Statistical Translation in Limited Domain Tasks *

Ismael García-Varea, Francisco Casacuberta
Instituto Tecnológico de Informática
Universidad Politécnica de Valencia
46071 Valencia, SPAIN
{fcn,ivarea}@iti.upv.es

Abstract

The statistical approach is an adequate framework for introducing automatic learning techniques in Machine Translation. One of the problems with Statistical Machine Translation is the design of efficient algorithms for translating a given input string. In this paper, we propose two algorithms in order to solve this problem, a Memory-Based algorithm and an Iterative DP-based Search. The results achieved in both approaches are very different and make us conclude that the memory-based technique used here is not a good solution for solving this problem. Both algorithms have been tested on the "Tourist Task" corpus, a limited domain task that was generated in a semi-automatic way. The best results obtained were a word-error rate of 0.52 and a sentence-error rate of 3.2, using the iterative-search algorithm.

1 Introduction

The statistical approach is an adequate framework for introducing automatic learning techniques in Machine Translation [2, 9, 14, 15].

Under this framework, given an input string $\mathbf{s}$ from $S^\star$ (S is a finite input alphabet), the *probabilistic translation* of $\mathbf{s}$ is an output string, $\hat{\mathbf{e}} \in E^\star$ (E is a finite output alphabet) such that

$$\hat{\mathbf{e}} = \arg\max_{\forall \mathbf{e} \in E^\star} \Pr(\mathbf{e}|\mathbf{s}). \tag{1}$$

Using Bayes' theorem, and taking into account that $\Pr(\mathbf{s})$ is not a function of $\mathbf{e}$,

$$\hat{\mathbf{e}} = \arg\max_{\forall \mathbf{e} \in E^\star} \Pr(\mathbf{s}|\mathbf{e})\Pr(\mathbf{e}). \tag{2}$$

Equation (2) is known as the **Fundamental Equation of Machine Translation** [3]. In Statistical Translation, the input string $\mathbf{s}$, which is to be translated, is interpreted as a distorted string of an original string $\mathbf{e}$ from $E^\star$ through a noisy channel. In this framework, $\Pr(\mathbf{e})$ represents the probability that the original string is produced, and $\Pr(\mathbf{s}|\mathbf{e})$ is the probability that the original string $\mathbf{e}$ is distorted in the observed string $\mathbf{s}$. In practice, an estimate of $\Pr(\mathbf{e})$ is performed from a *Language Model* and an estimate of $\Pr(\mathbf{s}|\mathbf{e})$ is performed from a *Translation Model*. In this paper we are going to attack the translation problem using the paradigms given by equations (1) and (2). Among the reasons reported by Brown et al. [3] for using (2) instead (1), it can be observed that in (2), good output Language Models can aid the process of searching which allows for focusing on the *well-formed* output strings.

Some interesting Translation Models were proposed in [3] and in [14]. With the model proposed in [14], a Dynamic Programming algorithm can be designed to solve (2) [10, 11]. However, the corresponding algorithms for the models 1 to 5 in [3] are based on a certain type of the $A^\star$ algorithm [2, 15]. Recently a new approach was proposed in [6] that was based on a single

Work partially supported by the Spanish CICYT under contract EXTRA (TIC97/0745-C02) and EUTRANS (ESPRIT LTR-30268).

Dynamic Programming-like algorithm which computes approximate solutions when the IBM-Model 2 from [3] was used. This approach can be improved through an iterative process in which, a Dynamic Programming-like technique similar to the one proposed in [6] is used to improve succesive solutions.

2 A Statistical Model for Machine Translation

The Translation Models introduced in [3] are based on the concept of alignment between the components of the *translation pairs* $(\mathbf{s},\mathbf{e}) \in S^\star \times E^\star$.

