Machine Translation with Grammar Association: Some Improvements and the Loco_C Model

Federico Prat
Departamento de Lenguajes y Sistemas Informáticos
Universitat Jaume I de Castelló
E-12071 Castellón de la Plana, Spain
fprat@lsi.uji.es

Abstract

Grammar Association is a technique for Machine Translation and Language Understanding introduced in 1993 by Vidal, Pieraccini and Levin. All the statistical and structural models involved in the translation process are automatically built from bilingual examples, and the optimal translation of new sentences can be efficiently found by Dynamic Programming algorithms. This paper presents and discusses Grammar Association state of the art, including a new statistical model: Loco_C.

1 Introduction

Grammar Association is a promising technique for facing Machine Translation and Language Understanding tasks,[1] first proposed by Vidal, Pieraccini, and Levin (1993). This technique combines statistical and structural models, all of which can be automatically built from a set of bilingual sentence pairs. Moreover, the optimal translation of new input sentences can be efficiently found by Dynamic Programming algorithms.

Basically, a Grammar Association system consists of three models: (1) an *input grammar* modelling the input language of the translation task; (2) an *output grammar* modelling its output language; (3) an *association model* describing how the use of certain elements (rules) of the input grammar is related (in the translation task) to the use of their corresponding elements in the output grammar. Using these models, the system performs the translation of input sentences as follows: (1) first, the input sentence is parsed using the input grammar, giving rise to an input derivation; (2) given the input derivation, the association model assigns a weight to each rule of the output grammar; (3) in the (now weighted) output grammar, a search for the optimal output derivation is carried out; (4) the sentence associated to that derivation is conjectured as translation of the input sentence.

We are interested in designing Machine Translation systems based on the principles of Grammar Association and within a statistical framework. Some steps we have taken towards this final end are presented in this work.

2 Grammar Association into a statistical framework

In most of the papers describing statistical approaches to Machine Translation, Bayes' rule is applied giving rise to the following Fundamental Equation,

$$\begin{aligned} Y^\star(X) &= \arg\max_{Y \in L_\mathrm{o}} \Pr(Y \mid X) \\ &= \arg\max_{Y \in L_\mathrm{o}} \Pr(Y) \cdot \Pr(X \mid Y) \ , \end{aligned}$$

meaning that the optimal translation $Y^\star$ of an input sentence X, the most probable sentence Y in the output language L_o given $X \in L_\mathrm{i}$, can be found by maximizing the product of two factors:

- The *a priori* probability of the output sentence, $\Pr(Y)$. In practice, it is computed by

[1] We view Language Understanding as a particular case of Machine Translation where the output language is aimed at representing the meaning of input sentences.

using a statistical model of the output language L_o.

- The conditional probability $\Pr\left(X \mid Y\right)$ of the input sentence X, given the output one Y. In practice, it is computed by using a statistical model of the *reverse* translation process.

This decomposition has the advantage of modularity in the modelling. An *ad hoc* statistical language model encapsulates the features that are inherent to the output language, while the reverse translation model can focus on relations between input and output words, assigning scores to sentence pairs without taking into account if the output sentence is well-formed.[2] An alternative, direct statistical approach with a model for computing $\Pr\left(Y \mid X\right)$ seems to require this single model to be complex enough to assign high scores only to pairs where the output sentence verifies two conditions: it is well-formed and means the same that the input one. Hence, for the sake of simplified modelling, Bayes' decomposition has become a typical choice in Machine Translation.

However, in the Grammar Association context, when developing (using Bayes' decomposition) the basic equations of the system presented in (Vidal et al., 1993), it is said that the reverse model for $\Pr\left(X \mid Y\right)$ *"does not seem to admit a simple factorization which is also correct and convenient"*, so *"crude heuristics"* were adopted in the mathematical development of the expression to be maximized. We are going to show that, by means of a direct modelling, Grammar Association can be set into a rigorous statistical framework without renouncing a convenient factorization for the search of the optimal translation to be efficient. Moreover, the main advantage of Bayes' decomposition, modularity, is inherently present in Grammar Association systems: relations between input and output are mainly modelled by a (direct) statistical association model and structural features of the output language are modelled by a grammar, which restricts the search space for the best translation.

