Finite-state models for computer assisted translation

Elsa Cubel[1] and **Jorge Civera** [2] and **Juan M. Vilar** [3] and **Antonio L. Lagarda**[1] and **Francisco Casacuberta**[1] and **Enrique Vidal**[1] and **David Picó**[2] and **Jorge González**[2] and **Luis Rodríguez**[1]

Abstract. Current methodologies for automatic translation cannot be expected to produce high quality translations. However, some techniques based on these methodologies can increase the productivity of human translators. The basis of one of these methodologies are finite-state transducers, which are adequate models for computer assisted translation. These models have proved its efficiency in many pattern recognition and artificial intelligence tasks such as speech recognition, handwriting recognition and machine translation for specific domains.

These finite-state models present some advantages. On the one hand, finite-state models can be learnt from bilingual corpus to infer transducers. On the other hand, there are well-known and efficient algorithms to perform the parse of the best translation according to these models (e.g. Viterbi search). In this paper, the concept of interactive search will be introduced along with some efficient techniques that solve the problem of producing a translation given a sentence in the source language and a prefix (from the output sentence) typed by the user. Needless to say that this system must run under real-time constraints to be useful for human translators.

This approach has been tested on a corpus of printer manuals and the first results reflect that human translators would only need to type the 25% of the characters of the whole translated text, increasing in this way their throughput and reducing their effort.

1 INTRODUCTION

Present translation technology has not been able to keep pace with the demands for high-quality translation. An alternative way to take advantage of the technologies developed is to use them in order to help human translators, as it is explained in [9]. In this paper, an approach that significantly increases the translator productivity is presented, having an enormous commercial potential. This work has been carried out in the TransType2 [11] project.

The aim of TT2 is the development of 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.

The innovative solution proposed by TT2 is to embed a data driven Machine Translation (MT) engine within 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 [3], in which the machine ensures significant productivity gains.

This approach has two important aspects: the models need to provide adequate completions and they have to do so efficiently. To fulfill these two requirements, Stochastic Finite-State Transducers (SFST) have proved in the past to be able to provide adequate translations [13, 7, 1, 6, 2]. In addition, it is shown in this paper that efficient parsing algorithms can be easily adapted in order to provide translations.

Moreover, hybrid finite-state and statistical translation techniques can be used to produce efficient SFSTs. The learning of SFST are improved by using statistical word-aligned training pairs together with n-gram language models [5]. The parsing (search) with SFSTs is carried out by the Viterbi algorithm adapted to the CAT framework.

The rest of the paper is structured as follows. Next 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 shown 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:

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

A *stochastic finite-state transducer* (SFST) is a finite-state network whose transitions are labeled by three items:

1. an input symbol (a word from the input vocabulary);
2. an output string (a sequence of words from the output vocabulary) and
3. a transition probability.

Figure 1 shows a small SFST for Spanish to English translation.

SFSTs are models that can be used to estimate the joint distribution $\Pr(\mathbf{t}, \mathbf{s})$ [10]. Given a SFST $\mathcal{T}$,

$$\hat{\mathbf{t}} \approx \underset{\mathbf{t}}{\text{argmax}}\, \Pr_{\mathcal{T}}(\mathbf{t}, \mathbf{s}) \tag{2}$$

SFSTs have been successfully applied into many translation tasks [13, 1, 6]. Current parsers for SFSTs produce a target sentence from a source sentence using the Viterbi algorithm [14] .

In the CAT paradigm, the decoder must produce one (or n-) best translation prediction(s) given a source sentence and a prefix of the sentence in the target language. Given a SFST $\mathcal{T}$, a source sentence $\mathbf{s}$ and a prefix of the source sentence $\mathbf{t}_p$, the goal is to find a suffix of the target sentence $\hat{\mathbf{t}}_s$:

[1] Instituto Tecnológico de Informática, Universidad Politécnica Valencia, E-46071 Valencia, Spain, email: tt2iti@iti.upv.es

[2] Departamento de Sistemas Informáticos y Computación, Universidad Politécnica de Valencia, E-46071 Valencia, Spain

[3] Departamento de Lenguajes y Sistemas Informáticos, Universitat Jaume I, E-12071 Castellón de la Plana, Spain