Formally, an alignment is a mapping between the sets of positions in $\mathbf{s}$ and $\mathbf{e}$: $\mathbf{a} \subset \{1,...,|\mathbf{s}|\} \times \{1,...,|\mathbf{e}|\}$. However, in [3], the concept of alignment is restricted to being a function $\mathbf{a}$: $\{1,...,|\mathbf{s}|\} \rightarrow \{0,...,|\mathbf{e}|\}$, where $a_j = 0$ means that the position j in $\mathbf{s}$ is not aligned with any position of $\mathbf{e}$. All the possible alignments between $\mathbf{e}$ and $\mathbf{s}$ are denoted by $\mathcal{A}(\mathbf{e},\mathbf{s})$ and the probability of translating a given $\mathbf{e}$ into $\mathbf{s}$ by an alignment is denoted by $\Pr(\mathbf{s},\mathbf{a}|\mathbf{e})$, therefore

$$\Pr(\mathbf{s}|\mathbf{e}) = \sum_{\mathbf{a} \in \mathcal{A}(\mathbf{s},\mathbf{e})} \Pr(\mathbf{s},\mathbf{a}|\mathbf{e}). \tag{3}$$

The first model for $\Pr(\mathbf{s},\mathbf{a}|\mathbf{e})$ proposed in [3] (Model 1) is

$$\Pr_{M1}(\mathbf{s},\mathbf{a}|\mathbf{e}) = \frac{\epsilon}{(|\mathbf{e}|+1)^{|\mathbf{s}|}} \prod_{j=1}^{|\mathbf{s}|} t(s_j \mid e_{a_j}), \tag{4}$$

where ϵ is a positive constant, $t(s_j \mid e_i)$ is the *translation probability* of the input word s_j given the output word e_i. If equation (4) is used in (3),

$$\Pr_{M1}(\mathbf{s}|\mathbf{e}) = \frac{\epsilon}{(|\mathbf{e}|+1)^{|\mathbf{s}|}} \prod_{j=1}^{|s|} \sum_{i=0}^{|e|} t(s_j \mid e_i). \tag{5}$$

The second model for $\Pr(\mathbf{s},\mathbf{a}|\mathbf{e})$ proposed in [3] (Model 2) is

$$\Pr_{M2}(\mathbf{s},\mathbf{a}|\mathbf{e}) = \epsilon \prod_{j=1}^{|\mathbf{s}|} t(s_j \mid e_{a_j}) \alpha(a_j|j;|\mathbf{s}|,|\mathbf{e}|), \tag{6}$$

where ϵ is a positive constant, $t(s_j|e_i)$ is the translation probability of s_j given e_i as in the first model, and $\alpha(a_j|j;|\mathbf{s}|,|\mathbf{e}|)$ is the *alignment probability*. This distribution gives us the alignment probability of the i-th word in the target sentence , given any position in the source sentence and the length of both sentences. If equation (6) is used in (3), we have,

$$\Pr_{M2}(\mathbf{s}|\mathbf{e}) = \epsilon \prod_{j=1}^{|\mathbf{s}|} \sum_{i=0}^{|\mathbf{e}|} t(s_j \mid e_i).\alpha(i|j;|\mathbf{s}|,|\mathbf{e}|) \tag{7}$$

Given training data as a set $\mathcal{T} = \{(\mathbf{s}^{(1)},\mathbf{e}^{(1)}),(\mathbf{s}^{(2)},\mathbf{e}^{(2)}),\cdots,(\mathbf{s}^{(K)},\mathbf{e}^{(K)})\}$ the estimation of the translation probabilities and the alignment probabilities for both models can be performed by using the transformations proposed in [3]. These transformations allow us to increase the product of (7) for all training pairs, and can achieve the global maximum for (5).

3 Translation Algorithms

In this section, we are going to describe two algorithms for obtaining the translation of a sentence from a given input language to another output language. The first algorithm proposed here is a *memory-based algorithm* which attempts to solve equation (1). The second one is an *iterative dynamic progrogramming-based search algorithm* which attempts to solve equation (2), by using an iterative process of solution refinement.

