Phrase-based finite state models [*]

Jorge González and Francisco Casacuberta

Departamento de Sistemas Informáticos y Computación
Universidad Politécnica de Valencia

Abstract. In the last years, statistical machine translation has already demonstrated its usefulness within a wide variety of translation applications. In this line, phrase-based alignment models have become the reference to follow in order to build competitive systems. Finite state models are always an interesting framework because there are well-known efficient algorithms for their representation and manipulation. This document is a contribution to the evolution of finite state models towards a phrase-based approach. The inference of stochastic transducers that are based on bilingual phrases is carefully analysed from a finite state point of view. Indeed, the algorithmic phenomena that have to be taken into account in order to deal with such phrase-based finite state models when in decoding time are also in-depth detailed.

1 Introduction

Machine Translation (MT) is an emerging area of research in computational linguistics which investigates the use of computer software to translate text or speech from one natural language to another. The goal of MT is very ambitious because it would allow for a reduction of the linguistic barriers which all the people have been ever involved with.

Statistical machine translation represents an interesting framework because the translation software being developed is language-independent, that is, different MT systems are built if different parallel training corpora are supplied.

Given a source sentence $\mathbf{s} = \mathbf{s}_1 \dots \mathbf{s}_J$, the goal of statistical machine translation is to find a target sentence $\hat{\mathbf{t}} = \mathbf{t}_1 \dots \mathbf{t}_{\hat{I}}$, among all possible target strings $\mathbf{t}$, that maximises the posterior probability, according to a source-channel model:

$$\hat{\mathbf{t}} = \operatorname{argmax}_{\mathbf{t}} \Pr(\mathbf{t}|\mathbf{s}) \tag{1}$$

Source-channel models are often applied the Bayes rule [1] to break them down into two different statistical models: a translation model to learn translations, and a language model, to score the quality of the proposed hypotheses [2, 3]:

$$\hat{\mathbf{t}} = \operatorname{argmax}_{\mathbf{t}} \Pr(\mathbf{s}|\mathbf{t}) \cdot \Pr(\mathbf{t}) \tag{2}$$

[*] This work is supported by the EC (FEDER) and the Spanish MEC under grant TIN2006-15694-C02-01.

The conditional probability $\Pr(\mathbf{t}|\mathbf{s})$ can also be approximated by a joint probability distribution $\Pr(\mathbf{s}, \mathbf{t})$ in order to be modelled by means of stochastic finite state transducers [4, 5]:

$$\hat{\mathbf{t}} = \text{argmax}_{\mathbf{t}} \Pr(\mathbf{s}, \mathbf{t}) \tag{3}$$

These models can integrate the probabilistic information that *state-of-the-art* phrase-based models [6–9] are used to explicitly separate into two distributions, that is, a target language model and a phrase translation dictionary.

This paper presents a natural evolution for finite state models in order to be based on bilingual phrases. Training and decoding algorithms are conveniently adapted to deal with such phrase-based finite state models. The main contributions are reflected on the translation results, which are clearly favourable to these phrase-based models, with respect to the original word-based approaches.

The organization of this document is as follows: next section presents a review of finite state models; sections 3 and 4 deal with, respectively, word-based and phrase-based finite state models; the experimental setup and results are described in section 5; and, finally, conclusions are summed up at the last section.

2 Finite state models

A weighted finite-state automaton is a tuple $\mathcal{A} = (\Gamma, Q, i, f, P)$, where Γ is an alphabet of symbols, Q is a finite set of states, functions $i : Q \rightarrow \mathbb{R}^+$ and $f : Q \rightarrow \mathbb{R}^+$ give a weight to the possibility of each state to be, respectively, initial and final, and parcial function $P : Q \times \{\Gamma \cup \lambda\} \times Q \rightarrow \mathbb{R}^+$ defines a set of transitions between pairs of states in such a way that each transition is labelled with a symbol from Γ (or the empty string λ), and is assigned a weight. An example of a weighted finite-state automaton can be observed in figure 1.