Let us begin assuming there are unambiguous grammars G_i and G_o describing, respectively, the input language L_i and the output one L_o. Thus, there is a one-to-one correspondence in each language relating sentences to their derivations and we can write

$$\Pr\left(Y \mid X\right) = \Pr\left(D_{G_\mathrm{o}}\left(Y\right) \mid D_{G_\mathrm{i}}\left(X\right)\right) ,$$

where $D_G\left(S\right)$ denotes the only derivation of sentence S in grammar G. Moreover, let us suppose the output grammar is context-free and rewriting probability of an output non-terminal using a certain rule is independent of which other output rules have been employed in the output derivation. Then, it follows that the probability of an output derivation D_o given an input one D_i can be expressed as

$$\Pr\left(D_\mathrm{o} \mid D_\mathrm{i}\right) = \prod_{r_\mathrm{o} \in D_\mathrm{o}} \Pr\left(r_\mathrm{o} \mid \mathit{left}\left(r_\mathrm{o}\right), D_\mathrm{i}\right) ,$$

with a term in the sum for each participation of a rule r_o in the derivation D_o, and $\mathit{left}\left(r_\mathrm{o}\right)$ denoting the left-hand side non-terminal of that rule. So, finally, we can find the most probable translation $Y^\star\left(X\right)$ of an input sentence X as the sentence associated to the output derivation given by

$$\operatorname*{arg\,max}_{D_\mathrm{o} \in \mathbf{D}\left(G_\mathrm{o}\right)} \prod_{r_\mathrm{o} \in D_\mathrm{o}} \Pr\left(r_\mathrm{o} \mid \mathit{left}\left(r_\mathrm{o}\right), D_{G_\mathrm{i}}\left(X\right)\right) ,$$

where $\mathbf{D}\left(G_\mathrm{o}\right)$ stands for the set of all possible derivations in G_o.

In practice, input and output grammars will be approximations inferred from samples and, more specifically, they will be acyclic finite-state automata. The restriction from context-free grammars to regular ones is due to the wide availability of inference techniques for these formal machines and to computational convenience. On the other hand, the output grammar has to be acyclic because of a more subtle point: the most probable derivation in the grammar will never make use of a cycle (no matter how high its probability is, avoiding the cycle always makes the derivation more probable). Hence, if we allowed the inference algorithm to model some features of the output language using cycles, system translations

[2] Note that model behaviour for syntactically incorrect input sentences is not important because input sentence is known and the search is just over the output language.

would never exhibit such features. Finally, for the sake of homogeneity, we choose to force input grammar to be acyclic too.

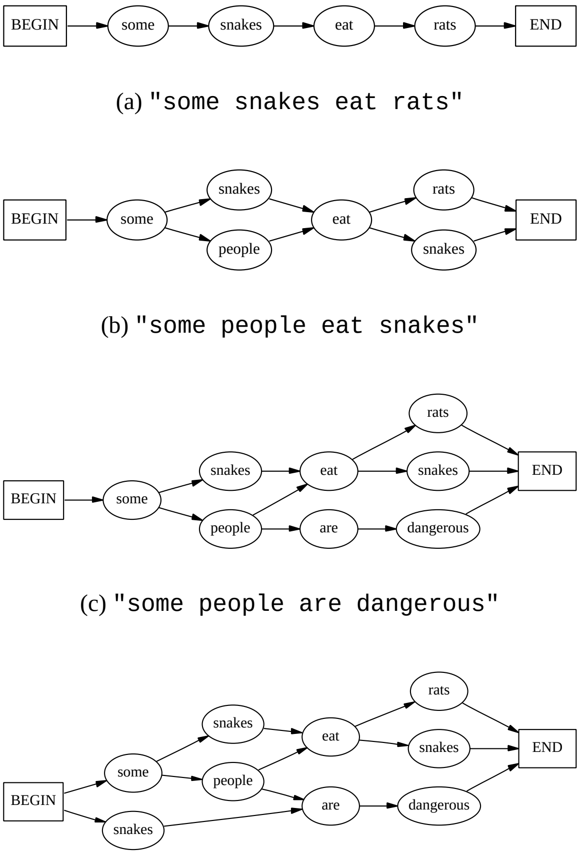
We can conclude this section saying that, inferring deterministic and acyclic finite-state automata, if we are able to learn association models for estimating, for each output rule, the probability of using that rule conditioned on having employed its left-hand side and the identity of the input derivation, then an efficient Dynamic Programming search for the optimal output derivation[3] can be used in order to provide the most probable translation.