3.1 The Memory-Based Algorithm

Based on the information given by the above translation model (7) and given an input sentence, here our goal is to obtain a translation sentence in an output language. For this reason, we are going to work directly with the $\Pr(\mathbf{e}|\mathbf{s})$ probability. Thus, in this case, the equation (7) will be:

$$\Pr(\mathbf{e}|\mathbf{s}) = \epsilon \prod_{i=1}^{|\mathbf{e}|} \sum_{j=0}^{|\mathbf{e}|} t(e_i \mid s_j)\alpha(j \mid i; |\mathbf{e}|, |\mathbf{s}|) \tag{8}$$

A sentence training set $\mathcal{T}$ is used to compute the parameters of equation (8), $t(e_i \mid s_j)$ and $a(j \mid i, |\mathbf{e}|, |\mathbf{s}|)$ as in [3].

This algorithm works as follows: Given an input string $\mathbf{s}$, we calculate the equation (8) for each $\{\mathbf{e}^k\}_{1\leq k\leq K}$ sentence of the training corpus. We choose the N ($n \leq N$) pairs $\{(\mathbf{s'}^n,\mathbf{e'}^n)\}_{1\leq n\leq N}$ from $\mathcal{T}$ with the highest probability $\Pr_{M2}(\mathbf{s}|\mathbf{e'}^n)$. A hypothesis of the translation of $\mathbf{s}$ is computed for each n ($1 \leq n \leq N$) $\mathbf{e'}^n$ as is pointed out in the following algorithm.

MEMORY-BASED ALGORITHM

INPUT

- The translation probabilities t.
- The alignment probabilities α.
- An input string $\mathbf{s} \in S^\star$.

OUTPUT

- $\arg\max_{\forall \mathbf{s}\in S} \Pr(\mathbf{e}|\ \mathbf{s})$

METHOD

1. Compute $\Pr(\mathbf{e}^k \mid \mathbf{s})$, $1 \leq k \leq K$
2. Choose the N-best hypothesis $\Pr(\mathbf{e'}^1 \mid \mathbf{s}) \geq \Pr(\mathbf{e'}^2 \mid \mathbf{s}) \geq \cdots \geq \Pr(\mathbf{e'}^N \mid \mathbf{s})$
3. Let $\mathbf{s'}^i$ be $(1 \leq i \leq N)$, the corresponding couple of $\mathbf{e'}^i$ in $\mathcal{T}$.
   Compute the Levenstein alignment [8] between $\mathbf{s'}^i$ y $\mathbf{s}$:
   If an aligned pair s'^i_j and s_l ($s'^i_j \neq s_l$), look for the word $\mathbf{e'}^i$ which is aligned with s'^i_j and substitute $\mathbf{e'}^i$ for
   $$\arg\max_{\forall e\in E} t(e \mid s_l)$$
   so that you have new strings $\mathbf{e'}^i$.
4. Compute $\Pr(\mathbf{e''}^i \mid \mathbf{s})$ $1 \leq i \leq N$, and choose:
   $$\arg\max_{1\leq i\leq N} \Pr(\mathbf{e''}^i \mid \mathbf{s})$$

A variant of this algorithm is to directly use the information that the translation model provides, instead of the edition distance in order to "correct" the mistranslated words. This means using the multiple alignment between the input sentence and its corresponding best translation in the corpus, and its corresponding source in the corpus (see Figures 1 and 2). With this variant, we can view the translation process as an alignment process between very similar words in the corpus, and their associated translation.

There is an alignment in $\mathcal{A}(\mathbf{e}, \mathbf{s})$ for which $\Pr(\mathbf{a}|\mathbf{e},\mathbf{s})$is greatest. Brown et al. call this *the Viterbi alignment* and denote it by $V(\mathbf{s} \mid \mathbf{e})$ [3]. For IBM Model 2 (and, thus, also for Model 1) finding $V(\mathbf{s} \mid \mathbf{e})$ is straigthforward. For each j, we simply choose a_j so as to make the product $t(s_j \mid e_{a_j})\alpha(a_j \mid j; |\mathbf{s}|, |\mathbf{e}|)$ as large as posible. In an algorithmic way, we can see that as:

for $j = 1 \ldots |\mathbf{s}|$ **do**
  $V(j) = \arg\max_{0 \le i \le |\mathbf{e}|} \quad t(s_j \mid e_i)\alpha(i \mid j; |\mathbf{s}|, |\mathbf{e}|)$
end_for_j

