Procesamiento del Lenguaje Natural, núm. 35 (2005), pp. 357-363 recibido 29-04-2005; aceptado 01-06-2005

Computer-Assisted Translation using Finite-State Transducers*

Jorge Civera, Elsa Cubel, Antonio L. Lagarda, Francisco Casacuberta y **Enrique Vidal**
Departamento de Sistemas
Infomáticos y Computación
Instituto Tecnológico de Informática
Universidad Politécnica de Valencia
E-46071 Valencia, Spain
{jorcisai,ecubel,alagarda,fcn,evidal}@iti.upv.es

Juan M. Vilar y **Sergio Barrachina**
Departamento de Lenguajes
y Sistemas Informáticos
Universitat Jaume I
E-12071 Castellón de la Plana,
Spain
{jvilar@lsi,barrachi@icc}.uji.es

Resumen: Traducción asistida por ordenador (CAT) es una aproximación alternativa a la traducción automática que integra el conocimiento humano en el proceso de la traducción automática. En este marco, un traductor interactúa con un sistema de traducción que dinámicamente ofrece una lista de traducciones que mejor completan la parte de oración ya traducida. La tecnología de transductores estocásticos de estados finitos se propone para dar apoyo a este sistema CAT. Este sistema fue evaluado en dos tareas reales de diferente complejidad en varias lenguas.
Palabras clave: traducción automática, traducción asistida por ordenador, transductores estocásticos de estados finitos

Abstract: Computer-Assisted Translation (CAT) is an alternative approach to machine translation, that integrates human expertise into the automatic translation process. In this framework, a human translator interacts with a translation system that dynamically offers a list of translations that best completes the part of the sentence already translated. Stochastic finite-state transducer technology is proposed to support this CAT system. The system was assessed on two real tasks of different complexity in several languages.
Keywords: machine translation, computer-assisted translation, stochastic finite-state transducers

1 Introduction

Our current society is characterised by the diversity of the coexistent languages and the necessity of the communication among its citizens. This fact is reflected in a vast number of official institutions (the EU parliament, the Canadian Parliament, UN sessions, Catalan and Basque Parliaments in Spain, etc.) and private companies (user's manuals, newspapers, books, etc.).

The aim of the present work is to develop a Computer-Assisted Translation (CAT) system that will help to solve a very pressing social problem: how to meet the growing demand for high-quality translation. This innovative system embeds a data-driven Machine Translation (MT) engine into an interactive translation environment. In this way, the system combines the best of two paradigms: the CAT paradigm, in which the human translator ensures high-quality output; and the MT paradigm, in which the machine ensures significant productivity gains.

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 approach has two important aspects: the models need to provide adequate completions and they have to do so efficiently under usability constrains. To fulfill these two requirements, Stochastic Finite-State Transducers (SFST) have been selected since they have proved to be able to provide adequate translations (Knight and Al-Onaizan, 1998; Amengual et al., 2000;

[*] This work has been supported by the the Spanish CICYT project TIC2003-08681-C02.

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 interactive translation is explained. Experimental results are presented in Section 4. Finally, some conclusions and future work are exposed in Section 5.

2 Machine translation with finite-state transducers

In a probabilistic framework, given a source sentence $\mathbf{s}$, the goal of MT is to find a target sentence $\hat{\mathbf{t}}$ that:

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

The joint distribution $\Pr(\mathbf{t}, \mathbf{s})$ can be modelled by a SFST $\mathcal{T}$ (Picó and Casacuberta, 2001):

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

SFSTs have been successfully applied into many translation tasks (Amengual et al., 2000; Casacuberta et al., 2001). Furthermore, there exist efficient search algorithms like Viterbi (Viterbi, 1967) for the best path.