A weighted finite-state transducer [10] is defined similarly to a weighted finite-state automaton, with the difference that transitions between states are labelled with pairs of symbols that belong to the cartesian product of two different (input and output) alphabets, $(\Sigma \cup \{\lambda\}) \times (\Delta \cup \{\lambda\})$.

When weights are probabilities, the range of functions i, f, and P is constrained to $[0, 1]$. Moreover, probabilistic models have to respect the *consistency* property in order to define a distribution of probabilities on the free monoid. In that case they are called stochastic finite-state models. Consistent probability distributions can be obtained by requiring a series of local constraints, that is:

- $\sum i(q) = 1$
- $\forall q \in Q : \sum P(q, u, q') + f(q) = 1$

Then, given some input/output strings $\mathbf{s}$ and $\mathbf{t}$, a stochastic finite-state transducer is able to associate them a joint probability $\Pr(\mathbf{s}, \mathbf{t})$.

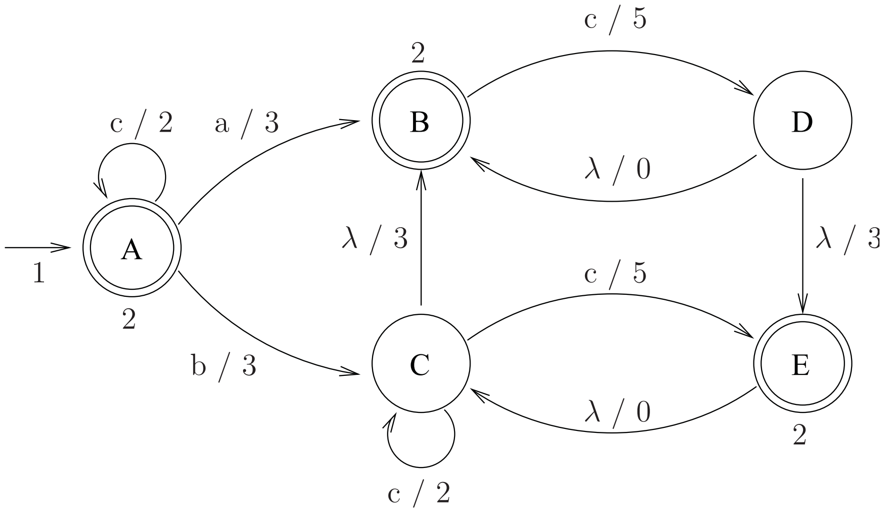

Fig. 1. A weighted finite-state automaton

2.1 Inference of stochastic transducers

The GIATI paradigm [11] has been revealed as an interesting approach to infer stochastic finite-state transducers through the modelling of languages. Rather than learning translations, GIATI first converts every pair of parallel sentences from the training corpus into only one string in order to, afterwards, infer a language model from.

More concretely, given a parallel corpus consisting of a finite sample C of string pairs: first, each training pair $(\bar{x}, \bar{y}) \in \Sigma^\star \times \Delta^\star$ is transformed into a string $\bar{z} \in \Gamma^\star$ from an extended alphabet, yielding a string corpus S; then, a stochastic finite-state automaton $\mathcal{A}$ is inferred from S; finally, transition labels in $\mathcal{A}$ are turned back into pairs of strings of source/target symbols in $\Sigma^\star \times \Delta^\star$, thus converting the automaton $\mathcal{A}$ into a transducer $\mathcal{T}$.

The first transformation is modelled by some labelling function $\mathcal{L} : \Sigma^\star \times \Delta^\star \rightarrow \Gamma^\star$, whereas the last transformation is defined by an inverse labelling function $\Lambda(\cdot)$, such that $\Lambda(\mathcal{L}(C)) = C$. Building a corpus of extended symbols from the original bilingual corpus allows for the use of many useful algorithms for learning stochastic finite-state automata (or equivalent models) that have been proposed in the literature about grammatical inference.