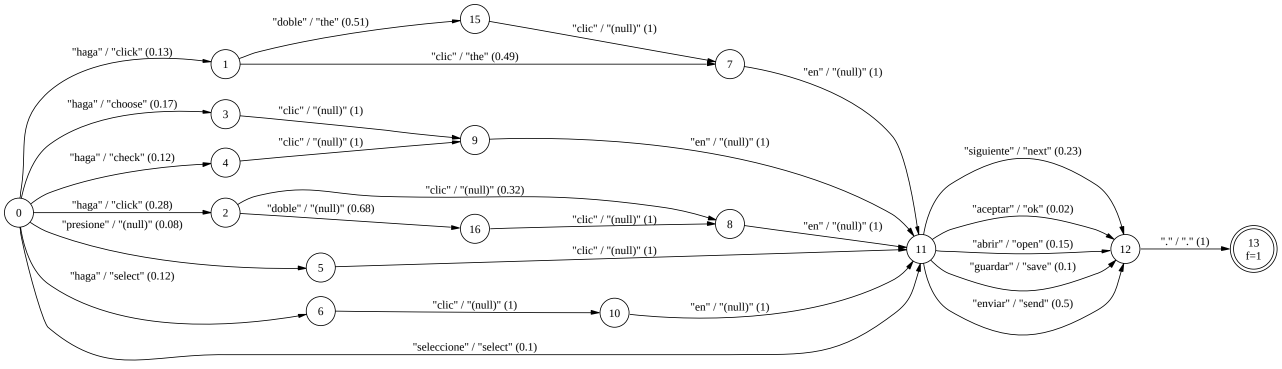

Figure 1. A transducer inferred from a parallel corpus

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

This equation is similar to the one for general translation but in this case, the optimization is performed on a set of target suffixes rather than the set of whole target sentences.

The inference of such SFSTs can be carried out by the Grammatical Inference and Alignments for Transducer Inference (GIATI) technique (the previous name of this technique was MGTI - Morphic-Generator Transducer Inference) [4]. 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.
2. Inferring a (stochastic) regular grammar. Typically, smoothed n-gram is inferred from the set of samples of strings obtained in the previous step.
3. Transforming the inferred regular grammar into a transducer. The symbols associated to the grammar rules are replaced by source/target symbols, thereby converting the grammar inferred in the previous step into a transducer.

3 INTERACTIVE SEARCH

In the previous section the training process undergone to generate a SFST $\mathcal{T}$ from a parallel corpus has been described. The aim of interactive search is to find the target sentence (or set of target sentences) that maximizes the *a posteriori* probability given a source sentence **s**, and minimize the edition cost between the target sentence $\hat{\mathbf{t}}$ and the prefix of the target sentence $\mathbf{t}_p$, introduced by the human translator.

The solution to this problem has been divided 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 **s**. A word graph is a structure with all possible translations that the SFST can produce from the source sentence. The size of this word graph, in terms of number of edges and states is smaller than the original SFST $\mathcal{T}$.

In a second phase, the search of the best translations is performed over the word graph $\mathcal{W}$, instead of the initial SFST $\mathcal{T}$. This search can be carried out taking into account not only the *a posteriori* probability of a given translation $\hat{\mathbf{t}}$, but also the minimum edition cost. In MT, the search is implemented efficiently by using Viterbi algorithm. This algorithm can be adapted for the purpose of CAT, as shown in the following sections.

3.1 Word graph derivation

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

An example of word graph is shown in Figure 2. This word graph has been obtained from the SFST represented in Figure 1 and the source sentence *"haga click en siguiente"*(click next).

As it appears in the previous example, a word graph is a finite-state network whose transitions are labeled by a target word (the same than the target word in the SFST) and a transition probability (the same than the probability in the SFST).

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

Finally, there are a couple of minor issues to deal with in the construction of the word graph. On the one hand, the output symbol for a given transition could be empty ("(null)") or could contain more than one word. Since the generated word graph is not deterministic, the inclusion of empty labelled transitions coming from the SFST is integrated easily. In case of transitions with more than one word, auxiliary states were created in order to assign only one word for each transition. 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 "unknown word" in the input vocabulary of the SFST.

