Using Word Posterior Probabilities in Lattice Translation

Vicente Alabau, Alberto Sanchis, Francisco Casacuberta

Departament de Sistemes Informàtics i Computació
Institut Tecnològic d'Informàtica
Universitat Politècnica de València, Spain
{valabau,asanchis,fcn}@iti.upv.es

Abstract

In this paper we describe the statistical machine translation system developed at ITI/UPV, which aims especially at speech recognition and statistical machine translation integration, for the evaluation campaign of the International Workshop on Spoken Language Translation (2007).

The system we have developed takes advantage of an improved word lattice representation that uses word posterior probabilities. These word posterior probabilities are then added as a feature to a log-linear model. This model includes a stochastic finite-state transducer which allows an easy lattice integration. Furthermore, it provides a statistical phrase-based reordering model that is able to perform local reorderings of the output.

We have tested this model on the Italian-English corpus, for clean text, 1-best ASR and lattice ASR inputs. The results and conclusions of such experiments are reported at the end of this paper.

1. Introduction

This paper describes the statistical machine translation (SMT) system developed at ITI/UPV for the evaluation campaign of the International Workshop on Spoken Language Translation (2007). The system proposed aims especially at speech recognition and statistical machine translation integration.

Although different approaches to speech input translation have been investigated [1, 2, 3], there is still a big gap between the translation performance from the correct text and from an automatic speech recognizer (ASR). The most simple approach performs the two processes in a serial manner: first, an input utterance is decoded into a sentence using a conventional ASR, and afterwards, this sentence is translated using a *text-to-text* translator. The main drawback of this approach is that the output of an ASR can contain misrecognized words and, consequently, the quality of the translated sentences decreases. In order to circumvent this problem, different solutions have been proposed [4, 5, 6]. In [4] N-best lists have been used for improving the quality of the translated sentences. In [5] the translation process is performed using as input a word lattice and acoustic recognition scores. In [6] the translation process is performed using confusion networks and posterior probabilities. All these approaches try to exploit a set of the most probable hypotheses instead of only the best one.

The use of stochastic finite-state transducers provides a fully integrated recognition-translation architecture in which the source and target sentences are obtained simultaneously [1, 3]. However, the experimental results are not consistently better than serial approach [3].

In our system, we propose the use of an improved word lattice representation for speech-to-speech translation following a semi-coupled architecture integrated in a log-linear model. Instead of using acoustic recognition scores [5] we use the word posterior probabilities computed over the word lattice, which take into account different word cooccurrences in the same interval of time while preserving the lattice structure.

2. A review to speech translation

In this section, a review of the formulation defined in [3] is presented. The problem of speech-input statistical translation can be formulated as:

$$\hat{e}_1^{\hat{I}} = \arg\max_{I,e_1^I} Pr(e_1^I|\vec{x}_1^T) \quad (1)$$

where $\vec{x}_1^T$ are the acoustic vectors and $\hat{e}_1^{\hat{I}}$ is the most probable translation of the speech utterance. The maximization is performed over all possible target sentences e_1^I and all possible lengths I.

The process can be stated as: $\vec{x}_1^T \rightarrow f_1^J \rightarrow e_1^I$ where f_1^J is the input decoding of $\vec{x}_1^T$, and e_1^I is the corresponding translation of f_1^J. Consequently, Eq. 1 can be decomposed by:

$$\arg\max_{I,e_1^I} Pr(e_1^I|\vec{x}_1^T) = \arg\max_{I,e_1^I} \sum_{f_1^J} Pr(e_1^I, f_1^J|\vec{x}_1^T) \quad (2)$$

with the practical assumption that $Pr(x_1^T|e_1^I, f_1^J)$ does not depend on the target sentence e_1^I, Eq. 2 can be decomposed by:

$$\arg\max_{I,e_1^I} Pr(e_1^I|\vec{x}_1^T) = \arg\max_{I,e_1^I} \sum_{f_1^J} Pr(e_1^I, f_1^J) Pr(\vec{x}_1^T|f_1^J) \quad (3)$$