Every extended symbol from Γ has to condense somehow the meaningful relationship that exists between the words in the input and output sentences. Discovering these relations is a problem that has been thoroughly studied in statistical machine translation and has well-established techniques for dealing with it. The concept of statistical alignment [1] formalises this problem. An alignment is a mapping between words from a source sentence and words from a target sentence. Whether this function is constrained to a one-to-one, a one-to-many or a many-to-many correspondence depends on the particular assumptions that we make. Constraining the alignment function simplifies the learning procedure but causes the model to lessen its expressive power. The available algorithms try to find a trade-off between complexity and expressiveness.

2.2 The search problem

Equation 3 expresses the MT problem in terms of a finite state model that is able to compute the expression $\Pr(\mathbf{s}, \mathbf{t})$. Given that only the input sentence is known, the model has to be parsed, taking into account all possible $\mathbf{t}$ that are compatible with $\mathbf{s}$. The best output hypothesis $\hat{\mathbf{t}}$ would be that one which corresponds to a path through the transduction model that, with the highest probability, accepts the input sequence as part of the input language of the transducer.

Although the navigation through the model is constrained by the input sentence, the search space can be extremely large. As a consequence, only the most scored partial hypotheses are being considered as possible candidates to become the solution. This search process is very efficiently carried out by the well known Viterbi algorithm [12].

3 Word-based finite state models

As it has been already mentioned, the inference of transducers will be done through the transformation of the bilingual training corpus into a corpus of strings, which a language model will be inferred from. This transformation will be based on the alignment function defined between every pair of bilingual sentences. According to the alignment degree, these transducers could be classified as word-based or phrase-based finite state models.

One-to-one and one-to-many alignment functions would produce word-based models, whereas many-to-many correspondences would bring to phrase-based models.

On the one hand, one-to-one models do not seem a very appropriate approach since they would require that source-target aligned sentences had exactly the same number of words. On the other hand, one-to-many alignment models have been a reference in statistical machine translation until the phrase-based tendencies took place at the research community. Word-based models constrain alignments so that one target word has to be aligned to only one source word.

The conversion of every pair of parallel sentences into an extended symbol string follows this algorithm:

```
for i = 1, j = 1, 2, ... J
      throw s[j]
      while ((i <= I) && (alignments[i] <= j))
          add t[i]
          i++;
while (i <= I)
      add t[i]
      i++;
```

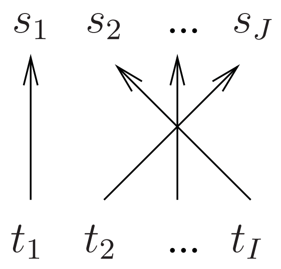

Fig. 2. An alignment situation

which means that compound symbols are left-to-right generated (`throw`), and where a target word t_i is merged (`add`) with its corresponding source word s_{a_i} iff their alignment $t_i \rightarrow s_{a_i}$ does not cross over any other alignment that has not been explored yet. If this is not possible, then the appearance of t_i is delayed until j reaches, then attached to, the last source word that is implied within the group of alignment crossing. Spurious source and target words are placed at their right position, given that a monotonous word order is always demanded. This procedure ensures that every extended symbol is composed of one and only one source symbol, optionally followed by an arbitrary number of target symbols. For example, the alignment in figure 2 would cause the string "$s_1 t_1$, s_2, s_3, ... , $s_J t_2 t_3 \ldots t_I$" to be produced. If a more detailed description about the labelling function is preferred, see [11].

A smoothed n-gram model may be inferred from the string corpus previously generated. Such a model can be expressed in terms of a stochastic finite-state automaton [13]. Figure 3 shows a general scheme for the representation of n-gram models through finite state machines.

No-backoff transitions jump from states in a determined layer to the one immediately above, thus increasing the history levels. Once the top level has been reached, n-gram transitions allow for movements inside this layer, from state to state, updating the history to the last $n-1$ seen events. Backoff transitions to lower history levels are taken if no way is found from a specific state for a given symbol $\mathbf{s}_j$. If the lowest level is reached and no unigram transition is found for $\mathbf{s}_j$, then a transition to the <unk> state is fired, thus considering $\mathbf{s}_j$ as an unknown word. There is only one initial state, which is denoted as <s>, and it is placed at the history level 1.