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

Once the word graph has been generated, the next step addresses the search for the n*-best translations* in the word graph for a given source sentence (n is a number given by the human translator). These n translations will be the most probable translations for the source sentence, i.e. the most probable paths in the word graph.

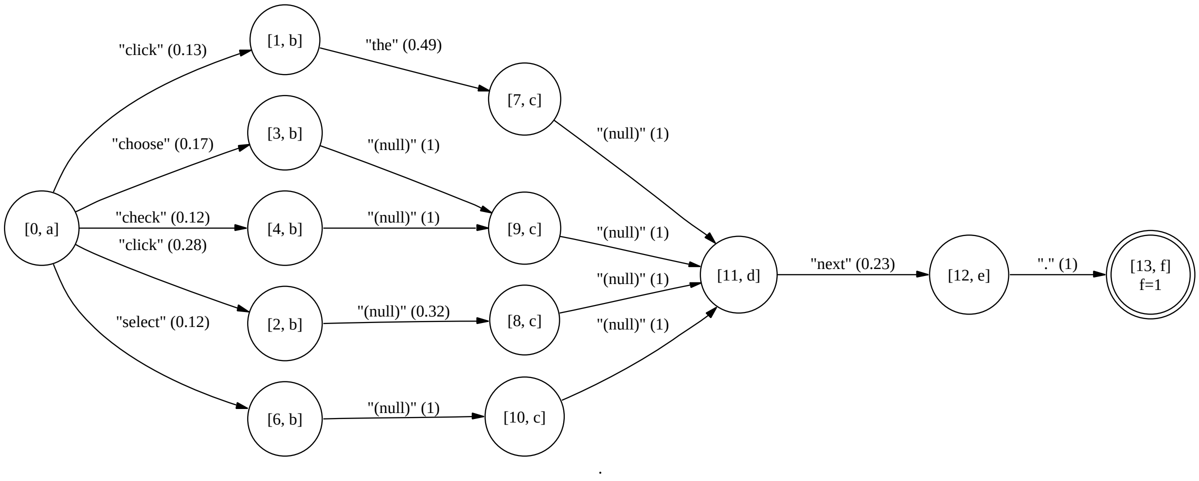

Figure 2. Word graph resulting from the SFST in Figure 1 and the source sentence “haga click en siguiente”

The application of this type of search is aimed at the core of the CAT. At first, given only the source sentence, the human translator is provided with the list of n-best translations found in the word graph. Then, the human translator will proceed to select a prefix of one of these n-best translations as correct, changing it if is necessary. This new prefix along with the source sentence will generate a new set of best suffix translations, that will be again modified by the human translator. This process is repeated as many times as necessary to achieve the desired final translation.

This dynamic adaptation requires to search a target suffix in the word graph that maximizes the *a posteriori* probability given a prefix of the target sentence. This operation implies to find prefixes in the word graph that are compatible with the prefix suggested by the user. This compatibility is not always possible, so the target prefix is approximated through the minimum edition cost.

The beam-search [12] technique has been implemented to reduce the computational cost of the search. During the word graph construction, two beam coefficients were employed. One to penalize those transitions leading to backoff states and another beam coefficient less restrictive for those transitions arriving at normal states. Finally, a third beam coefficient controls how far, in terms of number of edition operations, the current hypothesis could be from the best hypothesis. The application of this technique is imperative because of the real-time constraints under which the prototype is required to run.

3.3 Sample session

In this section, a TT2 interactive prototype [8], which uses the searching techniques presented in the previous sections, is presented. The user can customize 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 set to five and the length of these suggestions has not been bounded.