We approximate the sum over all possible source language sentences by the maximum. The purpose is to associate a source sentence to the input utterance whose translation is the target sentence searched for. From Eq. 3,

$$\arg\max_{I,e_1^I} Pr(e_1^I|\vec{x}_1^T) \approx \arg\max_{I,e_1^I} \max_{f_1^J} Pr(e_1^I, f_1^J) Pr(\vec{x}_1^T|f_1^J) \quad (4)$$

$Pr(e_1^I, f_1^J)$ refers to the translation model and $Pr(\vec{x}_1^T|f_1^J)$ is modeled by acoustic models (typically *Hidden Markov Models* (HMM)).

3. System description

3.1. Log-linear model

Recently, log-linear models [7] have become very popular in SMT. Their success relies on the fact that different sources of knowledge can be easily integrated into the model. Following the well-founded maximum entropy framework, the posterior probability $Pr(e_1^I|f_1^J, \vec{x}_1^T)$ is given by:

$$Pr(e_1^I|f_1^J, \vec{x}_1^T) = \frac{\exp\left(\sum_{m=1}^M \lambda_m h_m\left(e_1^I, f_1^J, \vec{x}_1^T\right)\right)}{\sum_{e'^{I'}_1} \exp\left(\sum_{m=1}^M \lambda_m h_m\left(e'^{I'}_1, f_1^J, \vec{x}_1^T\right)\right)} \quad (5)$$

As the denominator, which represents a normalization factor, only depends on the source sentence f_1^J and the acoustic vectors $\vec{x}_1^T$, they can be dropped from Eq. 5 in the search algorithm. The resulting decision rule remains as follows:

$$\hat{e}_1^{\hat{I}} = \arg\max_{I,e_1^I} \left\{ \sum_{m=1}^M \lambda_m h_m\left(e_1^I, f_1^J, \vec{x}_1^T\right) \right\} \quad (6)$$

It must be noted that Eq. 4 can be expressed in terms of Eq. 6 for $M = 2, \lambda_1 = \lambda_2 = 1$, and

$$\begin{aligned} h_1(e_1^I, f_1^J, \vec{x}_1^T) &= \log Pr(e_1^I, f_1^J) \\ h_2(e_1^I, f_1^J, \vec{x}_1^T) &= \log Pr(\vec{x}_1^T|f_1^J) \end{aligned}$$

The next subsection is devoted to explain the set of different features that have been included in the log-linear model.

3.2. Features

3.2.1. Word posterior probabilities

A word lattice G is a directed, acyclic, weighted graph. The nodes correspond to discrete points in time. The edges are triplets $[w,\ s,\ e]$, where w is the hypothesized word from node s to node e. The weights are the acoustic recognition scores associated to the word lattice edges. Any path from the initial to the final node forms a hypothesis f_1^J.

Given the acoustic observations $\vec{x}_1^T$, the posterior probability for a specific word (edge) $[w,\ s,\ e]$ can be computed by summing up the posterior probabilities of all hypotheses of the word lattice containing the edge $[w,\ s,\ e]$:

$$P([w,\ s,\ e] \mid \vec{x}_1^T) = \frac{1}{P(\vec{x}_1^T)} \sum_{\substack{f_1^J \in G: \\ \exists [w', s', e']: \\ w'=w, s'=s, e'=e}} P(f_1^J,\ \vec{x}_1^T) \quad (7)$$

The probability of the sequence of acoustic observations $P(\vec{x}_1^T)$ can be computed by summing up the posterior probabilities of all word lattice hypotheses:

$$P(\vec{x}_1^T) = \sum_{f_1^J \in G} P(f_1^J, \vec{x}_1^T) \quad (8)$$

These posterior probabilities can be efficiently computed based on the well-known *forward-backward* algorithm [8].

The posterior probability defined in Eq. 7 does not perform well because of a word w can occur with slightly different starting and ending times. This effect is represented in the word lattice by different word lattice edges and the posterior probability mass of the word is scattered among the different word segmentations (see Figure 1).