Since every unigram, bigram, etc., is represented as a transition consuming their last symbol, and given that all these extended symbols are composed of exactly one source word, the inverse labelling function can be straight-forwardly applied. This way, transition labels are turned back into pairs of source/target words to become a transducer.

Again, since every consuming transition implies that only one source symbol needs to be parsed, the beam-search Viterbi algorithm can be appropriately employed for decoding purposes.

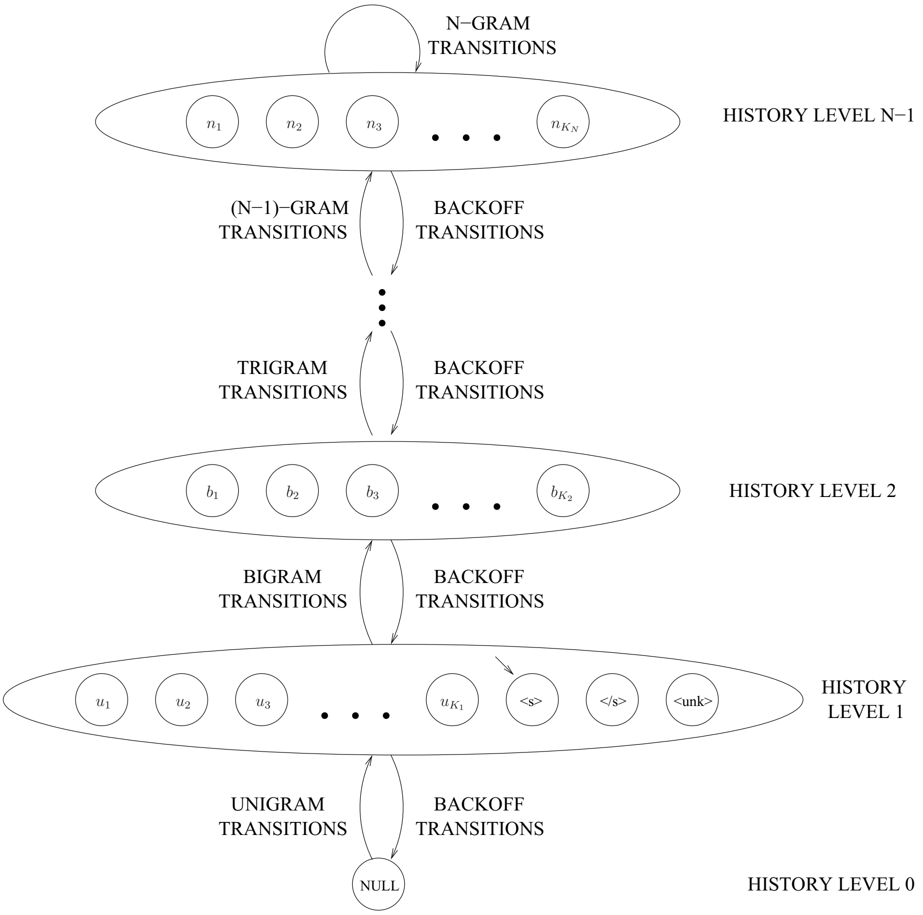

Fig. 3. A finite-state n-gram model

4 Phrase-based finite state models

Some recent researching lines are trying to merge the phrase-based methodology within a finite state framework [14]. There, a generative translation process, which is composed of several transduction models, is applied. Each constituent distribution of the model, including some well-known aspects in SMT, such as phrase reordering or spurious word insertion, is implemented as a weighted finite state transducer. The GIATI paradigm, however, tries to merge all these operations into only one transduction model.

Phrase-based finite state models come from the concept of monotonous bilingual segmentation, where it is assumed that only segments of contiguous words are considered, that every pair of source/target sentences is split up into the very same number of segments, and that they are one-to-one monotonously aligned.

On this occasion, extended symbol strings would be composed of their corresponding sequences of bilingual segments.

A bilingual segmentation of the training corpus can be approximated through a phrase-based statistical machine translation approach. In general terms, a statistical phrase-based model consist of a stochastic phrase translation table. From this table, those phrase pairs that best match a parallel training sample can be selected to approximate a bilingual segmentation. Such a phrase selection can be monotonously generated by translating the source-training sentences with that phrase-based model, since decoding implies looking for the best segmentation.

