From Machine Translation to Computer Assisted Translation using Finite-State Models

Jorge Civera, Elsa Cubel, Antonio L. Lagarda, David Picó,
Jorge González, Enrique Vidal, Francisco Casacuberta
Instituto Tecnológico de Informática
Dpto. de Sistemas Informáticos y Computación, Universidad Politécnica de Valencia
E-46071 Valencia, Spain
jorcisai@iti.upv.es
Juan M. Vilar, Sergio Barrachina
Dpto. de Lenguajes y Sistemas Informáticos, Universidad Jaime I,
E-12071 Castellón de la Plana, Spain
jvilar@lsi.uji.es

Abstract

State-of-the-art machine translation techniques are still far from producing high quality translations. This drawback leads us to introduce an alternative approach to the translation problem that brings human expertise into the machine translation scenario. In this framework, namely Computer Assisted Translation (CAT), human translators interact with a translation system, as an assistance tool, that dinamically offers, a list of translations that best completes the part of the sentence already translated. In this paper, finite state transducers are presented as a candidate technology in the CAT paradigm. The appropriateness of this technique is evaluated on a printer manual corpus and results from preliminary experiments confirm that human translators would reduce to less than 25% the amount of work to be done for the same task.

1 Introduction

State-of-the-art machine translation techniques are still far from producing high quality translations. This drawback leads us to introduce an alternative approach to the translation problem that brings human expertise into the machine translation scenario. (Langlais et al., 2000) proposed this idea that can be illustrated as follows. Initially, the human translator is provided with a possible translation for the sentence to be translated. Unfortunately in most of the cases, this translation is not perfect, so the translator amends it and asks for a translation of the part of the sentence still to be translated (completion). This latter interaction is repeated as many times as needed until the final translation is achieved.

The scenario described in the previous paragraph, can be seen as an iterative refinement of the translations offered by the translation system, that without possessing the desired quality, help the translator to increase his/her productivity. Nowadays, this lack of translation excellence is a common characteristic in all machine translation systems. Therefore, the human-machine synergy represented by the CAT paradigm seems to be more promising than fully-automatic translation in the near future.

The CAT paradigm has two important aspects: the models need to provide adequate completions and they have to do so efficiently to perform under usability constrains. To fulfill these two requirements, Stochastic Finite State Transducers (SFST) have been selected since they have proved in the past to be able to provide adequate translations (Vidal, 1997; Knight and Al-Onaizan, 1998; Amengual et al., 2000; Casacuberta et al., 2001; Bangalore and Ricardi, 2001). In addition, efficient parsing algorithms can be easily adapted in order to provide completions.

The rest of the paper is structured as follows. The following section introduces the general setting for machine translation and finite state models. In section 3, the search procedure for an interactive translation is presented. Experimental results are presented in section 4. Finally, some conclusions and future work are explained in section 5.

2 Machine translation with finite-state transducers

Given a source sentence s, the goal of MT is to find a target sentence $\hat{\mathbf{t}}$ that maximizes:

$$\hat{\mathbf{t}} = \underset{\mathbf{t}}{\operatorname{argmax}} \Pr(\mathbf{t} \mid \mathbf{s}) = \underset{\mathbf{t}}{\operatorname{argmax}} \Pr(\mathbf{t}, \mathbf{s}) \quad (1)$$

The joint distribution $\Pr(\mathbf{t}, \mathbf{s})$ can be modeled by a Stochastic Finite State Transducer $\mathcal{T}$ (Picó and Casacuberta, 2001):

$$\hat{\mathbf{t}} = \underset{\mathbf{t}}{\operatorname{argmax}} \Pr(\mathbf{t}, \mathbf{s}) \approx \underset{\mathbf{t}}{\operatorname{argmax}} \Pr_{\mathcal{T}}(\mathbf{t}, \mathbf{s}) \quad (2)$$

A *Stochastic Finite-State Transducer* (SFST) is a finite-state network whose transitions are labeled by three items:

1. a source symbol (a word from the source language vocabulary);
2. a target string (a sequence of words from the target language vocabulary) and
3. a transition probability.

They have been successfully applied into many translation tasks (Vidal, 1997; Amengual et al., 2000; Casacuberta et al., 2001). Furthermore, there exist efficient search algorithms like Viterbi (Viterbi, 1967) for the best path and the Recursive Enumeration Algorithm (REA) (Jiménez and Marzal, 1999) for the n-best paths.