To deal with this problem, we have considered a method proposed in [8]. Given a specific word (edge) $[w,\ s,\ e]$ and a specific point in time $t \in [s, e]$, we compute the posterior probability of the word w at time t by summing up the posterior probabilities of the word lattice edges $[w, s', e']$ with identical word w and for which t is within the interval time $[s', e']$:

$$P_t(w \mid \vec{x}_1^T) = \sum_{[w,\ s',\ e']:\ t \in [s', e']} P([w,\ s',\ e'] \mid \vec{x}_1^T) \quad (9)$$

Based on Eq. 9, the posterior probability for a specific word $[w,\ s,\ e]$ is computed as the maximum of the frame time posterior probabilities:

$$P([w,\ s,\ e] \mid \vec{x}_1^T) = \max_{s \le t \le e} P_t(w \mid \vec{x}_1^T) \quad (10)$$

The probability computed by Eq. 10 is in the interval $[0, 1]$ since, by definition, the sum of the word posterior probabilities for a specific point in time must sum to one (this property can be appreciated in the Figure 1). However, it must be noticed that the resulting lattice does not represent a real probability distribution. Figure 1 shows an example of the word lattice with the word posterior probabilities computed following the Eq. 7. Figure 2 shows the same word lattice after Eq. 10 is computed.

We will use these posterior probabilities to compute the conditional probability of the acoustic signal given a source hypothesis:

$$Pr(f_1^J|\vec{x}_1^T) = \prod_{j=1}^J P([w_j, s_j, e_j]|\vec{x}_1^T) \quad (11)$$

where s_j and e_j are the starting and the ending time, respectively, of the source word w_j.

Hence, the word posterior feature is given by:

$$h_{\mathrm{G}}(e_1^I, f_1^J, \vec{x}_1^T) = \log Pr(f_1^J|\vec{x}_1^T)$$

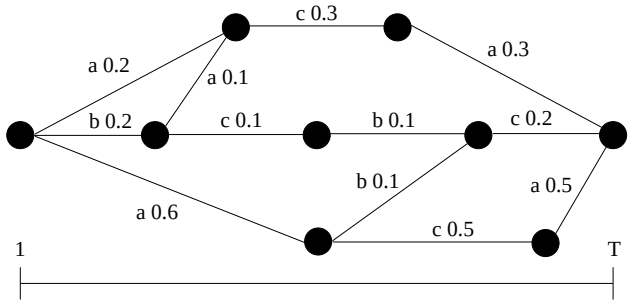

Figure 1: Word lattice with the word posterior probabilities.

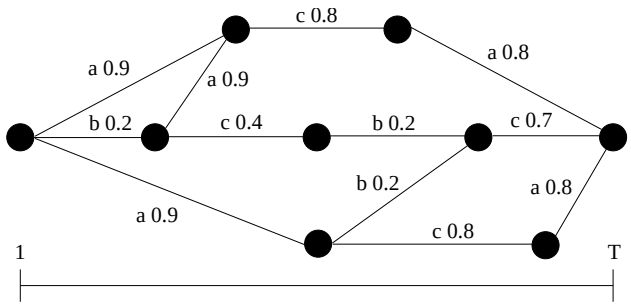

Figure 2: Word lattice after the frame posterior probabilities are computed.

3.2.2. Statistical finite-state transducers

The joint probability distribution $Pr(e_1^I, f_1^J)$ in Eq. 4 can be adequately modelled by means of a statistical finite-state transducer (SFST). SFSTs have been thoroughly studied [9, 10] and several approaches to infer SFSTs from corpora have been proposed in recent years [11, 12, 13].

We have used the Grammatical Inference and Alignments for Transducer Inference (GIATI) technique for inferring the SFST [14]. This technique uses a finite sample of bilingual pairs (parallel corpus) for inferring the SFST in three steps:

1. Building training strings. Each training pair is transformed into a single string from an extended alphabet to obtain a new sample of strings. The transformation of a parallel corpus into a corpus of single sentences is performed with the help of statistical alignments: each word (or substring) is joined with its translation in the output sentence, creating an *extended* symbol.