Again, a smoothed n-gram model can be inferred from the extended symbol corpus. Nevertheless, last step of GIATI cannot be applied as directly as word-based models do. As figure 4 shows, no-backoff transitions are labelled with a many-to-many extended symbol and they are assigned only one probability.

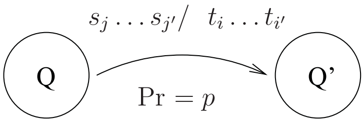

Fig. 4. Phrase-based automata transitions

If a transition label only contains one source symbol, the transformation is the same as for word-based models. However, the inverse labelling algorithm needs to divide all transitions including more than one source symbol.

These transitions are divided by the length of the source segment, putting only one source symbol on every resulting transition. The output segments are delayed to their last transition, which is reaching Q', thus forcing the previous ones to produce the empty string λ. Finally, probabilities are placed at their first transition, leaving 1-probability to the others. Figure 5 shows how this algorithm works.

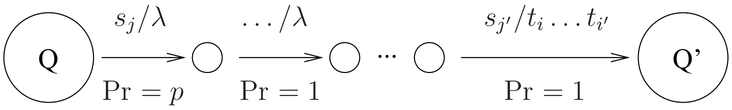

Fig. 5. Phrase-based transducer transitions

Intermediate states are artificially created on-the-fly and do not belong to the original automaton model. They are non-final states that should be parsed through until a real state is being reached, i.e. Q' in figure 5.

Actually, these transition sequences have to be seen as a unique transition: the one corresponding to $\mathrm{Pr}(s_j \dots s_{j'}, t_i \dots t_{i'}|\mathrm{Q})$, that is, the phrase translation probability after a given history Q.

When in decoding time, the search algorithm takes into account such a special situation, thus trying to follow all the paths coming from a determinate state which are compatible with the input string that has not been analysed yet. This parsing behaviour, i.e. the non-stop at intermediate states thing, can be easily implemented adding some extra conditions to the Viterbi algorithm and including more information within the trellis structure that is commonly employed.

Yet another change to the search algorithm is needed because of the phrase-based nature of the proposed approach. Given a starting state Q, a successful path from Q to any Q' would take into account a phrase-based n-gram event, that is, a phrase pair $(s_j \dots s_{j'}, t_i \dots t_{i'})$ that was seen during training after a given history Q. However, if only these paths are explored, the model may not be as effective as it would be able to be.

As a result, backoff transitions must be always allowed in order to cover all compatible phrases in the model, not only the ones which have been seen after a given history, but from lower levels as well. One more constraint has to be included into the parsing algorithm: any directly reaching state Q' is unable to be reached through a path that implies a backoff transition between Q and Q'. Backoff transitions are followed in order to consider all the possible segmentations of the input sentence.

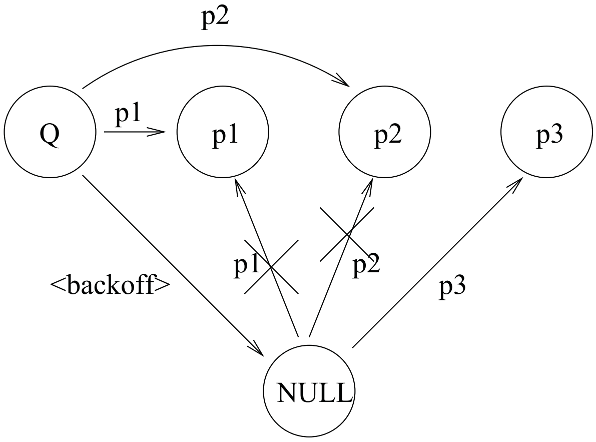

Fig. 6. Compatible transitions for a phrase-based bigram model

Figure 6 shows a parsing example over a finite-state representation of a bigram model. Given a reaching state Q, phrases *p1*, *p2* and *p3* are all compatible with the portion of the input sentence that has not been parsed yet. However, the bigram (Q, *p3*) did not occur throughout the training corpus, therefore there is no a direct transition from Q to *p3*. A backoff transition enables the access to *p3* because the bigram (Q, *p3*) turns into a unigram event that is actually inside the model. Again, unigram transitions to *p1* and *p2* must be ignored because their corresponding bigram events were successfully found one level above.