3 Using ECGI language models

The ECGI algorithm (Rulot and Vidal, 1987) is a heuristic technique for the inference of acyclic finite-state automata from positive samples, and determinism can be imposed *a posteriori* by a well-known transformation for regular grammars. Therefore, in principle, ECGI provides exactly the kind of language model Grammar Association needs. Moreover, it was (without imposing determinism) the inference technique employed in (Vidal et al., 1993).

Informally, ECGI works as follows. With the first sample sentence, it builds an initial automaton consisting in a linear path representing the sentence. Words label states (instead of arcs) and there are two special non-labelled states: the initial one and the final one. For each new sentence, if it is already recognized by the automaton built so far, nothing happens; otherwise, if the current model does not recognize the sentence, new arcs and states are added to the most suitable path (according to a minimum-cost criterion) for recognition to be possible. In a sense, it is like constructing a new path for the new sentence and then finding a maximal merge with a path in the automaton.

For further discussion on some features of the ECGI algorithm, let us first consider the following set of five sentences: (1) "some snakes eat rats"; (2) "some people eat snakes"; (3) "some people eat rats"; (4) "some people are dangerous"; (5) "snakes are dangerous". Figure 1 shows how ECGI incrementally builds an automaton able to recognize the whole training set and, moreover, performs some generalizations. For instance, after considering the two first sentences (subfigure b), two more sentences are also represented in the current automaton: "some snakes eat snakes" and "some people eat rats". Thus, when this last sentence is actually presented to the algorithm, there is no need for the automaton to be updated. On the contrary, sentences 4 and 5 imply the addition of new elements and the finally inferred automaton is the one shown in subfigure d.



(a) "some snakes eat rats"

(b) "some people eat snakes"

(c) "some people are dangerous"

(d) "snakes are dangerous"

Figure 1: The ECGI algorithm: an example.

Though successful application of ECGI to a variety of tasks has been reported,[4] the method

[3] Obviously, any algorithm for finding the minimum-cost path in a graph is applicable.

[4] For instance, ECGI has been applied to problems as different as speech understanding (Prieto and Vidal, 1992), hand-written digit recognition (Vidal et al., 1995), and music composition (Cruz and Vidal, 1997)

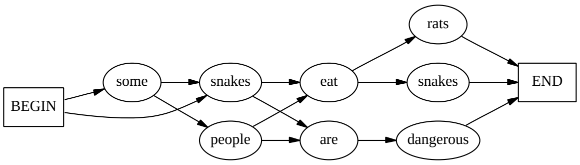


Figure 2: An alternative automaton.

suffers from some drawbacks. For instance, the level of generalization is sometimes lower than expected. In the example presented in Figure 1, when "snakes are dangerous" is employed for updating the model in subfigure c, instead of adding a new state and two arcs to the path corresponding to "some people are dangerous", the solution in Figure 2 seems to be an appealing alternative: adding just two arcs, more reasonable generalization is obtained. Nevertheless, ECGI chooses the solution in Figure 1 because it searches for *just one path* to be modified with a minimal number of new elements, and does not take into account combinations of different paths.

On the other hand, ECGI can suffer from inadequate generalization, especially at early stages of the incremental construction of the automaton. If "some people eat snakes" and "snakes are dangerous" were the first two sentences presented to ECGI, the algorithm would try to make use of the state "snakes" of the initial model for representing the occurrence of that word in the second sentence, leading to an automaton which would recognize “sentences” as "some people eat snakes are dangerous", or simply "snakes". The situation that produces this kind of undesired behaviour of the method is characterized by the confluence of a couple of circumstances: a word in a new sentence is also present in the current model, but with a different function, and that automaton has not enough adequate structural information for offering a better merging to the new sentence.

As pointed out by Prieto and Vidal (1992), a proper ordering of the set of sentences presented to ECGI can provide more compact models, and we think that better ones too. The ordering we propose here simply follows, first, a decreasing-length criterion and then, for breaking ties, applies any dictionary-like ordering. Thus, we try to avoid the problem discussed in the previous paragraph by providing the inference algorithm with as much as possible structural information at first stages of automaton construction and, moreover, dictionary-like ordering inside each length is aimed at frequently presenting to ECGI new sentences that are similar to the previous ones.

