D(x_b, x_{b+1}) \sum_{c=1}^{|\bar{y}|-1} p_T(\bar{y}_1^c \mid \bar{x}_1^b) p_T(\bar{y}_{c+1}^{\cdot} \mid \bar{x}_{b+1}^{\cdot}) \\ & + \mathcal{D}_I(x_b, x_{b+1}) \sum_{c=1}^{|\bar{y}|-1} p_T(\bar{y}_{c+1}^{\cdot} \mid \bar{x}_1^b) p_T(\bar{y}_1^c \mid \bar{x}_{b+1}^{\cdot}) \Bigg) \end{aligned}$$

were p_I represents the probability assigned by model 1 to a pair of sentences.

2.3 Model training

The training of the model parameters is done maximizing the likelihood of the training sample. For each training pair $(\bar{x}, \bar{y})$ and each parameter P relevant to it, the value of

$$\mathcal{C}(P) = \frac{P}{p_T(\bar{y} \mid \bar{x})} \frac{\partial\, p_T(\bar{y} \mid \bar{x})}{\partial\, P} \qquad (1)$$

is computed. This corresponds to the *counts* of P in that pair. As the model is polynomial on all its parameters except for the cuts (the $\mathcal{B}$'s), Baum-Eagon's inequality (Baum and Eagon, 1967) guarantees that normalization of the counts increases the likelihood of the sample. For the cuts, Gopalakrishnan's inequality (Gopalakrishnan et al., 1991) is used.

Table 1: Statistics of the training corpus. Vocabulary refers to the number of different words.

Language	Sentences	Words	Vocabulary
Romanian	48 481	976 429	48 503
English	48 481	1 029 507	27 053

The initial values for the dictionary are trained using model 1 training and then a series of iterations are made updating the values of every parameter. Some additional considerations are taken into account for efficiency reasons, see (Vilar and Vidal, 2005) for details.

A potential problem here is the large number of parameters associated with cuts and directions: two for each possible pair of words. But, as we are interested only in aligning the corpus, no provision is made for the data sparseness problem.

3 The task

The aim of the task was to align a set of 200 translation pairs between Romanian and English. As training material, the text of 1984, the Romanian Constitution and a collection of texts from the Web were provided. Some details about this corpus can be seen in Table 1.

4 Splitting the corpus

To reduce the high computational costs of training of the parameters of MAR, a heuristic was employed in order to split long sentences into smaller parts with a length less than l words.

Suppose we are to split sentences $\bar{x}$ and $\bar{y}$. We begin by aligning each word in $\bar{y}$ to a word in $\bar{x}$. Then, a score and a translation is assigned to each substring $\bar{x}_i^j$ with a length below l. The translation is produced by looking for the substring of $\bar{y}$ which has a length below l and which has the largest number of words aligned to positions between i and j. The pair so obtained is given a score equal to sum of: (a) the square of the length of $\bar{x}_i^j$; (b) the square of the number of words in the output aligned to the input; and (c) minus ten times the sum of the square of the number of words aligned to a nonempty position out of $\bar{x}_i^j$ and the number of words outside the segment chosen that are aligned to $\bar{x}_i^j$.

These scores are chosen with the aim of reducing the number of segments and making them as "complete" as possible, ie, the words they cover are aligned to as many words as possible.

After the segments of $\bar{x}$ are so scored, the partition of $\bar{x}$ that maximizes the sum of scores is computed by dynamic programming.

The training material was split in parts up to ten words in length. For this, an alignment was obtained by training an IBM model 4 using GIZA++ (Och and Ney, 2003). The test pairs were split in parts up to twenty words. After the split, there were $141\,945$ training pairs and 337 test pairs. Information was stored about the partition in order to be able to recover the correct alignments later.

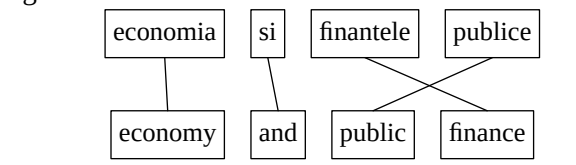
5 Aligning the corpus

The parameters of the MAR were trained as explained above: first ten IBM model 1 iterations were used for giving initial values to the dictionary probabilities and then ten more iterations for retraining the dictionary together with the rest of the parameters.