Example 1 *This example shows the functionality and the interaction between the TT2 prototype and a translator for a given sentence drawn from the Xerox corpus. The reference target sentence is given below:*

Reference target sentence: *Los requisitos de hardware mínimos para los controladores de impresora para Macintosh son:*

Source sentence: *The minimum hardware requirements for the Macintosh printer drivers are:*

Hypothesis 0.0: *El requisitos de hardware mínimos para esta:*

Hypothesis 0.1: *Los requisitos de hardware mínimos para esta:*

Hypothesis 0.2: *El requisitos para esta:*

Hypothesis 0.3: *El requisitos de hardware mínimos para los:*

Hypothesis 0.4: *El requisitos de hardware mínimos para el:*

User interaction 0: *Hypothesis 0.1 would be selected and the cursor would be positioned at the begining of the word ”esta”. Then, the translator would type ”l”, that is, the next character in the reference target sentence.*

Prefix 0: ***Los requisitos de hardware mínimos para l***

Hypothesis 1.0: ***Los requisitos de hardware mínimos para l****os:*

Hypothesis 1.1: ***Los requisitos de hardware mínimos para l****os:*

Hypothesis 1.2: ***Los requisitos de hardware mínimos para l****a:*

Hypothesis 1.3: ***Los requisitos de hardware mínimos para l****as:*

Hypothesis 1.4: ***Los requisitos de hardware mínimos para l****os controladores de impresora propios de CentreWare son:*

User interaction 1: *Hypothesis 1.4 would be selected and the cursor would be positioned between the character ”p” and ”r” of the word ”propios”. Then, the translator would type ”a”, that is, the next character in the reference target sentence.*

Prefix 1: ***Los requisitos de hardware mínimos para los controladores de impresora pa***

Hypothesis 2.0: ***Los requisitos de hardware mínimos para los controladores de impresora pa*** *de CentreWare son:*

Hypothesis 2.1: ***Los requisitos de hardware mínimos para los controladores de impresora pa*** *de CentreWare:*

Hypothesis 2.2: ***Los requisitos de hardware mínimos para los controladores de impresora pa*** *de CentreWare son:*

Hypothesis 2.3: ***Los requisitos de hardware mínimos para los controladores de impresora pa*** *de CentreWare:*

Hypothesis 2.4: ***Los requisitos de hardware mínimos para los controladores de impresora pa*** *propios de CentreWare son:*

User interaction 2: *No hypothesis would be selected in this case. The translator would need to type the following character ”r” in the reference target sentence. This last interaction would be repeated several times, since no hypothesis matches the reference target sentence. The translator would end up typing ”para Macintosh ”.*

Prefix 2: ***Los requisitos de hardware mínimos para los controladores de impresora para Macintosh***

Hypothesis 3.0: ***Los requisitos de hardware mínimos para los controladores de impresora para Macintosh*** *son:*

Hypothesis 3.1: ***Los requisitos de hardware mínimos para los controladores de impresora para Macintosh*** *CentreWare son:*

Hypothesis 3.2: ***Los requisitos de hardware mínimos para los controladores de impresora para Macintosh*** *de CentreWare son:*

Hypothesis 3.3: ***Los requisitos de hardware mínimos para los controladores de impresora para Macintosh****:*

Hypothesis 3.4: ***Los requisitos de hardware mínimos para los controladores de impresora para Macintosh*** *CentreWare:*

User interaction 3 : *Hypothesis 3.0 would be selected and the user would accepted the target sentence.*

Final hypothesis: *Los requisitos de hardware mnimos para los controladores de impresora para Macintosh son:*

4 EXPERIMENTAL RESULTS

4.1 Corpus features

The corpus employed for the experiments is the Xerox corpus. It involves the translation of technical Xerox manuals from English to Spanish, French and German and vice versa. The data used for training and test purposes are shown in Table 1.

Table 1. Features of the Xerox Corpus: training, vocabulary and test sizes are measured in thousands of words

	EN / ES	EN / DE	EN / FR
TRAINING	600/700	600/500	600/700
VOCABULARY	26 / 30	25 / 27	25 / 37
TEST	8 / 9	9 / 10	11 / 10
PERPLEXITY (3gram)	107/60	93/169	193/135

4.2 Translation quality evaluation

The assessment of the techniques presented in section 3 has been carried out using the *Key-Stroke Ratio* (KSR) that counts the number of key-strokes needed to achieve the reference target sentence plus the acceptance key-stroke divided by the number of running characters.

$$KSR = \frac{number\ of\ key\ strokes+1}{number\ of\ characters}$$

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 one translation and with five translations for a given source sentence. In the latter case, the target sentence out of those five translations that minimizes the KSR was selected. The results are shown in Table 2.