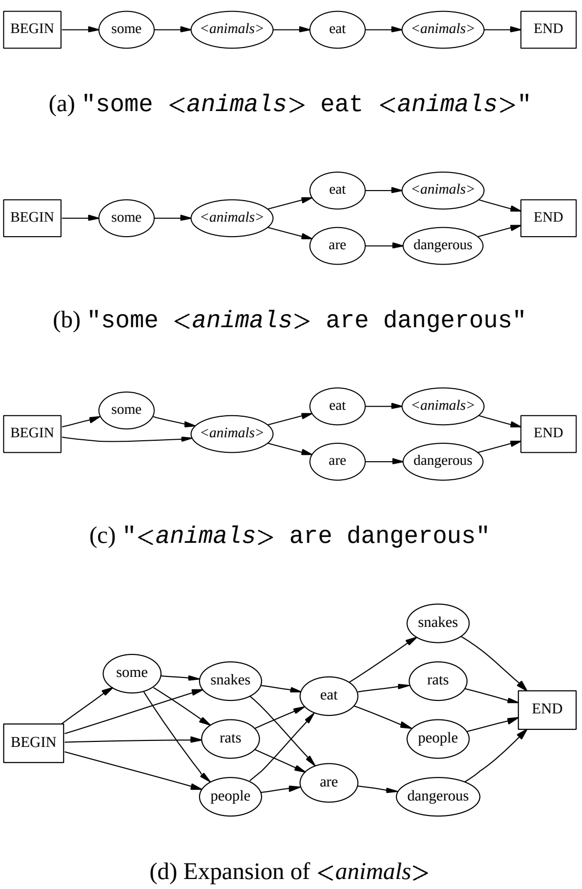
Furthermore, a very common way to reduce the complexity of problems involving languages is the definition of word categories, which can be manually designed or automatically extracted from data (Martin et al., 1995). We think categorization helps in solving the problem of undesired merges and also in increasing the generalization abilities of ECGI. In order to illustrate this point, let us consider a category <*animals*> consisting of words "snakes", "rats" and "people" in the very simple example of Figure 1. Words can be substituted for the appropriate category in the original sentences; then, the modified sentences are presented to the inference algorithm; finally, categories in the automaton are expanded. Figure 3 shows the automata that are successively built in that process.

As said at the beginning of this section, determinism must be imposed *a posteriori* for the language models to fit our formal framework. In addition, we will apply them a minimization process in order to simplify the problem that the corresponding association model will have to solve.

4 Loco_C: A new association model

Following a data-driven approach, a Grammar Association system needs to learn from examples an association model capable to estimate the probabilities required by our recently developed framework, that is, the probability of each rule in the grammar that models the output language, conditioned on its left-hand side and the derivation of the input sentence.

Among the different association models we have studied (Prat, 1998), it is worth emphasizing one we have specifically developed for playing that role in Grammar Association systems: the Loco_C model. We based our design on the IBM models 1 and 2 (Brown et al., 1993), but taking into account that our model must generate correct derivations in a given grammar, not any se-



(a) "some <*animals*> eat <*animals*>"

(b) "some <*animals*> are dangerous"

(c) "<*animals*> are dangerous"

(d) Expansion of <*animals*>

Figure 3: Using a category <*animals*> for "snakes", "rats" and "people" in the example of Figure 1.

quence of rules.[5] Moreover, we wanted to model the probability estimation for each output rule as an adequately weighted mixture,[6] along with keeping the maximum-likelihood re-estimation of its parameters within the growth transformation framework (Baum and Eagon, 1967; Gopalakrishnan et al., 1991). After exploring some similar alternatives (and discarding them because of their poor results in a few translation experiments), Loco_C was finally defined as explained below.[7]

The Loco_C model assumes a random generation process (of an output derivation, given an input one) which begins with the starting symbol of the output grammar as the "current sentential form" and then, while the current sentential form contains a non-terminal, iteratively performs the following sequence of two random choices: in *Choice 1*, one of the rules in the input derivation is chosen; in *Choice 2*, the non-terminal in the current sentential form is rewritten using a randomly chosen rule of the output grammar.