One possible way of inferring SFSTs is the Grammatical Inference and Alignments for Transducer Inference (GIATI) technique (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), 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 (Eq. 1). $\mathbf{t}_p$ represents a prefix of the target sentence obtained as a result of the interaction between the human translator and the machine translation system, and $\mathbf{t}_s$ represents the translation offered by the system given $\mathbf{t}_p$

Then, given $\mathbf{t}_p\hat{\mathbf{t}}_s$, the CAT cycle proceeds by letting the user establish a new, longer acceptable prefix. To this end, he or she has to accept a part ($\mathbf{a}$) of $\mathbf{t}_p\hat{\mathbf{t}}_s$ (or, more typically, just a prefix of $\hat{\mathbf{t}}_s$). After this point, the user may type some keystrokes ($\mathbf{k}$) in order to amend some remaining incorrect parts. Therefore, the new prefix (typically) encompasses $\mathbf{t}_p$ followed by the accepted part of the system suggestion, $\mathbf{a}$, plus the text, $\mathbf{k}$, entered by the user. Now this prefix, $\mathbf{t}_p\,\mathbf{a}\,\mathbf{k}$, becomes a new $\mathbf{t}_p$, thereby starting a new CAT prediction cycle.

Ergonomics and user preferences dictate exactly when the system can start its new cycle, but typically, it is started after each user-entered word or even after each new user keystroke.

Perhaps the simplest formalization of the process of hypothesis suggestion of a CAT system is as follows. Given a source text $\mathbf{s}$ and a user validated *prefix* of the target sentence $\mathbf{t}_p$, search for a *suffix* of the target sentence that maximises the *a posteriori* probability over all possible suffixes:

$$\hat{\mathbf{t}}_s = \operatorname*{argmax}_{\mathrm{t}_s} \Pr(\mathbf{t}_s \mid \mathbf{s}, \mathbf{t}_p) \ . \quad (3)$$

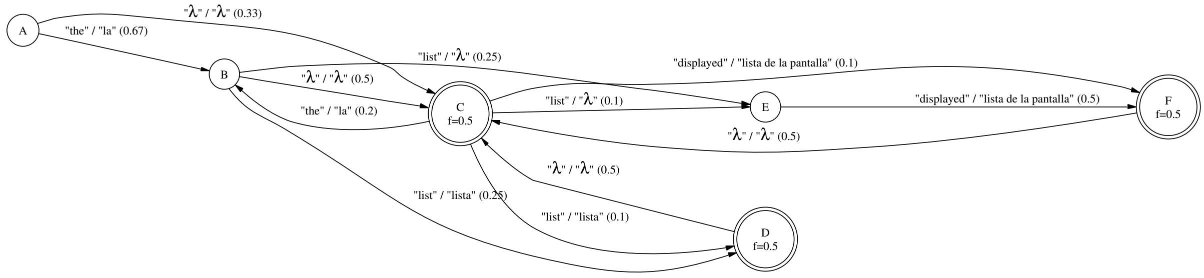

Figure 1: A SFST inferred from a parallel corpus composed by two pairs of sentences: *the list # la lista* and *the displayed list # la lista de la pantalla*

Taking into account that $\Pr(\mathbf{t}_p \mid \mathbf{s})$ does not depend on $\mathbf{t}_s$, we can write:

$$\hat{\mathbf{t}}_s = \underset{\mathbf{t}_s}{\operatorname{argmax}} \Pr(\mathbf{t}_p \mathbf{t}_s \mid \mathbf{s}) \ , \qquad (4)$$

Eq. 4 is similar to Eq. 1, but here the maximisation is carried out over a set of suffixes, rather than full sentences as in Eq. 1. This joint distribution can be adequately modeled by means of SFSTs (Civera et al., 2004).

The solution to this maximisation problem has been devised in two phases. The first phase copes with the extraction of a word graph $\mathcal{W}$ from a SFST $\mathcal{T}$ given a source sentence $\mathbf{s}$. In a second phase, the search of the best translation suffix is performed over the word graph $\mathcal{W}$ given a prefix $\mathbf{t}_p$ of the target sentence.

3.1 Word graph derivation

A word graph $\mathcal{W}$ is a compact representation of all the possible translations that a SFST can produce from a given source sentence $\mathbf{s}$. In fact, the word graph could be seen as a kind of weighted finite-state automaton in which the probabilities are not normalized.