2. Inferring a (stochastic) regular grammar. Typically, a smoothed n-gram is inferred from the sample of strings obtained in the previous step.

3. Transforming the inferred regular grammar into a transducer. The symbols associated to the grammar rules are transformed into source/target symbols by applying an adequate transformation.

An interesting feature of SFSTs is that the maximization of Eq. 4 can be performed in a fully integrated recognition-translation manner. This is possible since each transition of the SFST is labelled with a source word and its corresponding target translation. Thus, each transition is expanded by the acoustical representation of the source words. Following the standard speech recognition searching algorithm over the SFST, the optimal source and target sentences are obtained simultaneously. In the current implementation, we have performed an integrated search with the input lattice. The translation feature function is given by:

$$h_{\mathrm{T}}(e_1^I, f_1^J, \vec{x}_1^T) = \log Pr(e_1^I, f_1^J)$$

3.2.3. Output penalty

Usually, SMT models produce sentences which differ in length from the reference sentence. This provokes an important loss of performance in the BLEU measure since it heavily penalizes short sentences. To adjust the length of these sentences, an output word penalty has been added to the model:

$$h_{\mathrm{WP}}(e_1^I, f_1^J, \vec{x}_1^T) = I$$

3.2.4. Output language model

In order to help the fluency an output language model has been added:

$$h_{\mathrm{OL}}(e_1^I, f_1^J, \vec{x}_1^T) = \log Pr(e_1^I)$$

3.3. Phrase-based local reordering model

The GIATI technique described above is known to have limited capability to model non-monotonous translations. Hopefully, the languages for the task we approached are very monotonous for non-local relationships. However, they may have many local inversions. We have addressed this problem by reordering the target sentence in a similar fashion to the work by [15].

From the alignments we can define a new target language e^r which is obtained by monotonizing e respect to f. Both e and e^r are closely related so that they have the same vocabulary, but differ in the word order. Two new parallel corpus can be constructed: one from f and e^r, the other from e^r and e.

It is important to note that in Eq. 6 the target language is now monotonized, so the decision rule remains as follows:

$$\hat{e^r}_1^{\hat{I}} = \arg\max_{I, e^r{}_1^I} \left\{ \sum_{m=1}^{M} \lambda_m h_m \left(e^r{}_1^I, f_1^J, \vec{x}_1^T \right) \right\} \qquad (12)$$

The output sentence $\hat{e^r}_1^{\hat{I}}$ resulting from the maximization is not in the appropriate order. Hence, the reordering problem can be defined as follows:

$$\hat{e}_1^{\hat{I}} = \arg\max_{e_1^{\hat{I}}} Pr(e_1^{\hat{I}}) Pr(\hat{e^r}_1^{\hat{I}} | e_1^{\hat{I}}) \qquad (13)$$

where $Pr(e_1^{\hat{I}})$ is a target language model, and $Pr(\hat{e^r}_1^{\hat{I}} | e_1^{\hat{I}})$ is the reordering model. Therefore, the reordering model is applied to the best translation from the log-linear model in a serial manner.

4. IWSLT07 evaluation campaign

The experiments were carried out on the Basic Travel Expression Corpus (BTEC) task [16] for the Italian-English language pair. BTEC is a multilingual corpus which contains sentences related to travel expressions similar to those that are found in traveler's phrase books. The statistics for this task are shown in Table 1. The corpus consists of six development sets. However, the experiments were conducted with three of them: *dev4*, *dev5b*. and *dev5b*. The development sets were not added to the training data as they did not show to improve the results. No additional data was exploited during the experiments.