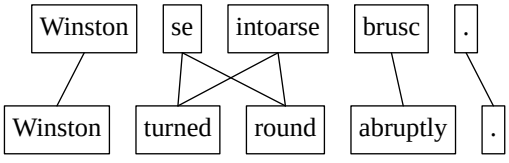
The alignment of a sentence pair has the form of a tree similar to those in Figure 1. Each interior node has two children corresponding to the translation of the two parts in which the input sentence is divided. The leaves of the tree correspond to those segments that were translated by model 1.

As the reference alignments do not have this kind of structure it is necessary to "flatten" them. The procedure we have employed is very simple: if we are in a leaf, every output word is aligned to every input word; if we are in an interior node, the "flat" alignments for the children are built and then combined. Note that the way leaves are labeled tends to favor recall over precision.

The flat alignment corresponding to the trees of Figure 1 are:



and



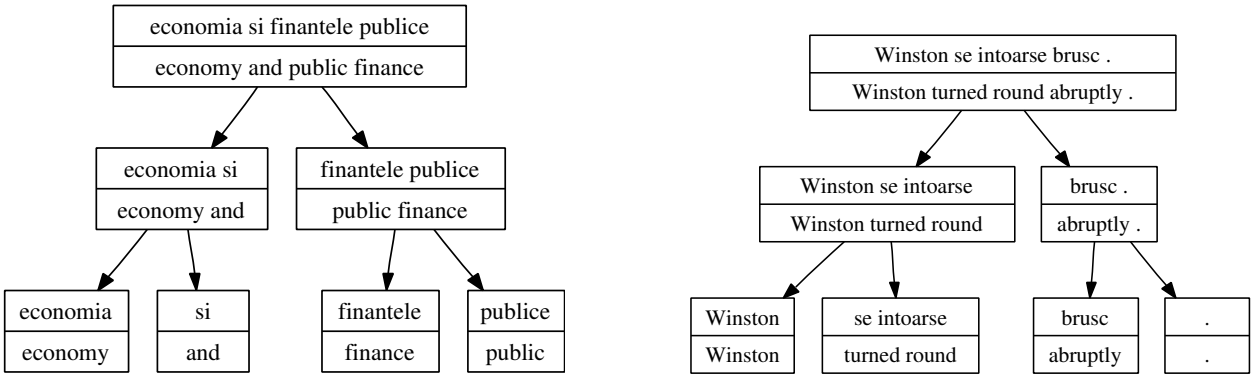


Figure 1: Two trees representing the alignment of two pair of sentences.

Precision	Recall	F-Measure	AER
0.5404	0.6465	0.5887	0.4113

Table 2: Results for the task

6 Results and discussion

The results for the alignment can be seen in Table 2. As mentioned above, there is a certain preference for recall over precision. For comparison, using GIZA++ on the split corpus yields a precision of 0.6834 and a recall of 0.5601 for a total AER of 0.3844.

Note that although the definition of the task allowed to mark the alignment as either *probable* or *sure*, we marked all the alignments as *sure*, so precision and recall measures are given only for sure alignments.

There are aspects that deserve further experimentation. The first is the split of the original corpus. It would be important to evaluate its influence, and to try to find methods of using MAR without any split at all. A second aspect of great importance is the method used for "flattening". The way leaves of the tree are treated probably could be improved if the dictionary probabilities were somehow taken into account.

7 Conclusions

We have presented the experiments done using a new translation model for finding word alignments in parallel corpora. Also, a method for splitting the input before training the models has been presented.

References

Leonard E. Baum and J. A. Eagon. 1967. An inequality with applications to statistical estimation for probabilistic functions of Markov processes and to a model for ecology. *Bulletin of the American Mathematical Society*, 73:360–363.

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

Yonggang Deng, Shankar Kumar, and William Byrne. 2004. Bitext chunk alignment for statistical machine translation. Research Note 50, CLSP Johns Hopkins University, April.

P. S. Gopalakrishnan, Dimitri Kanevsky, Arthur Nádas, and David Nahamoo. 1991. An inequality for rational functions with applications to some statistical problems. *IEEE Transactions on Information Theory*, 37(1):107–113, January.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Juan Miguel Vilar and Enrique Vidal. 2005. A recursive statistical translation model. In *Workshop on Building and Using Parallel Texts*, Ann-Arbour (Michigan), June.

Juan Miguel Vilar Torres. 1998. *Aprendizaje de Traductores Subsecuenciales para su empleo en tareas de dominio restringido*. Ph.D. thesis, Departamento de Sistemas Informáticos y Computación, Universidad Politécnica de Valencia, Valencia (Spain). (in Spanish).

Dekai Wu. 1997. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational Linguistics*, 23(3):377–403.