This algorithm involves the training process in both directions ($\mathbf{s} \rightarrow \mathbf{e}$ and $\mathbf{e} \rightarrow \mathbf{s}$). We need to do the training in the first direction in order to obtain the alignment of the input sentence and its best translation in the corpus. The training in the second direction is needed in order to obtain the alignment between this best translation and its corresponding source. This double training will provide some relevant information about the best translated sentence obtained. It will help to decide which words must be modified and also how to do it (step 3 of the algorithm).

3.2 The Iterative-Search Algorithm

The second algorithm is an *iterative-search algorithm* which attempts to solve equation (2), as in the approach proposed in [6], but based on an iterative process of solution refinement in which a better solution can be built from the solution achieved in a previous iteration by using a Dynamic Programming-like technique.

In a more formal way, the problem here is to design an efficient algorithm for searching $\hat{\mathbf{e}}$ (or an approximation to $\hat{\mathbf{e}}$).

From (2) we derive

$$\max_{\mathbf{e} \in E^\star} \Pr(\mathbf{e})\Pr(\mathbf{s}|\mathbf{e}) = \max_{I} \max_{\mathbf{e}_1^I \in E^I} .\Pr(\mathbf{e}_1^I)\Pr(\mathbf{s}_1^{|\mathbf{s}|} \mid \mathbf{e}_1^I) \tag{9}$$

In other words, the maximisation in (2) can be performed by searching the best output string $\mathbf{e}_1^I$ for each possible I, and then by searching the optimal I.

Let us suppose that the length of the output string is known. The Translation Model used is (7) and, the Languaje Model will be a stochastic regular grammar, given by $G_R = (N, E, R, q_0, p)$, where: N is the set of non-terminals, E is the output alphabet, R is the set of rules like $q \rightarrow eq'$ or $q \rightarrow e$ (we assume that a state $F \in N$ exist in order to allow this last rule to be rewritten by $q \rightarrow eF$), q_0 is the first symbol of the grammar and p is a probabilistic function like $p : R \rightarrow]0, 1]$ such that $\forall q \in N$:

$$\sum_{e \in E; q' \in N} p(q \rightarrow eq') = 1$$

For the sake of simplicity we will use $p(q_{i-1}, e_i, q_i)$ instead of $p(q_{i-1} \rightarrow e_i q_i)$, where q_i is the state reached when the symbol e_i is produced, beginning in the state where e_{i-1} was produced, and so on.

From (7) and (9) we can obtain:

$$\max_{\forall \mathbf{e}_1^I \in E^I; \forall q_1^I \in N^I} \left(Pr_{G_R}(\mathbf{e}_1^I) Pr(\mathbf{s}_1^{|\mathbf{s}|} \mid \mathbf{e}_1^I) \right) = \tag{10}$$

$$\max_{\forall \mathbf{e}_1^I \in E^I; \forall q_1^I \in N^I} \left(p(q_0, e_1, q_1) \prod_{i=2}^{I} p(q_{i-1}, e_i, q_i) \prod_{j=1}^{|\mathbf{s}|} \sum_{i=0}^{I} t(s_j|e_i) \alpha(i|j; |\mathbf{s}|, I) \right)$$

The formal way to calculate this maximisation can be found in [4] or in [7] by using the following algorithm.

ALGORITHM ITERATIVE-SEARCH

INPUTS

- The translation probabilities t.
- The alignment probabilities α.
- An output language model Pr_{G_R}.
- An input string $\mathbf{s} \in S^\star$.

OUTPUT

- $\arg\max_{\forall e \in E} \left(\Pr_{G_R}(\mathbf{e}) \Pr_{M2}(\mathbf{s}|\mathbf{e}) \right)$

METHOD

- initialisation

  Compute a first approach ($\hat{\mathbf{e}}$) to the solution by using the algorithm proposed in [6].