There are a couple of minor issues to deal with in this construction. On the one hand, the output symbol for a given transition could be an empty string (which are represented by λ in Figures 1 and 2) or could contain more than one word. In this latter case, auxiliary states were created in order to assign only one word for each transition and simplify the posterior search procedure. On the other hand, it is possible to have words in the input sentence that do not belong to the input vocabulary in the SFST. This problem is solved with the introduction of a special generic "unknown word" in the input vocabulary of the SFST.

This process can be better understood through a simple example. Assume that we have to translate the sentence "the list" using the SFST of Figure 1. The resulting word graph is shown in Figure 2. Intuitively, the word graph generated retains those transitions in the SFST that were compatible with the source sentence along with their transition probability and output symbol(s). Those states that are reached at the end of the parsing process of the source sentence, over the SFST, are considered final states (as well as those states reachable with λ-transitions from them).

Once the word graph is constructed, it can be used to find the best completion for the part of the translation typed by the human translator. Note that the word graph depends only on the input sentence, so it is used repeatedly for finding the completions of all the different prefixes provided by the user.

3.2 Search of translations given a prefix of the target sentence

Ideally, the search problem consists in finding the target suffix $\mathbf{t}_s$ that maximises the *a posteriori* probability given a prefix $\mathbf{t}_p$ of the target sentence and the input sentence $\mathbf{s}$, as described in Eq. 4. To simplify this search, it will be divided into two steps or phases. The first one would deal with the parsing of $\mathbf{t}_p$ over the word graph $\mathcal{W}$. This parsing procedure would end reaching a set of states Q_p that define paths from the initial state whose associated translations include $\mathbf{t}_p$. To clarify this point, it is important to note that each state q in the word graph defines a set of translation prefixes P_q. This set of translation prefixes is obtained from the concatenation of the output symbols of the different paths that reach this state q from the initial state. Therefore, the set P_q of each state in Q_p includes $\mathbf{t}_p$.

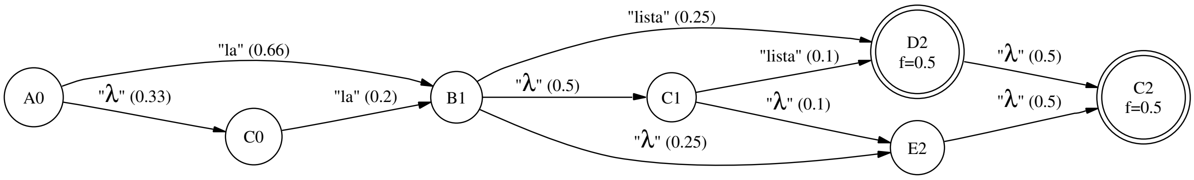

Figure 2: Word graph resulting from the SFST in Figure 1 given the source sentence "the list". States are labeled with the SFST state-id and the position of the word in the source sentence being parsed to reach that SFST state.

In practice, 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 is the *most similar* to $\mathbf{t}_p$ in some string distance metric. The metric that will be employed is the well-known minimum edit distance based on three basic edit operations: insertion, substitution and deletion. Therefore, this phase needs to be redefined in terms of the search of those states in $\mathcal{W}$ whose set P_q contains $\mathbf{t}'_p$, that is, the set of states Q_p. It should be remarked that $\mathbf{t}'_p$ is not unique, but there exist a set of prefixes in $\mathcal{W}$ whose minimum edit distance to $\mathbf{t}_p$ is the same and the lowest possible.

The computation of Q_p, given a translation prefix $\mathbf{t}_p$, is efficiently carried out by applying an adapted version of the error-correcting algorithm for regular grammars over the word graph $\mathcal{W}$. This algorithm returns the minimum edit cost $c(q)$ with respect to $\mathbf{t}_p$ for each state q in $\mathcal{W}$. To be more precise, this minimum edit cost is the lowest minimum edit cost between $\mathbf{t}_p$ and the set of prefixes P_q of each state q. Finally, Q_p is defined as those states that minimize the minimum edition cost.