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

	KSR (1-best)	KSR (5-best)
En-Es	28.3	25.4
Es-En	34.5	30.7
En-Fr	58.7	53.9
Fr-En	58.9	54.4
En-De	67.5	61.4
De-En	59.5	54.4

The best results were obtained between English and Spanish language pairs, in which the human translator would only need to type 25% of the total characters of the reference sentences. In other words, this would result in a factor of 4 increasement in the productivity of human translators.

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.

Some pairs of languages such as English and French present somewhat higher error rates, as it is also the case between English and German, reflecting the complexity of the task.

5 CONCLUSIONS AND FUTURE WORK

As the results have shown, finite-state transducers are adequate models for computer assisted translation. Besides, they can be easily learnt from a parallel corpus.

The concept of interactive search has been introduced in this paper along with some efficient techniques (word graph derivation and n-best hypothesis generation) that solve the parsing problem, taking into account the prefix concept, under real-time contraints.

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 and/or bilingual categories in finite-state transducers are topics that leave an open door to future research.

ACKNOWLEDGEMENTS

The authors would like to thank the reasearchers involved in the TT2 project who have developed the methodologies that are presented in this paper.

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

REFERENCES

[1] 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, 'The EuTrans-I Speech Translation System', *Machine Translation*, **15**, 75–103, (2000).

[2] S. Bangalore and G. Ricardi, 'A finite-state approach to machine translation', in *Second Meeting of the North American Chapter of the Association for Computational Linguistic s*, (2001).

[3] Peter F. Brown, John Cocke, Stephen Della Pietra, Vincent J. Della Pietra, Frederick Jelinek, John D. Lafferty, Robert L. Mercer, and Paul S. Rossin, 'A Statistical Approach to Machine Translation', *Computational Linguistics*, **16**(2), 79–85, (1990).

[4] F. Casacuberta, H. Ney, F. J. Och, E. Vidal, J. M. Vilar, S. Barrachina, I. García-Varea, D. Llorens, C. Martínez, S. Molau, F. Nevado, M. Pastor, D. Picó, and A. Sanchís, 'Some approaches to statistical and finite-state speech-to-speech translation', *Computer Speech and Language*, **18**, 25–47, (2004).

[5] Francisco Casacuberta, 'Inference of finite-state transducers by using regular grammars and morphisms', in *Grammatical Inference: Algorithms and Applications*, ed., A.L. Oliveira, volume 1891 of *Lecture Notes in Computer Science*, 1–14, Springer-Verlag, (2000). 5th International Colloquium Grammatical Inference -ICGI2000-. Lisboa. Portugal.

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

[7] Kevin Knight and Yaser Al-Onaizan, 'Translation with finite-state devices', in *Machine Translation and the Information Soup: Third Conference of the Association for Machin e Translation in the Americas*, ed., E. Hovy D. Farwell, L. Gerber, volume 1529, pp. 421–437, Langhorne, PA, USA, (October 1998). AMTA'98.

[8] Antonio Lagarda, Luis Rodríguez, Elsa Cubel, Enrique Vidal, and Francisco Casacuberta, 'TransType2. Un sistema de ayuda a la traducción', in *Proceedings of the XIX Congreso de la Sociedad Espanola para el Procesamiento del Lenguaje Natural*, pp. 345–346, Alcalá de Henares, Spain, (September 2003).

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

[10] David Picó and Francisco Casacuberta, 'Some statistical-estimation methods for stochastic finite-state transducers', *Machine Learning*, **44**, 121–142, (July-August 2001).

[11] 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. TT2. TransType2 - Computer Assisted Translation. Project Technical Annex, 2001.

[12] C. Tillman and H. Ney, 'Word re-ordering and dynamic programming beam search algorithm for statistical machine translation', *Computational Linguistics*, **29**, 97–133, (2003).

[13] Enrique Vidal, 'Finite-State Speech-to-Speech Translation', in *Int. Conf. on Acoustics Speech and Signal Processing (ICASSP-97), proc., Vol.1*, pp. 111–114, Munich, (1997).

[14] Andrew Viterbi, 'Error bounds for convolutional codes and a asymtotically optimal decoding algorithm', *IEEE Transactions on Information Theory*, **13**, 260–269, (1967).