- iteration

  While not convergence **do**

  Compute a new approach ($\hat{\mathbf{e}}$) to the solution by using a prefix that is built during the current iteration and a suffix that was computed in the previous iteration.

  end of While

END-OF-ALGORITHM

The length of the output sentence is set statistically around the mean of the output lengths for each length of the input sentence.

The computational time complexity of each iteration is $O(|\mathbf{s}| \times \mathbf{I_{max}} \times \mathbf{n_I} \times |\mathbf{E}|)$, where $\mathbf{I_{max}}$ is the maximum output length allowed and $\mathbf{n_I}$ is the number of output lengths tested.

4 Experiments and Results

We selected a subtask of the general "Traveller Task" [13] to perform experiments with the algorithms proposed here. The general domain of the task was a visit by a tourist to a foreign country. This domain included a great variety of different scenarios, from limited-domain applications to unrestricted natural language. The task used for the experiments reported here corresponded to a scenario of human-to-human communication situations at the reception desk of a hotel. This task provided a small "seed corpus" from which a large set of sentence pairs was generated in a semi-automatic way [1]. From the different pairs of languages that were generated, only Spanish to English was considered for this work. The parallel corpus consisted

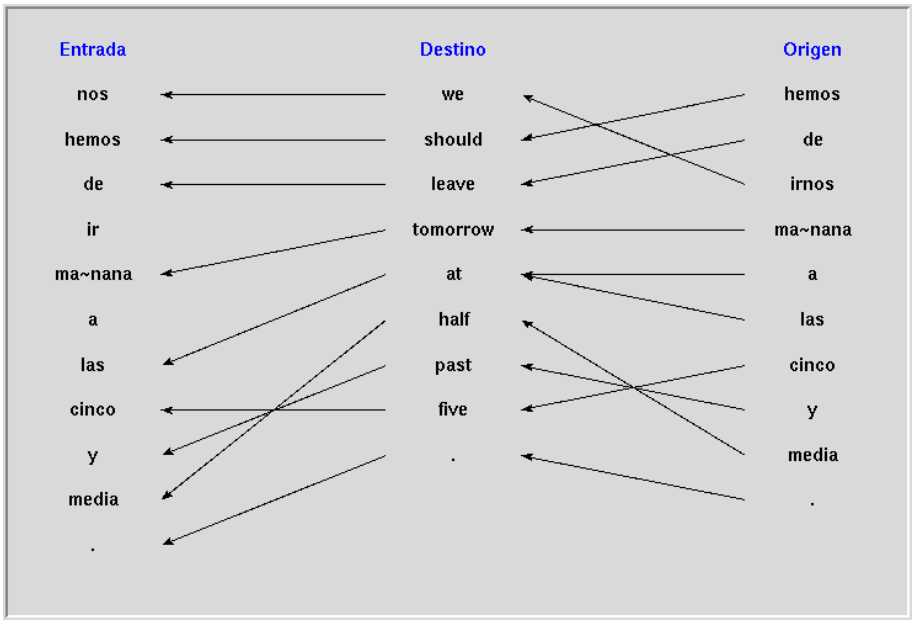

Figure 1: Multiple alignment example from *Source* to *Input*. The direction of the arrows shows the way that the *Input*(left) is best aligned with the (*Source*, *Target*)(middle,right) pair of the corpus. This alignment direction and the opposite direction give us the information used in the translation process.