Table 1: *Corpus statistics for training, development, and test. OOV stands for Out Of Vocabulary words.*

		Italian	English
train	Sentences	19971	
	Running words	$172k$	$189k$
	Vocabulary	$10,152$	$7,165$
dev4	Sentences	489	
	Running words	$4,831$	$6,848$
	OOV words	224	208
dev5a	Sentences	500	
	Running words	$5,607$	$7,491$
	OOV words	296	264
dev5b	Sentences	996	
	Running words	$8,487$	$11,968$
	OOV words	591	611
test	Sentences	724	
	Running words	$6,420$	$9,054$
	OOV words	542	439

4.1. Practical aspects

4.1.1. Punctuation and case restoration

The evaluation is case sensitive and with punctuation marks. We followed the approach[1] given by the IWSLT06 organizers instead. First, punctuation and case were removed from the training. Next, in the postprocess module punctuation and case were restored, in that particular order, using the *hidden-ngram* and *disambig* tools [17]. Finally, words after sentence boundaries[2] were capitalized. This method provided the best results in our experiments.

4.1.2. Sentence splitting in training

Several samples in the training set consist of two or more sentences. Occasionally, after the alignment estimation, some alignments have crossed sentence boundaries. In principle, this is not desirable since usually each sentence is independent from the others. Furthermore, the heuristics used in GIATI to extract bilingual pairs are very sensitive to long alignments. Therefore, in order to improve the alignments, the training sentences were splitted at sentence boundaries when applicable.

4.1.3. Recognition score scaling

When computing word posterior probabilities, it is necessary to scale properly the acoustic and language model scores [18]. Otherwise, the resulting probability distribution would be likely to have a peak on the best recognition sentence, affecting negatively the additive combination of posteriors. In our experiments, this parameter was adjusted to minimize the word error rate of the source sentence on the development set.

4.1.4. Lattice pruning

One important aspect to keep in mind when translating word lattices is that the computational cost increases enormously. However, most of the hypotheses have very low probability and are not worth being explored. In fact, it has been observed that an appropriate pruning can even benefit translation results since very unlikely hypothesis will never succeed.

As it has be shown in [19], lattice pruning by confidence measures achieve good results. It is for that reason, that we pruned the word lattices to reduce lattice density by removing paths which do not reach a threshold. We selected the threshold that, while keeping the translation performance, had a reasonable lattice density.

4.2. Experimental results

As mentioned at the beginning of this section, we participated in the Italian-English track. For all experiments, two accuracy measures are reported: BLEU [20] and NIST [21]. The parameters for the development sets were optimized for BLEU. For the test set, the parameters of the *dev5b* were used owing to the similar conditions of both data sets. Furthermore, we found out later that parameters from *dev5b* performed better. The results presented in this section were obtained after the official submission period was over.

Table 2 shows the results for the development and test sets depending on which features were added to the model for the clean input. The baseline model is a GIATI transducer. The feature codes are the following: *WP*, output word insertion penalty; *OL*, output language model; *RM*, reordering model. The experiments were run on the clean Italian-English data.

The *WP* feature consistently improves the BLEU for all development datasets although the NIST scores are worse. On the contrary, for the test set *WP* makes the system perform worse in terms of BLEU while improving the NIST score. This fact suggests that the parameters are overfitted.

[1] http://www.slc.atr.jp/IWSLT2006/downloads/case+punc_tool_using_SRILM.instructions.txt

[2] We define a sentence boundary as the beginning of the sentence or any of these punctuation marks: .?!

Table 2: *Effect of adding features to the baseline model for the clean input. The features are:* WP*, output word insertion penalty;* OL*, output language model;* RM*, reordering model.*

	dev4		dev5a		dev5b		test	
	BLEU	NIST	BLEU	NIST	BLEU	NIST	BLEU	NIST
baseline	36.29	7.59	31.96	7.06	12.53	4.02	22.80	5.49
+WP	37.45	7.35	32.55	6.82	14.07	3.77	22.09	5.56
+OL	37.06	7.42	32.55	6.91	12.63	4.06	22.79	5.52
+WP+OL	38.19	7.20	32.67	6.66	13.44	4.20	21.79	5.56
+RM	37.53	7.95	32.74	7.41	13.94	4.30	23.46	5.74
+WP+OL+RM	38.98	7.81	32.86	7.18	14.34	4.37	23.22	5.86