One possible way of inferring SFSTs is the Grammatical Inference and Alignments for Transducer Inference (GIATI) technique (the previous name of this technique was MGTI - Morphic-Generator Transducer Inference) (Casacuberta et al., 2004). Given a finite sample of string pairs, it works 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 ”extended alphabet” contains words or substrings from source and target sentences coming from training pairs.
2. Inferring a (stochastic) regular grammar. Typically, 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, thereby transforming the grammar inferred in the previous step into a transducer.

The transformation of a parallel corpus into a corpus of single sentences is performed with the help of statistical alignments: each word is joined with its translation in the output sentence, creating an “extended word”. This joining is done taking care not to invert the order of the output words. The third step is trivial with this arrangement. In our experiments, the alignments are obtained using the GIZA software (Och and Ney, 2000; Al-Onaizan et al., 1999), which implements IBM statistical models (Brown et al., 1993).

3 Interactive search

The concept of interactive search is closely related to the CAT paradigm. This paradigm introduces the new factor $\mathbf{t}_p$ into the general machine translation equation (Equation 1). $\mathbf{t}_p$ represents a prefix in the target language obtained as a result of the interaction between the human translator and the machine translation system.

As a side effect of this reformulation, the optimization defined in Equation 3 is performed over the set of target suffixes rather than the set of complete target sentences. Thence, the goal of CAT in the finite-state transducer framework is to find a prediction of the best suffix $\hat{\mathbf{t}}_s$, given a source sentence $\mathbf{s}$, a prefix of the target sentence $\mathbf{t}_p$ and a SFST $\mathcal{T}$:

$$\begin{aligned} \hat{\mathbf{t}}_s &= \underset{\mathbf{t}_s}{\operatorname{argmax}} \Pr(\mathbf{t}_s \mid \mathbf{s}, \mathbf{t}_p) = \underset{\mathbf{t}_s}{\operatorname{argmax}} \Pr(\mathbf{t}_p \mathbf{t}_s, \mathbf{s}) \approx \\ &\approx \underset{\mathbf{t}_s}{\operatorname{argmax}} \Pr_{\mathcal{T}}(\mathbf{t}_p \mathbf{t}_s, \mathbf{s}) \end{aligned} \quad (3)$$

A transducer can be understood as a weighted graph in which every path is a possible source-target sentence pair represented in a compact manner. Given a source sentence $\mathbf{s}$ to be translated, this sentence is initially employed to define a set of paths in the transducer, whose sequence of source symbols is compatible with the source sentence. Equation 3 is just defining the most probable path (target suffix $\hat{\mathbf{t}}_s$) among those that are compatible, having $\mathbf{t}_p$ as a target prefix.

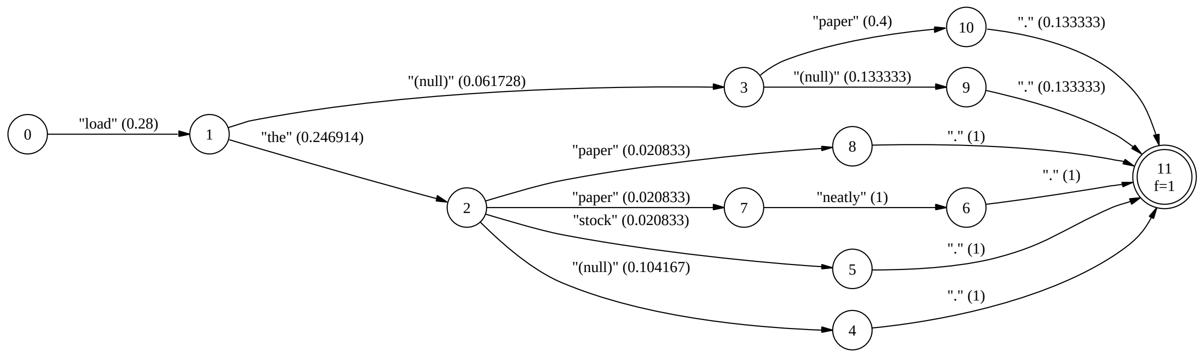

Figure 1: Resultant word graph given the source sentence ”cargue el papel”

The search for this path (the product of the probabilities associated with its edges is maximum) is performed according to the Viterbi decoding over the set of paths that were compatible with the source sentence. The concatenation of the target symbols of this best path will give place to the target sentence (translation).