The second phase would be the search of the most probable translation suffix from any of the states in Q_p according to the Viterbi approach (Viterbi, 1967). Finally, the complete search procedure extracts a translation from the word graph whose prefix is $\mathbf{t}_p$ and its remaining suffix is the resulting translation suffix $\mathbf{t}_s$.

4 Experimental framework and results

The SFST models introduced in the previous sections were assessed through some series of experiments with two different corpora that were acquired and preprocessed in the framework of the TransType2 (TT2) project (SchlumbergerSema S.A. et al., 2001). In this section, these corpora, the assessment metrics and the results are presented.

Table 1: The "XRCE" corpus English(En) to Spanish(Sp), German(Ge) and French(Fr). Trigrams models were used to compute the test perplexity. (K denotes ×1.000, and M denotes ×1.000.000).

		En/Sp	En/Ge	En/Fr
Train	Sent. pairs (K)	56	49	53
	Run. words (M)	0.6/0.7	0.6/0.5	0.6/0.7
	Vocabulary (K)	26/30	25/27	25/37
Test	Sentences (K)	1.1	1.0	1.0
	Run. words (K)	8/9	9/10	11/10
	Perplexity	107/60	93/169	193/135

4.1 XRCE and EU corpora

Two bilingual corpora extracted from different semantic domains were used in the evaluation of the CAT system described. The language pairs involved in the assessment were English-Spanish, English-French and English-German.

The first corpus, namely *XRCE* corpus, was obtained from a miscellaneous set of printer user manuals. The main characteristics of this corpus are summarised in Table 1.

It is important to remark that the English manuals are different in each pair of languages. The size of the vocabulary in the training set is about 25.000 words in most of the language pairs what can be considered to be a broad lexicon. In the test set, even though all test sets have similar size, the perplexity varies abruptly over the different language pairs.

The second dataset was compiled from the Bulletin of the European Union, which exists in the 11 official languages of the European Union. This corpus is known as the *EU* corpus. It is publicly available on the Internet. A summary of the features of this corpus is presented in Table 2.

Table 2: The “EU” corpora English(En) to Spanish(Sp), German(Ge) and French(Fr). Trigrams models were used to compute the test perplexity (K denotes ×1.000, and M denotes ×1.000.000).

		En/Sp	En/Ge	En/Fr
Train	Sent. pairs (K)	214	223	215
	Run. words (M)	5.9/6.6	6.5/6.1	6.0/6.6
	Vocabulary (K)	84/97	87/153	85/91
Test	Sentences	800	800	800
	Run. words (K)	20/23	20/19	20/23
	Perplexity	96/72	95/153	97/71

The size of the vocabulary of this corpus is at least three times larger than that of the *XRCE* corpus. These numbers together with the amount of running words and sentences reflect the challenging nature of this task. However, the perplexity of the *EU* test set is similar to that of the *XRCE.* This phenomenon can be intuitively explained through the more uniform grammatical structure of the sentences in the *EU* corpus.

4.2 Translation quality evaluation

The assessment of the CAT system presented has been carried out based on two measures:

1. *Translation Word Error Rate* (TWER). It is defined as the minimum number of word substitution, deletion and insertion operations required to convert the target sentence provided by the translation system into the reference translation, divided by the number of words of the reference translation.

   This metric is employed to evaluate the quality of the complete translations offered by the system when no prefix is taken into consideration, that is, no interaction with the user is assumed.

2. *Key-Stroke Ratio* (KSR). Number of interactions, as the sum of mouse actions (to select **a**) and keystrokes (to type **k**), that are necessary to achieve the reference translation plus the final translation-acceptance keystroke divided by the number of characters of the reference translation.

   KSR reflects the ratio between the number of interactions of a fictitious user when translating a given text using a CAT system compared to the number of interactions, which this user would need to translate the same text without using a CAT system. Thus, this measure gives a clear idea of the amount of work that a fictitious user would be saving when using a CAT system.