Table 3: *Results for different input conditions.* GER *represents the best result achievable by using lattices.* LAT *corresponds to our integrated approach using word posteriors.*

	dev4		dev5a		dev5b		test	
	BLEU	NIST	BLEU	NIST	BLEU	NIST	BLEU	NIST
1BEST	33.53	6.92	26.97	6.12	13.21	4.19	21.50	5.56
LAT	33.69	6.95	27.24	6.14	13.35	4.16	20.57	5.43
GER	34.11	7.02	27.49	6.18	13.90	4.29	22.64	5.77
CLEAN	38.98	7.81	32.86	7.18	14.34	4.37	23.22	5.86

Regarding the reordering model, it may be seen that the improvements are consistent in all datasets. Taking into account that only the single best translation was reordered, we may expect further improvements from an integrated reordering model.

Table 3 shows the results for the different input conditions. The *CLEAN* condition refers to the correct transcription, while the *1BEST* condition refers to the single best output from the ASR module. In between, it is the *LAT* condition, where we applied the method described in Subsection 3.2.1 to compute word posterior probabilities. We also have added a *GER* condition. For this condition, we extracted from the lattices the sentences which provided the lower word error rate. This condition represents the upper limit of using lattices for translation since no better sentence can be found in the lattice. The different input conditions are sorted Table 3 in a way that it is easy to notice the improvement in performance as the input quality increases.

In all development sets, *LAT* outperforms *1BEST* for both measures. Although these improvements are not very impressive, it must be observed that the room for improvement is very small. Although Table 4 shows that recognition performance of *GER* can be much better, improvements in BLEU (Table 3) are not that good. Furthermore, it may be observed that large lattices are more likely to encode better sentences. Unfortunately, processing large lattices is much more memory and time consuming. Although our system implements beam search, the heuristics did not show useful for this particular problem, so we had to apply posterior lattice pruning as described in Subsection 4.1.4. This issue would be probably solved with the use of a more elaborated pruning. As a consequence of pruning and parameter overfitting, lattice translation did not performed as expected, leading to scores worse than *1BEST*.

Table 4: *Lattice statistics for the different datasets.* GER *is the graph error rate, i.e. the word error rate for the most accurate sentence in the graph. The last column represents the number of words per lattice in average.*

	WER	GER	N. words (avg)
dev4	22.31	20.38	255.1
dev5a	24.33	22.29	264.8
dev5b	10.29	3.73	2994.1
test	10.70	4.39	2996.6

5. Conclusions

In this work, we have described the ITI/UPV system for speech recognition and machine translation integration. We have presented a novel technique to perform this integration by using word posterior probabilities as a feature to a log-linear model. Although the system is in an early stage of development we have shown that it is able to perform improvements in translation performance.

There are still a lot of issues to cover. However, we are encouraged to solve this problems for the next evaluation campaign.

6. Acknowledgements

Work supported by the "Agència Valenciana de Ciència i Tecnologia" under grant GRUPOS03/031, the EC (FEDER),

the Spanish MEC under grant TIN2006-15694-CO2-01, "Programas de Apoyo a la Investigacion y Desarrollo 2007 UPV", and the "Programa d'Incentiu a la Investigació 2004 UPV".

7. References

[1] E. Vidal, "Finite-State Speech-to-Speech Translation," in *Int. Conf. on Acoustics Speech and Signal Processing, Vol.1*, Munich, 1997, pp. 111–114.

[2] H. Ney, "Speech Translation: Coupling of Recognition and Translation," in *Int. Conf. on Acoustics Speech and Signal Processing*, Phoenix, 1999, pp. 1149–1152.

[3] F. Casacuberta, H. Ney, F. Och, E. Vidal, J. Vilar, S. Barrachina, I. Garcia-Varea, D. Llorens, C. Martinez, S. Molau, F. Nevado, M. Pastor, D. Pico, and A. Sanchis., "Some approaches to statistical and finite-state speech-to-speech translation," *Computer Speech and Language*, vol. 18, pp. 25–47, 2004.