The solution to the search problem has been devised in two phases. The first one copes with the extraction of a word graph $\mathcal{W}$ from a SFST $\mathcal{T}$ given a source sentence $\mathbf{s}$. A word graph represents the set of paths whose sequence of source symbols is compatible with the source sentence $\mathbf{s}$.

The second phase involves the search for the best translation over the word graph $\mathcal{W}$. To be more precise, in the present work the concept of best translation has been extended to a set of best translations (n-best translations). This search can be carried out efficiently taking into account not only the *a posteriori* probability of a given translation $\hat{\mathbf{t}}$, but also the minimum edit cost with respect to the target prefix. The way in which this latter criterium is integrated in the search process will be explain in section 3.2.

3.1 Word-graph derivation

A word graph represents the set of all possible translations for a given source sentence $\mathbf{s}$ that were embeded in the SFST $\mathcal{T}$. The derivation of the word graph is performed by intersecting the SFST $\mathcal{T}$ with the source sentence $\mathbf{s}$ defining a subgraph in $\mathcal{T}$ whose paths are compatible with the source sentence.

Interactive search can be simplified significantly by using this representation of the set of target sentences, since the inclusion of edit cost operations along with the search procedure introduces some peculiarities that can be solved efficiently in the word graph. An example of word graph is shown in Figure 1.

3.2 Search for n-best translations given a prefix of the target sentence

The application of this type of search is aimed at the core of CAT. In this paradigm, given a source sentence $\mathbf{s}$, the human translator is provided with a list of n translations, also called n-best translations. Then, the human translator will proceed to accept a prefix of one of these n-best translations as correct, appending some rectifications to the selected prefix. This new prefix of the target sentence $\mathbf{t}_p$ together with the source sentence $\mathbf{s}$ will generate a new set of best translations that will be again modified by the human translator. This process is repeated as many times as neccessary to achieve the desired final translation.

Ideally, the task would be to find the target suffix $\mathbf{t}_s$ that maximizes the probability *a posteriori* given a prefix $\mathbf{t}_p$ of the target sentence and the input sentence. In practice, however, it may happen that $\mathbf{t}_p$ is not present in the word graph $\mathcal{W}$. The solution is to use not $\mathbf{t}_p$ but a prefix $\mathbf{t}'_p$ that minimizes the edition distance with $\mathbf{t}_p$ and is compatible with $\mathcal{W}$. Therefore, the score of a target translation $\mathbf{t} \equiv \mathbf{t}_p\,\mathbf{t}_s$ is characterized by two functions, the edit cost between the target prefix $\mathbf{t}_p$ and the optimal prefix $\mathbf{t}'_p$ found in the word graph $\mathcal{W}$ and the *a posteriori* probability of $\mathbf{t}_s$ ($Pr(\mathbf{t}_s \mid \mathbf{t}'_p)$). In order to value more significantly those translations that were closer to the user preferences, the list of n-best translations has been prioritized using two criteria: first, the minimum edit cost and then, by the *a* *posteriori* probability.

The algorithm proposed to solve this search problem is an adapted version of the Recursive Enumeration Algorithm (REA) described in (Jiménez and Marzal, 1999) that integrates the minimum edit cost algorithm in the search procedure to deal with words, introduced by the user, that are not present in the word graph. This algorithm consists of two parts:

- Forward search that calculates the 1-best path from the initial state to every state in the word graph $\mathcal{W}$. Paths in the word graph are weighted not only based on their *a posteriori* probability, but also on their edit cost respect to the target sentence prefix.

  To this purpose, ficticious edges have been inserted into the word graph to represent edition operations like insertion, substitution and deletion. These edition operations have been included in the word graph in the following way:

  – **Insertion:** An insertion edge has been "inserted" as a loop for each state in the word graph with unitary cost.

  – **Deletion:** A deletion edge is "added" for each arc in the word graph having the same source and target state than its sibling arc with unitary cost.

  – **Substitution:** Each arc in the word graph is treated as a substitution edge whose edit cost is proportional to the levenshtein distance between the symbol associated with this arc and the word prefix employed to traverse this arc during the search. This substitution cost is zero when the word prefix matches the symbol in the word graph arc.

- Backward search that enumerates candidates for the k-best path along the $(k-1)$-best path. This recursive algorithm defines the next best path that arrives at a given state q as the next best path that reaches q' plus the arc leaving from q' to q. If this next best path arriving at state q' has not been calculated yet, then the next best path procedure is called recursively until a 1-best path is found or no best paths are found.