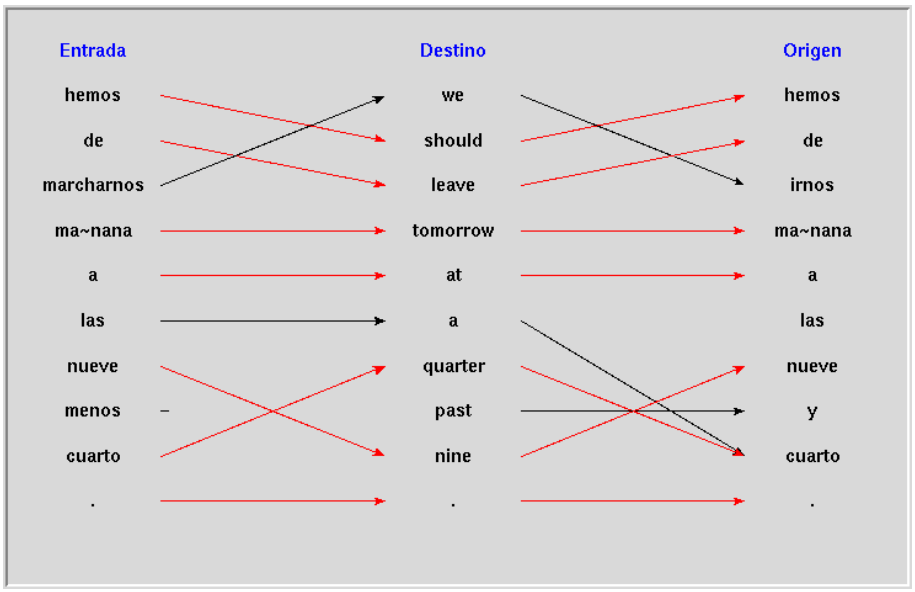

Figure 2: Multiple alignment example from *Input* to *Source*. The direction sense of the arrows shows the way that the *Input* is best aligned with the (*Source*, *Target*) pair of the corpus.

Spanish:	¿podría bajarme la bolsa de viaje a la *ocho treinta*?
English:	could you send down my travel bag to room number *eight three oh* ?
Spanish:	hemos de marcharnos el martes *veintiuno* a las *doce y media* .
English:	we should leave on Tuesday *the twenty-first* at *half past twelve*.
Spanish:	por favor , ¿ cuánto cuesta una habitación individual?
English:	how much does a single room cost , please ?
Spanish:	¿les importaría subir mis maletas al taxi?
English:	would you mind putting my suitcases in the taxi?
Spanish:	por favor , tengo hecha una reserva a nombre de *Beatriz Valls*.
English:	I have made a reservation for *Beatriz Valls*.
Spanish:	hemos reservado una habitación para *siete* días.
English:	we have booked a double room for *seven* days.

Table 1: Spanish-English sentences from the Traveller Task. The parts of the corpus to be categorized are in italics.

Error-Rate Percentage		
Categories	**Word Level**	**Sentence Level**
NO	72.24 %	98.4 %
YES	13.67 %	83.1 %

Table 2: Translation results using the Memory-based Algorithm. Word-Error Rate and Sentence-Error Rate for 1,000 test sentences.

of 65,000 sentence pairs (24,160 different sentence pairs). The input and output vocabulary sizes were 178 and 140, and the average input and output sentence lengths were 9.4 and 8.7, respectively.

From this corpus, a sub-corpus of 5,000 random sentence pairs was selected for training purposes. Testing was carried out with 1,000 input random sentences which were generated independently from the training set.

Under these circumstances two different experiments were done. Both of them used the task described above, the first one in its original form, and the other one categorizing its critical parts, i.e. the proper names, dates, hours and numbers.

The output language model was a Stochastic Regular Grammar built by the ECGI algorithm [12]. The output test-set perplexity of the inferred ECGI grammar was 3.53.

The results achieved with the Memory-based Algorithm are shown in Table 2, with and without categories. In Figure (1) and Figure (2) we can see two examples of the multiple alignment between an input sentence and the sentence-pair chosen in the corpus.

The results achieved with the Iterative-search Algorithm are shown in Table 3, with and

Error-Rate Percentage		
Categories	**Word Level**	**Sentence Level**
NO	2.06 %	15.1 %
YES	0.57 %	3.2 %

Table 3: Translation results using the Iterative-Search Algorithm. Word-Error Rate and Sentence-Error Rate for 1,000 test sentences.

without categories. The number of iterations used in this algorithm was three.

The results with both algorithms in the categorized case are shown without resolving the categorization.