5 Experiments

This approach has been applied to the EuroParl corpus, that is, the benchmark corpus of the NAACL 2006 and 2007 shared tasks of the Workshops on Machine Translation of the Association for Computational Linguistics.

The EuroParl corpus is built on the proceedings of the European Parliament, which are published on its web and are freely available. Because of its nature, this corpus has a large variability and complexity, since the translations into the different official languages are performed by groups of human translators. The fact that not all translators agree in their translating criteria implies that a given source sentence can be translated in various different ways throughout the corpus.

Since the proceedings are not available in every language as a whole, a different subset of the corpus is extracted for every different language pair, thus evolving into somewhat different corpora for each pair.

5.1 Corpus characteristics

Several shared tasks involving, among others, French, English and Spanish languages, were proposed during the NAACL 2006 and 2007 Workshops on Machine Translation.

French→English and Spanish↔English experiments were carried out over the 2006 EuroParl benchmark corpus, whereas only Spanish↔English translation was tackled from the 2007 data.

The characteristics of these corpora can be seen in Table 1.

Table 1. *Characteristics of the EuroParl corpora*

		2006				2007	
		Fr	**En**	**Sp**	**En**	**Sp**	**En**
Training	Sentences	688031		730740		964791	
	Run. words	15.6 M	13.8 M	15.7 M	15.2 M	20.9 M	20.3 M
	Vocabulary	80348	61626	102216	64070	113026	81754
Dev-Test	Sentences	2000		2000		2000	
	Run. words	66200	57951	60332	57951	60243	58059

5.2 System evaluation

We evaluated the quality of a statistical machine translation system by using the following evaluation measures:

BLEU *(Bilingual Evaluation Understudy) score*: This indicator computes the precision of unigrams, bigrams, trigrams, and tetragrams with respect to a set of reference translations, with a penalty for too short sentences [15]. BLEU measures accuracy, not error rate.

WER *(Word Error Rate)*: The WER criterion calculates the minimum number of editions (substitutions, insertions or deletions) needed to convert the system hypothesis into the sentence considered ground truth. Because of its nature, this measure is considered to be a pessimistic indicator.

5.3 Translation results

On the one hand, word-based finite state models are based on statistical alignments, which were obtained from the application of the public available tool GIZA++ [16] to the corresponding training corpora. On the other hand, phrase-based finite state models are required to operate with a bilingual segmentation of the training corpus. These bilingual segmentations were provided by means of a statistical phrase-based machine translation system such as Pharaoh [17].

Table 2. *Translation results over the EuroParl corpora*

	Word-based		**Phrase-based**	
Corpus	BLEU	WER	BLEU	WER
2006 fr→en	20.0	64.1	28.0	61.9
2006 sp→en	20.6	63.9	27.6	61.6
2006 en→sp	16.8	67.9	26.4	62.3
2007 sp→en	21.9	62.9	28.0	59.6
2007 en→sp	20.1	64.9	25.3	60.8

From the translation results that are presented in Table 2, it can be concluded that phrase-based finite-state models clearly outperform the models that are strictly based on words, within the context of such a EuroParl translation task. Phrase-based finite state models are almost achieving a relative improvement of 35% of BLEU over the language pairs and translation directions that have been tested on.

6 Conclusions and further work

Phrase-based alignment models have become the predominant technology in statistical machine translation. However, finite state models are always an interesting approach to be taken into account in translation matters because they present some advantages with respect to the use of pure source-channel models.

The idea of using phrase-based (rather than word-based) dictionaries can also be brought to a finite state framework. This paper has presented the implementation details that are needed to build a phrase-based finite state model from a bilingual segmentation of the training corpus. Indeed, the algorithmic phenomena that have to be taken into account in order to deal with such phrase-based finite state models when in decoding time have also been in-depth described.