To reduce the computational cost of the search, the beam-search technique (Ney et al., 1992) has been implemented. During the word graph construction, two beam coefficients were employed to penalize those edges leading to backoff states over those ones arriving at normal states. Finally, a third beam coefficient controls how far in terms of number of edition operations a hypothesis.

4 Experimental results

4.1 Corpus features

The corpus employed to perform experiments was the Xerox corpus (SchlumbergerSema S.A et al., 2001). It involves the translation of technical Xerox manuals from English to Spanish, French and German and vice-versa. Some statistics about the data used for training and test purposes are shown in Table 1.

4.2 Sample session

A TT2 interactive prototype, which uses the searching techniques presented in the previous sections, has been implemented. The user is able to customized this prototype in different ways: number of suggested translations, length in number of words of these suggestions, etc. In the example below, the number of suggestions is five and the length of these suggestions has not been bound.

Example 1 *This example shows the functionality and the interaction between the TT2 prototype and a translator through a translation instance from English to Spanish for a given sentence drawn from the Xerox corpus. For better understanding of this example the reference target sentence is given below:*

Reference target sentence: *`Instalación de controladores de impresora y archivos PPD.`*

Source sentence: *Installing the Printer Drivers and PPDs.*

Hypothesis 0.0: *Instalación del los controladores de impresión y archivos PPD adaptados.*

Hypothesis 0.1: *Instalación del los controladores de impresión y ver los archivos PPD.*

Table 1: Features of Xerox Corpus: training, vocabulary and test sizes measured in thousands of words.
SIM: Currently used *"reversible"* preprocessing.
RAW: Original corpus without preprocess.
PERPLEXITY: Measure how well a language model describes the test set.

	EN / ES		EN / DE		EN / FR	
	RAW	SIM	RAW	SIM	RAW	SIM
TRAINING	600/700	600/700	600/500	500/600	600/700	500/400
VOCABULARY	26 / 30	8 / 11	25 / 27	8 / 10	25 / 37	8 / 19
TEST	8 / 9	8 / 10	9 / 10	11 / 12	11 / 10	12 / 12
PERPLEXITY (3-gram)	107/60	48/33	93/169	51/87	193/135	73/52

Hypothesis 0.2: *Instalación de la los controladores de impresión y archivos PPD adaptados.*

Hypothesis 0.3: *Instalación de la los controladores de impresión y ver los archivos PPD.*

Hypothesis 0.4: *Instalación de la esta los controladores de impresión y ver los archivos PPD.*

User interaction 0: *Hypothesis 0.2 is selected and the cursor is positioned at the beginning of the word "los". Then, the translator would type the character "c", that is, the next character in the reference target sentence.*

Prefix 0: ***Instalación de c***

Hypothesis 1.0: ***Instalación de c*** *los controladores de impresión y archivos PPD adapatados.*

Hypothesis 1.1: ***Instalación de c*** *los controladores de impresión y ver los archivos PPD.*

Hypothesis 1.2: ***Instalación de c*** *esta los controladores de impresión y archivos PPD adapatados.*

Hypothesis 1.3: ***Instalación de c*** *esta los controladores de impresión y ver los archivos PPD.*

Hypothesis 1.4: ***Instalación de c****ontroladores de impresión y fax y en archivos PPD adapatados.*

User interaction 1: *Hypothesis 1.4 is selected and the cursor is positioned between the character "s" and "i" of the word "impresión". Then, the translator would type the next character in the reference target sentence: "o".*

Prefix 1: ***Instalación de controladores de impreso***

Hypothesis 2.0: ***Instalación de controladores de impreso****ra y archivos PPD adaptados.*

Hypothesis 2.1: ***Instalación de controladores de impreso****ra y ver los archivos PPD.*

Hypothesis 2.2: ***Instalación de controladores de impreso****ra/fax y ver los archivos PPD.*

Hypothesis 2.3: ***Instalación de controladores de impreso****ra/fax y archivos PPD adaptados.*

Hypothesis 2.4: ***Instalación de controladores de impreso****ra y fax de CentreWare y ver los archivos PPD.*

User interaction 2: *Hypothesis 2.0 is selected and the cursor is positioned at the end of the word "PPD". The translator would just need to add the character ".".*