Table 3: Results for the *XRCE* corpus based on 3-gram language models

XRCE	KSR	TWER
En-Sp	24.4	30.8
En-Ge	52.1	70.7
En-Fr	48.7	63.2

Table 4: Results for the *EU* corpus based on 5-gram language models

EU	KSR	TWER
En-Sp	38.9	54.5
En-Ge	45.6	64.2
En-Fr	35.7	51.8

4.3 Experimental results

These experimental results were obtained with GIATI transducers based on smooth trigram language models for the *XRCE* corpus (see Table 3) and smooth 5-gram language models for the *EU* corpus (see Table 4).

The translation metrics presented in the previous section were calculated on the test set for the three pairs of languages, as it is shown on the left-most column of Tables 3 and 4.

Analysing the results accomplished in the *XRCE* corpus, it is observed that the TWER and KSR rates for English/Spanish language pair are substantially lower than those obtained in the rest of language pairs. A possible reason behind the error rate discrepancies between English/Spanish pair with respect to English/German and English/French pairs could be found in the perplexity differences shown in Table 1. For example, the Spanish test perplexity is significantly lower than that of the rest of languages and this fact is transformed into better translation results. Another reason for the outperforming results of the English/Spanish pair comes from the hand of the random partition in training and test datasets, that could have been resulted in a simpler test set for the English/Spanish pair.

This rational is compatible with the results obtained for the *EU* corpus. In these results, English/Spanish pair exhibit similar error rates to those of the English/French pair, but significantly better than those of the English/German pair. This same tendency is followed by perplexity values appearing in Table 2. As it can be observed, the German language seems to be more complex than the other languages and this is reflected in the translation results.

As the reader may have noticed, TWER results in both corpora are not sufficiently good to support a pure machine translation system based on SFSTs inferred by the GIATI technique. However, if the system is evaluated as a CAT system (KSR), a productivity gain is clearly manifested. For example in the *XRCE* corpus, translating from English into Spanish, the user would only need to perform 24.4% of the interactions that would be required without this CAT system. On the other hand, the KSR results are about 50% for the English/French and English/German pair. Even though in these cases, the number of interactions is halved with respect to the effort that would entail to translate the same test set without a CAT system.

In the *EU* corpus, the best KSR result was obtained for the English/French language pair, followed by the results in the English/Spanish language pair, and finally the worst result was achieved in English/German language pair. Despite the important difference in size between *XRCE* and *EU*, the results are similar and for some language pairs even lower in the *EU* corpus. The perplexity numbers on both corpora partially explain these results being somewhat correlated with the TWER and KSR results. For instance, the English/French language pair presents lower perplexity and better results in the *EU* corpus than in the *XRCE* corpus.

5 Conclusions and future work

In the present work, SFSTs have been revisited and applied to CAT. In this case, SFSTs that are easily learnt from parallel corpora were inferred by the GIATI technique, which was briefly reviewed. Moreover, the concept of interactive search has been introduced in this paper along with some well-known techniques, i.e. error-correcting parsing, that allow the calculation of the suffix translation that better completes the prefix written by the user. It is fundamental to remember that usability and response-time are vital features for CAT systems. CAT systems need to provide translation suffixes after each user interaction and this imposes the necessity of efficient algorithms to solve the search problem.

As it was preempted in the introduction, current machine translation systems are not able to provide high quality translations and SFST techniques are not an exception. Nevertheless, the capability of SFSTs to suggest translation suffixes that aid a human translator to increase his or or her productivity in a CAT framework should not be neglected. The results presented on two different corpora seem to support the idea of the benefits of the incorporation of machine translation techniques into the translation process to reduce human translator effort without sacrificing high-quality translations.

Finally, the introduction of morpho-syntactic information, bilingual categories or more powerful smoothing techniques on the source and target languages, in SFSTs, are topics still to be explored in future research.

References

Amengual, Juan C., 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.

Bangalore, S. 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.*

Brown, Peter F., 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.

Casacuberta, Francisco, 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.

Casacuberta, Francisco, 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.

Civera, J., J. M. Vilar, E. Cubel, A. L. Lagarda, S. Barrachina, E. Vidal, F. Casacuberta, D. Picó, and J. González. 2004. From machine translation to computer assisted translation using finite-state models. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP04)*, Barcelona.

Knight, Kevin 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.

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

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

SchlumbergerSema S.A., Intituto 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.

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