The behaviour of the model depends on two kinds of parameters, each one guiding one of the choices mentioned above. Formally, given an input derivation D_i and an output non-terminal n_o to be rewritten, the probability of an input rule $r_\mathrm{i} \in D_\mathrm{i}$ to be chosen in *Choice 1* depends on parameters of the form $\alpha\,(r_\mathrm{i} \mid n_\mathrm{o})$ and can be expressed as

$$\frac{\alpha\,(r_\mathrm{i} \mid n_\mathrm{o})}{\displaystyle\sum_{r'_\mathrm{i} \in D_\mathrm{i}} \alpha\left(r'_\mathrm{i} \mid n_\mathrm{o}\right)}\,.$$

On the other hand, once a particular input rule r_i is chosen, the probability of an output rule r_o whose left-hand side is n_o to be chosen in *Choice 2* is directly given by a parameter of the form $\rho\,(r_\mathrm{o} \mid r_\mathrm{i})$. Hence, $\Pr\,(r_\mathrm{o} \mid \mathit{left}\,(r_\mathrm{o})\,, D_\mathrm{i})$ takes in Loco_C the form

$$\sum_{r_\mathrm{i} \in D_\mathrm{i}} \frac{\alpha\,(r_\mathrm{i} \mid \mathit{left}\,(r_\mathrm{o}))}{\displaystyle\sum_{r'_\mathrm{i} \in D_\mathrm{i}} \alpha\left(r'_\mathrm{i} \mid \mathit{left}\,(r_\mathrm{o})\right)} \cdot \rho\,(r_\mathrm{o} \mid r_\mathrm{i})$$

of a weighted mixture depending on two kinds of trainable parameters:

- $\alpha\,(r_\mathrm{i} \mid n_\mathrm{o})$: Measures the importance of r_i in choosing an adequate rewriting rule for n_o.[8]

[5] In those simple IBM translation models, an output sequence (of words) is randomly generated from a given input one by first choosing its length and then, for each position in the output sequence, independently choosing an element (word). If the relation between input and output derivations (sequences of rules) has to be explicitly modelled, the choices of output elements can no longer be independent because a rule is only applicable if its left-hand side has just appeared in the output derivation.

[6] In IBM models, all words in the input sequence have the same influence in the random choice of output words (model 1) or they have a relative influence depending on their positions (model 2). In the case of derivations, we are interested in modelling those relative influences taking into account rule identities (instead of rule positions).

[7] Full details on the discarded models, Loco_1, Loco_A, and Loco_B, can be found (in Spanish) in pages 52–60 of (Prat, 1998).

[8] Note that learning these parameters performs a sort of "automatic variable selection" of the input rules that are relevant for discriminatively choosing among the next applicable output rules.

	MLA Task
Spanish:	"un círculo oscuro está encima de un círculo"
English:	"a dark circle is above a circle"
Spanish:	"se elimina el cuadrado oscuro que está debajo del círculo y del triángulo"
English:	"the dark square which is below the circle and the triangle is removed"
	Simplified Tourist Task
Spanish:	"nos vamos a ir el día diez a la una de la tarde."
English:	"we are leaving on the tenth at one in the afternoon."
Spanish:	"¿puedo pagar la cuenta con dinero en efectivo?"
English:	"can I pay the bill in cash?"

Figure 4: Examples of sentence pairs.

- $\rho\,(r_\mathrm{o} \mid r_\mathrm{i})$: Measures how much r_i agrees in using the rule r_o.

Consequently, the corresponding likelihood function is not polynomial, but rational, so Baum-Eagon inequality (1967) cannot be applied and Gopalakrishnan et al. inequality (1991) must be used, instead, in order to develop a Loco_C model re-estimation algorithm based on growth transformations. Fortunately, both the computational complexity of the resulting re-estimation algorithm (same order as with IBM model 1) and the experimental results are satisfactory.

5 Experimental results

In a first series of experiments, we were interested in knowing whether or not our proposals actually improve Grammar Association state of the art. To this end, a simple artificial Machine Translation task was employed. The corpus consists of pairs of sentences describing two-dimensional scenes with circles, squares and triangles in Spanish and English (some examples can be found in Figure 4, where the task is referred to as *MLA Task*). There are 29 words in the Spanish vocabulary and 25 in the English one.

Table 1: Results of an English-to-Spanish translation experiment with the original Grammar Association system, using 10,000 pairs of the *MLA Task* for training and 200 for testing.