Prefix 2: ***Instalación de controladores de impresora y archivos PPD.***

Hypothesis 3.0: ***Instalación de controladores de impresora y archivos PPD.***

Hypothesis 3.1: ***Instalación de controladores de impresora y archivos PPD.****:*

Hypothesis 3.2: ***Instalación de controladores de impresora y archivos PPD.****.*

Hypothesis 3.3: ***Instalación de controladores de impresora y archivos PPD.****..*

Hypothesis 3.4: ***Instalación de controladores de impresora y archivos PPD.****:.*

User interaction 3 : *Hypothesis 3.0 is selected and the user accepts the target sentence.*

Final hypothesis: *Instalación de controladores de impresora y archivos PPD.*

4.3 Translation quality evaluation

The assessment of the techniques presented in section 3 has been carried out using three measures:

1. *Translation Word Error Rate* (TWER): It is defined as the minimum number of word substitution, deletion and insertion operations to convert the target sentence provided by the transducer into the reference translation. Also known as edit distance.
2. *Character Error Rate* (CER): Edit distance in terms of characters between the target sentence provided by the transducer and the reference translation.
3. *Key-Stroke Ratio* (KSR): Number of keystrokes that are necessary to achieve the reference translation plus the acceptance keystroke divided by the number of running characters.
4. *BiLingual Evaluation Understudy* (BLEU) (Papineni et al., 2002): Basically is a function of the k-substrings that appear in the hypothesized target sentence and in the reference target sentence.

These experiments were perfomed with 3-gram transducers based on the GIATI technique. On the leftmost column appears the language pair employed for each experiment, English (En), Spanish (Es), French (Fr) and German (De). The main two central columns compare the results obtained with 1-best translation to 5-best translations. When using 5-best translations, that target sentence out of these five, that minimizes most the correspondent error measure is selected. The results are shown in Table 2.

The best results were obtained between English and Spanish language pairs, in which the human translator would need to type less than 25% of the total reference sentences. In other words, this would result in a theoretically factor of 4 increase in the productivity of human translators. In fact, preliminary subjective evaluations have received positive feedback from professional translators when testing the prototype.

Table 2: Results for the Xerox Corpus comparing 1-best to 5-best translations

	3-gram (1-best)			3-gram (5-best)		
RAW	KSR	CER	TWER	KSR	CER	TWER
En-Es	26.0	29.1	42.3	23.4	24.4	37.2
Es-En	27.4	33.1	50.1	24.1	24.9	42.7
En-Fr	53.7	55.4	77.5	49.3	48.7	70.5
Fr-En	54.0	55.6	74.2	49.9	49.4	68.8
En-De	59.4	61.2	82.4	54.0	54.7	76.6
De-En	52.6	60.3	77.9	48.0	53.4	72.7

Furthermore, in all cases there is a clear and significant improvement in error measures when moving from 1 to 5-best translations. This gain in translation quality dimishes in a log-wise fashion as the number of best translations increases. However, the number of hypotheses should be limited to the user capability to skim through the candidate translations and decide on which one to select.

Table 3 presents the results obtained on a simplified version of the corpus. This simplification consists on tokenization, case normalization and the substitution of numbers, printer codes, etc. by their correspondent category labels.

Table 3: Results for the Xerox Corpus comparing 1-best to 5-best translations

	3-gram (1-best)			3-gram (5-best)		
SIM	WER	CER	BLEU	WER	CER	BLEU
En-Es	31.8	24.7	0.67	26.8	20.3	0.71
Es-En	34.3	27.8	0.62	27.0	20.4	0.69
En-Fr	64.2	48.8	0.43	57.2	42.8	0.45
Fr-En	59.2	48.5	0.42	53.6	42.5	0.45
En-De	72.1	55.3	0.32	65.8	49.1	0.35
De-En	64.7	53.9	0.36	58.4	47.7	0.39

Pair of languages as English and French presents somewhat higher error rates, as is also the case between English and German, reflecting the complexity of the task faced in these experiments.

5 Conclusions and future work

Finite-state transducers have been successfully applied to CAT. These models can be learnt from parallel corpora. The concept of interactive search has been introduced in this paper along with some efficient techniques (word graph derivation and n-best) that solve the parsing problem given a prefix of the target sentence under real-time constraints.

The results show that the 5-best approach clearly improves the quality of the translations, with respect to the 1-best approximation.

The promising results achieved in the first experiments provide a new field in machine translation still to be explored, in which the human expertise is combined with machine translation techniques to increase productivity without sacrifying high-quality translation.