[4] V. Quan, "Integrated n-best re-ranking for spoken language translation," in *9th European Conference on Speech Communication and Technology, Interspeech*, Lisbon, Portugal, 2005.

[5] E. Matusov, S. Kanthak, and H. Ney, "On the integration of speech recognition and statistical machine translation," in *9th European Conference on Speech Communication and Technology, Interspeech*, Lisbon, Portugal, 2005, pp. 3177–3180.

[6] N. Bertoldi and M. Federico, "A new decoder for spoken language translation based on confusion networks," in *IEEE Automatic Speech Recognition and Understanding Workshop*, 2005.

[7] F. J. Och and H. Ney, "Discriminative training and maximum entropy models for statistical machine translation," in *ACL '02: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*. Morristown, NJ, USA: Association for Computational Linguistics, 2001, pp. 295–302.

[8] F. Wessel, R. Schluter, K. Macherey, and H. Ney, "Confidence measures for large vocabulary continuous speech recognition," *IEEE Trans. Speech and Audio Processing*, vol. 9, no. 3, Mar 2001.

[9] E. Vidal, F. Thollard, C. de la Higuera, F. Casacuberta, and R. Carrasco, "Probabilistic finite-state machines - part I," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 7, pp. 1013–1025, 2005.

[10] ——, "Probabilistic finite-state machines - part II," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 7, pp. 1025–1039, 2005.

[11] S. Kumar, Y. Deng, and W. Byrne, "A weighted finite state transducer translation template model for statistical machine translation," *Journal of Natural Language Engineering*, vol. 12(1), pp. 35–75, dec 2005.

[12] F. Casacuberta and E. Vidal, "Machine translation with inferred stochastic finite-state transducers," *Computational Linguistics*, vol. 30, no. 2, pp. 205–225, 2004.

[13] C. Allauzen, M. Mohri, M. Riley, and B. Roark, "A generalized construction of integrated speech recognition transducers," in *IEEE International Conference on Acoustic, Speech and Signal Processing*, vol. 1. IEEE Press, May 2004, pp. 761–764.

[14] D. Picó, "Combining statistical and finite-state methods for machine translation," Tesis Doctoral en Informtica, Departamento de Sistemas Informticos y Computacin, Universidad Politcnica de Valencia, 2005.

[15] G. Sanchis and F. Casacuberta, "N-Best reordering in Statistical Machine Translation." in *Proc. of IV Jornadas en Tecnología del Habla*, Zaragoza, Spain, 2006.

[16] T. Takezawa, E. Sumita, F. Sugaya, H. Yamamoto, and S. Yamamoto, "Toward a broad-coverage bilingual corpus for speech translation of travel conversations in the real world," in *LREC-2002: Third International Conference on Language Resources and Evaluation*, Las Palmas de Gran Canaria, Spain, May 2002, pp. 147–152.

[17] A. Stolcke, "Srilm - an extensible language modeling toolkit," in *Proc. Intl. Conf. Spoken Language Processing*, Denver, Colorado, September 2002.

[18] L. Mangu, E. Brill, and A. Stolcke, "Finding consensus among words: Lattice-based word error minimization," in *In Proc. Eurospeech'99*, Budapest, pp. 495–498.

[19] R. Zhang, G. Kikui, H. Yamamoto, and W. Lo, "A decoding algorithm for word lattice translation in speech translation," in *Proc. of IWSLT2005*, October 2005, pp. 33–39.

[20] K. Papineni, S. Roukos, T. Ward, and W. Zhu, "Bleu: a method for automatic evaluation of machine translation," Thomas J. Watson Research Center, Tech. Rep. RC22176, September 2001.

[21] G. Doddington, "Automatic evaluation of machine translation quality using n-gram co-occurrence statistics," in *Proceedings of the the second international conference on human language technology research (HLT 2002)*, San Diego, California, March 24-27 2002, pp. 138–145.