Sentence Sorting	Minimum Deterministic	Length Constraint	Correct Translations
No	No	No	84.5%
No	No	Yes	88.0%
No	Yes	No	95.5%
No	Yes	Yes	97.0%
Yes	No	No	82.5%
Yes	No	Yes	87.0%
Yes	Yes	No	99.5%
Yes	Yes	Yes	99.5%

Let us begin considering English-to-Spanish translation, with 10,000 pairs for training the systems and 200 different ones for testing purposes. We carefully implemented the original Grammar Association system described in (Vidal et al., 1993), tuned empirically a couple of smoothing parameters, trained the models and, finally, obtained an 84.5% of correct translations.[9] Then, we studied the impact of: (1) sorting, as proposed in Section 3, the set of sentences presented to ECGI; (2) making language models deterministic and minimum; (3) constraining the best translation search to those sentences whose lengths have been seen, in the training set, related to the length of the input sentence. As shown in Table 1, all the proposed measures were beneficial and we got a final 99.5% of correct translations (that is, just one translation was wrong). Hence, we decided to apply those measures to all our Grammar Association systems and, in particular, to our Loco_C one. This system, after tuning some minor parameters (for instance, the number of re-estimation iterations for the model was fixed to 500), got a 99.0% of correct translations.

Then, in order to further compare our two systems (which will be referred to as IOGA, for Improved Original Grammar Association, and simply Loco_C) without more manual tuning, both were tested with 1,000 new sentence pairs: in this case, IOGA got a 99.4% and Loco_C got

[9] For each bilingual sentence pair (X, Y) employed for testing a system, we consider that the system achieves a *correct translation* only if it produces exactly the sentence Y as output when it is provided with the sentence X as input.

a 99.9%.

In a second series of experiments, we wanted to compare our best system, Loco_C, with ReConTra, the recurrent connectionist system described in (Castaño and Casacuberta, 1997), where a 98.4% of correct translations is reported on the Spanish-to-English *MLA Task* with just 3,000 pairs for training. In the same conditions, Loco_C got a 92.8% of correct translations on a 1,000 pair test set (IOGA, just an 81.6%).

Since the *MLA Task* is an artificial task where each language can be exactly modelled by an acyclic finite-state automaton, we decided to use those exact automata in our systems in order to measure the impact of perfect language modelling. In this case, Loco_C reached perfect results (100.0%), while IOGA got a 95.0%. As a conclusion to this second series of experiments, we can point out that our systems are quite sensitive to the quality of language models and, also, that Loco_C is a very good association model.

Our last series of experiments were carried out on a different, more complex task (but artificial too). It was extracted from the task defined for the first phase of the EuTrans project (Amengual et al., 1996) and covers just a small subset of the situations tourists can face when leaving hotels (some examples can be found in Figure 4, where the task is referred to as *Simplified Tourist Task*). There are 178 words in the Spanish vocabulary and 140 in the English one. We defined a standard scenario in which Spanish-to-English translation must be performed on 1,000 sentences after training the corresponding models with 5,000 pairs.

In that scenario, Loco_C achieved an 80.8% of correct translations, where errors are mainly due to lack of coverage in the language models, especially in the input one: only 85.7% of the Spanish sentences in the test set could be correctly parsed with the inferred model, so we decided to apply word categories to improve the generalization capabilities of ECGI as exemplified in Section 3. Using automatic categorization (Martin et al., 1995) for extracting 75 Spanish word classes and 50 English ones, the resulting language models achieved perfect coverage and the Loco_C system performance increased to 98.0%.

In order to put the previous figure into context, it is worth saying that the best result obtained by ReConTra in the same scenario was 91.1%. On the other hand, combining automatic bilingual categorization and Subsequential Transducers as described in (Barrachina and Vilar, 1999), a 98.4% of correct translations can be achieved for an adequate choice of the number of word classes (60), though only a 68.7% is obtained by the same system in the absence of categorization.

6 Concluding remarks

Our work presents a set of improvements on previous state of the art of Grammar Association: first, by providing better language models to the original system described in (Vidal et al., 1993); second, by setting the technique into a rigorous statistical framework, clarifying which kind of probabilities have to be estimated by association models; third, by developing a novel and especially adequate association model: Loco_C.

On the other hand, though experimental results are quite good, we find them particularly relevant for pointing out directions to follow for further improvement of the Grammar Association technique. One of these directions consists in exploring better language models, refining the categorization methods employed in this work or substituting ECGI for some kind of merge-based inference algorithm (Thollard et al., 2000). Exploiting data-driven bilingual categorization (Barrachina and Vilar, 1999) is another promising way to improve the performance of our system.