Finally, the introduction of morpho-syntactic information or bilingual categories in finite-state transducers, are topics that leave an open door to future research. As well as some improvements in the search algorithms to reduce the computational cost of finding a path in the word graph with the minimum edit cost.

Acknowledgments

The authors would like to thank all the reasearchers involved in the TT2 project who have contributed to the development of the methodologies presented in this paper.

This work has been supported by the European Union under the IST Programme (IST-2001-32091).

References

Yaser Al-Onaizan, Jan Curin, Michael Jahr, Kevin Knight, John Lafferty, Dan Melamed, Franz J. Och, David Purdy, Noah Smith, and David Yarowsky. 1999. Statistical machine translation: Final report. Workshop on language engineering, Johns Hopkins University, Center for Language and Speech Processing, Baltimore, MD, USA.

Juan C. Amengual, José M. Benedí, Asunción Castano, Antonio Castellanos, Víctor M. Jiménez, David Llorens, Andrés Marzal, Moisés Pastor, Federico Prat, Enrique Vidal, and Juan M. Vilar. 2000. The EuTrans-I speech translation system. *Machine Translation*, 15:75–103.

S. Bangalore and G. Ricardi. 2001. A finite-state approach to machine translation. In *Second Meeting of the North American Chapter of the Association for Computational Linguistics*.

Peter F. Brown, Stephen Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–312.

Francisco Casacuberta, David Llorens, Carlos Martínez, Sirko Molau, Francisco Nevado, Hermann Ney, Moisés Pastor, David Picó, Alberto Sanchis, Enrique Vidal, and Juan M. Vilar. 2001. Speech-to-speech translation based on finite-state transducers. In *International Conference on Acoustic, Speech and Signal Processing*, volume 1. IEEE Press, April.

Francisco Casacuberta, Hermann Ney, Franz J. Och, Enrique Vidal, Juan M. Vilar, Sergio Barrachina, Ismael García-Varea, David Llorens, Carlos Martínez, Sirko Molau, Francisco Nevado, Moisés Pastor, David Picó, and Alberto Sanchís. 2004. Some approaches to statistical and finite-state speech-to-speech translation. *Computer Speech and Language*, 18:25–47.

Víctor M. Jiménez and Andrés Marzal. 1999. Computing the k shortest paths: a new algorithm and an experimental comparison. In J. S. Vitter and C. D. Zaroliagis, editors, *Algorithm Engineering*, volume 1668 of *Lecture Notes in Computer Science*, pages 15–29, London, July. Springer-Verlag.

Kevin Knight and Yaser Al-Onaizan. 1998. Translation with finite-state devices. In E. Hovy D. Farwell, L. Gerber, editor, *Machine Translation and the Information Soup: Third Conference of the Association for Machine Translation in the Americas*, volume 1529, pages 421–437, Langhorne, PA, USA, October. AMTA'98.

Philippe Langlais, George Foster, and Guy Lapalme. 2000. Unit completion for a computer-aided translation typing system. *Machine Translation*, 15(4):267–294.

Hermann Ney, Dieter Mergel, Andreas Noll, and Annedore Paeseler. 1992. Data driven organization for continuous speech recognition. In *IEEE Transactions on Signal Processing*, volume 40, pages 272–281.

Franz J. Och and Hermann Ney. 2000. Improved statistical alignment models. In *ACL00*, pages 440–447, Hong Kong, China, October.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia.

David Picó and Francisco Casacuberta. 2001. Some statistical-estimation methods for stochastic finite-state transducers. *Machine Learning*, 44:121–142, July-August.

SchlumbergerSema S.A, Instituto Tecnológico de Informática, Rheinisch Westfälische Technische Hochschule Aachen Lehrstul für Informatik VI, Recherche Appliquée en Linguistique Informatique Laboratory University of Montreal, Celer Soluciones, Société Gamma, and Xerox Research Centre Europe. 2001. TT2. TransType2 - computer assisted translation. Project technical annex.

Enrique Vidal. 1997. Finite-state speech-to-speech translation. In *Int. Conf. on Acoustics Speech and Signal Processing (ICASSP-97), proc., Vol.1*, pages 111–114, Munich.

Andrew Viterbi. 1967. Error bounds for convolutional codes and a asymtotically optimal decoding algorithm. *IEEE Transactions on Information Theory*, 13:260–269.