Experiments concerning several language pairs from the EuroParl corpus have been carried out. Translation results from phrase-based finite-state models are clearly outperforming the ones from a word-based finite state framework. An approximate relative improvement of 35% over the BLEU metric is observed for most of the language pairs and translation directions that have been tested on.

Phrase-based finite state models come from the concept of monotonous bilingual segmentation. The experiments reported here are based on a single bilingual segmentation per every pair of training sentences. That is, any other way of splitting a given pair to produce a different monotonous bilingual segmentation is therefore discarded. Learning from all the possible segmentations (rather than from the most likely one) that are compatible with a given alignment of a training pair will probably enrich the models, since the useful information that is extracted from the training data increases. This will be part of our future work.

References

1. Brown, P.F., Cocke, J., Pietra, S.D., Pietra, V.J.D., Jelinek, F., Lafferty, J.D., Mercer, R.L., Roossin, P.S.: A statistical approach to machine translation. Computational Linguistics **16**(2) (1990) 79–85
2. Brown, P.F., Pietra, S.D., Pietra, V.J.D., Mercer, R.L.: The mathematics of statistical machine translation: Parameter estimation. Computational Linguistics **19**(2) (1993) 263–311
3. Ney, H., Nießen, S., Och, F.J., Tillmann, C., Sawaf, H., Vogel, S.: Algorithms for Statistical Translation of Spoken Language. IEEE Trans. on Speech and Audio Processing, Special Issue on Language Modeling and Dialogue Systems **8** (January 2000) 24–36
4. Casacuberta, F., Ney, H., Och, F.J., Vidal, E., Vilar, J.M., Barrachina, S., García-Varea, I., Llorens, D., Martínez, C., Molau, S.: Some approaches to statistical and finite-state speech-to-speech translation. Computer Speech & Language **18**(1) (2004) 25–47
5. Casacuberta, F., Vidal, E.: Machine translation with inferred stochastic finite-state transducers. Computational Linguistics **30**(2) (2004) 205–225
6. Tomás, J., Casacuberta, F.: Monotone statistical translation using word groups. In: Procs. of the Machine Translation Summit VIII. (2001) 357–361
7. Marcu, D., Wong, W.: A phrase-based, joint probability model for statistical machine translation. In: Procs. of the Conference on Empirical Methods in Natural Language Processing and Very Large Corpora EMNLP-02. (2002)
8. Zens, R., Och, F.J., Ney, H.: Phrase-based statistical machine translation. In Jarke, M., Koehler, J., Lakemeyer, G., eds.: KI. Volume 2479 of Lecture Notes in Computer Science., Springer (2002) 18–32
9. Zens, R., Ney, H.: Improvements in phrase-based statistical machine translation. In: HLT-NAACL. (2004) 257–264
10. Mohri, M., Pereira, F., Riley, M.: Weighted finite-state transducers in speech recognition. Computer Speech & Language **16**(1) (2002) 69–88

11. Casacuberta, F., Vidal, E., Picó, D.: Inference of finite-state transducers from regular languages. Pattern Recognition **38**(9) (2005) 1431–1443
12. Jelinek, F.: Statistical Methods for Speech Recognition. The MIT Press (January 1998)
13. Llorens Piñana, D.: Suavizado de autómatas y traductores finitos estocásticos. PhD Thesis, Departamento de Sistemas Informáticos y Computación, Universidad Politécnica de Valencia (2000)
14. Kumar, S., Deng, Y., Byrne, W.: A weighted finite state transducer translation template model for statistical machine translation. Nat. Lang. Eng. **12**(1) (2006) 35–75
15. Papineni, K., Roukos, S., Ward, T., Zhu, W.: Bleu: a method for automatic evaluation of machine translation (2001)
16. Och, F.J., Ney, H.: A systematic comparison of various statistical alignment models. Computational Linguistics **29**(1) (2003) 19–51
17. Koehn, P.: Pharaoh: A beam search decoder for phrase-based statistical machine translation models. In Frederking, R.E., Taylor, K., eds.: AMTA. Volume 3265 of Lecture Notes in Computer Science., Springer (2004) 115–124