Finally, let us say that, obviously, the experimental results on simple artificial tasks presented in this work are not intended for convincing the reader that our Grammar Association systems could obtain similar performances on complex tasks as, for instance, *the Hansards* (the bilingual proceedings of the Canadian parliament). Our controlled experiments were mainly aimed at showing that our proposals improve Grammar Association, along with comparing this technique with a couple of different ones and providing easy-to-analyse results. For these simple purposes, we find our experimental work adequate. However, natural translation tasks should be faced soon, in the next stage of our research. This implies, for instance, trying to cope with severe data sparseness. In this regard, we are optimistic: on one hand, because we trust in bilingual categorization for reducing the negative effects of sparseness (Vilar et al., 1995); on the other hand, because some additional experiments carried out with Grammar Association systems on the Spanish-to-English *MLA Task* with just 500 pairs for training show acceptable results. For instance, our `Loco_C` achieved an 88.3% of correct translations[10] while, in the same scenario, ReConTra performance drops to 53.1% (Castaño and Casacuberta, 1997).

Acknowledgements

Most of the work presented here was carried out under the kind supervision of Dr. Francisco Casacuberta, and the author want to express his gratitude to him.

Furthermore, it is worth saying that this work has been partially supported by grant P1A99-10 from Fundació Caixa Castelló-Bancaixa (NEUROTRAD project).

References

J. C. Amengual, J. M. Benedí, A. Castaño, A. Marzal, F. Prat, E. Vidal, J. M. Vilar, C. Delogu, A. Di Carlo, H. Ney, and S. Vogel. 1996. Definition of a machine translation task and generation of corpora. Technical Report D1, EuTrans (IT-LTR-OS-20268).

S. Barrachina and J. M. Vilar. 1999. Bilingual clustering using monolingual algorithms. In *Procs. of the TMI'99*, pages 77–87.

L. 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.

P. F. Brown, S. A. Della Pietra, V. J. Della Pietra, and R. L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

M. A. Castaño and F. Casacuberta. 1997. A connectionist approach to machine translation. In *Procs. of the EuroSpeech'97*, volume 1, pages 91–94.

P. P. Cruz and E. Vidal. 1997. A study of grammatical inference algorithms in automatic music composition. In *Preprints of the VII National Symposium on Pattern Recognition and Image Analysis*, volume 1, pages 43–48.

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

S. Martin, J. Liermann, and H. Ney. 1995. Algorithms for bigram and trigram word clustering. In *Procs. of the EuroSpeech'95*.

F. Prat. 1998. *Traducción automática en dominios restringidos: Algunos modelos estocásticos susceptibles de ser aprendidos a partir de ejemplos*. Ph.D. thesis, Depto. de Sistemas Informáticos y Computación, Universidad Politécnica de Valencia (Spain).

N. Prieto and E. Vidal. 1992. Learning language models through the ECGI method. *Speech Communication*, 11(2–3):299–309.

H. Rulot and E. Vidal. 1987. Modelling (sub)string-length based constraints through a grammatical inference method. In *Pattern Recognition Theory and Applications*, volume F30 of *NATO ASI*, pages 451–459. Springer-Verlag.

F. Thollard, P. Dupont, and C. de la Higuera. 2000. Probabilistic DFA inference using Kullback-Leibler divergence and minimality. In *Procs. of the ICML'2000*, pages 975–982.

E. Vidal, R. Pieraccini, and E. Levin. 1993. Learning associations between grammars: A new approach to natural language understanding. In *Procs. of the EuroSpeech'93*, pages 1187–1190.

E. Vidal, H. Rulot, J. M. Valiente, and G. Andreu. 1995. Application of the error-correcting grammatical inference algorithm (ECGI) to planar shape recognition. Technical Report DSIC-II/2/95, Depto. de Sistemas Informáticos y Computación, Universidad Politécnica de Valencia (Spain).

J. M. Vilar, A. Marzal, and E. Vidal. 1995. Learning language translation in limited domains using finite-state models: Some extensions and improvements. In *Procs. of the EuroSpeech'95*, pages 1231–1234.

[10] In this experiment, 17 Spanish word classes and 15 English ones were automatically extracted from the training pairs in order to increase ECGI generalization capabilities.