5 Conclusions

Two algorithms for translating input sentences have been explained here. In the first algorithm we have only used the distribution probabilities obtained with the training of the translation model. In the second one, we have also used a stochastic regular grammar.

The main conclusions that can be drawn are:

- Good results can be achieved with the *iterative-search algorithm* without too high a computational cost. The best results achieved with this algorithm were a word-error rate of 0.57% and a sentence-error rate of 3.2%. Nevertheless, the statistical approach to *memory-based algorithms* produced bad results. One of the possible reasons could be the reason reported by Brown et al. [3] for using equation (2) instead of equation (1). It can be observed that, in this case, good output language models can aid the process of searching which allows focusing on the *well-formed* output strings.

- Comparing Tables 2 and 3, we find best results when the categorized corpus has been used. This is what we expected since the complexity of the corpus decreases when categorization is done.

References

[1] 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. *Final Report, Part I. ESPRIT project No. 20268 EUTRANS*

[2] P. Brown, J. Cocke, S. Della Pietra, V. Della Pietra, J. Jelinek, J. Lafferty, R. Mercer and P. Roosssina. 1990. A Statistical Approach to Machine Translation. In *Computational Linguistics*, 16:79–85.

[3] P. Brown, S. Della Pietras, V. Della Pietra and R. Mercer. 1993. The Mathematics of Statistical Machine Translation: Parameter Estimation. In *Computational Linguistics*, 19:263–310.

[4] F. Casacuberta, I. García-Varea, H. Ney. 1997. An Iterative Search Algorithm for Statistical Translation. *Technical report* No. DSIC-II/44/97, Dpto. Sistema Informáticos y Computación, Universidad Politécnica de Valencia.

[5] A. Castellanos, I. Galiano, E. Vidal. 1994. Application of OSTIA to Machine Translation Tasks. In *Grammatical Inference and Applications*. Lecture Notes in Computer Science-Lecture Notes in Artificial Intelligence. Vol. 862. pp. 93-105. Eds. R.C. Carrasco y J. Oncina. Springer-Verlag.

[6] I. García-Varea and F. Casacuberta. 1997. A Search Procedure for Statistical Translation. In *Proceedings of the VII National Symposium on Pattern Recognition and Image Analysis.* pp 199–204. Barcelona Spain.

[7] I. García-Varea, F. Casacuberta and H. Ney. 1998. An Iterative, DP-Based Search for Statistical Machine Translation. To *be submitted to the ICSLP'98 (5th International Conference on Spoken Language Processing).*

[8] V.I. Levenstein. 1965. Binary codes capable of correcting deletions, insertions, and reversals. (Russian) *Doklady Akademii nauk SSSR*, Vol. 163, No. 4, pp.854–848 (also *Cybernetics and Control Theory*, Vol. 10, No. 8, pp.707–710, 1966).

[9] I. D. Melamed. 1997. A Word-to-Word Model of Translational Equivalence. In *Proceedings of the ACL97*. pp 490–497. Madrid Spain.

[10] C. Tillmann, S. Vogel, H. Ney and A. Zubiaga. 1997. A DP based Search Using Monotone Alignements in Statistical Translation. In *Proceedings of the ACL97*. pp 289–296. Madrid Spain.

[11] C. Tillmann, S. Vogel, H. Ney and A. Zubiaga. 1997. Accelerated DP based Search for Statistical Translation. In *Proceedings of the EuroSpeech97*. Vol. 5, pp 2667–2670. Madrid Spain.

[12] E. Vidal and N. Prieto. 1992. Learning Language Models through the ECGI method. In *Speech Communication*, 11:299–309.

[13] E. Vidal. 1997. Finite-State Speech-to-Speech Translation. In *Proceedings of the International Conference on Acoustic Speech, and Signal Processing.*

[14] S. Vogel, H. Ney and C. Tillmann. 1996. HMM-Based Word Alignment in Statistical Translation. In *Proceedings of the Cooling96.*

[15] Y. Wang and A. Waibel 1997. Decoding Algorithm in Statistical Machine Translation. In *Proceedings of the ACL97*. pp 366–372. Madrid Spain.