Some approaches to statistical and finite-state speech-to-speech translation

F. Casacuberta a,*, H. Ney b, F.J. Och b, E. Vidal a, J.M. Vilar d, S. Barrachina c, I. García-Varea a, D. Llorens d, C. Martínez a, S. Molau b, F. Nevado a, M. Pastor a, D. Picó a, A. Sanchis a, C. Tillmann b

a Departament de Sistemes Informàtics i Computació, Institut Tecnològic d’Informàtica, Universitat Politècnica de València, Camino de Vera, s/n, Valencia 46071, Spain
b Lehrstuhl für Informatik VI, RWTH Aachen, University of Technology, Aachen D-52056, Germany
c Departament d’Enginyeria i Ciències dels Computadors, Universitat Jaume I, Castelló de la Plana 12071, Spain
d Departament de Llenguatges i Sistemes Informàtics, Universitat Jaume I, Castelló de la Plana 12071, Spain

Received 3 December 2001; received in revised form 14 April 2003; accepted 20 May 2003

Abstract

Speech-input translation can be properly approached as a pattern recognition problem by means of statistical alignment models and stochastic finite-state transducers. Under this general framework, some specific models are presented. One of the features of such models is their capability of automatically learning from training examples. Moreover, the stochastic finite-state transducers permit an integrated architecture similar to one used in speech recognition. In this case, the acoustic models (hidden Markov models) are embedded into the finite-state transducers, and the translation of a source utterance is the result of a (Viterbi) search on the integrated network. These approaches have been followed in the framework of the European project EuTRANS. Translation experiments have been performed from Spanish to English and from Italian to English in an application involving the interaction of a customer with a receptionist at the frontdesk of a hotel.

© 2003 Elsevier Ltd. All rights reserved.

Keywords: Speech-input translation; Statistical alignment models; Stochastic finite-state transducers

* Corresponding author. Tel.: +34-96-387-7241; fax: +34-96-387-7239.
E-mail address: fcn@iti.upv.es (F. Casacuberta).
1. Introduction

The statistical framework has traditionally proved very useful in most pattern recognition problems. This is particularly the case of Automatic Speech Recognition (ASR), where the statistical approaches have led to the development of quite successful practical systems. Recent works clearly suggest that these approaches can be naturally extended for the development of speech-input machine translation (MT) systems (Vidal, 1997; Ney, 1999; Ney, 2000; Amengual et al., 2000; Ney et al., 2000; Tillamn et al., 2000). Under this paradigm, a speech-input MT system is automatically built from sufficiently large training sets of text translation examples along with adequate speech (acoustic) data of the source language. Following the great success achieved in ASR, we expect that the use of these techniques can dramatically reduce development costs in many speech-input MT applications, provided the costs of obtaining the required training sets can be kept reasonably low.

The problem of speech-input MT can be properly formulated as an extension of ASR. To this end, different levels are used to model the successive mappings from the acoustic signal into phonemes, words and syntactically correct sentences. Typically, all these levels are integrated into a global (HMM) model and a search is carried out for the best syntactically correct word sequence given the acoustic signal. In speech-input MT an additional mapping is required, from source-language word sequences into sentences of a different language. This mapping, however, entails an important difference with respect to the traditional mappings involved in ASR. While acoustic-phonetic and phonetic-lexical mappings are pure sequential or monotonic mappings, the mapping underlying sequences of words from a source-language to a target-language can seldom be considered monotonic.

This kind of non-monotonic mapping can be properly modeled with the concept of alignment. During the last two decades, adequate statistical alignment models (SAM) and the corresponding training algorithms have been introduced and explicitly developed for their application in MT (Brown et al., 1990; Brown et al., 1993; Al-Onaizan et al., 1999; Ney et al., 2000). On the other hand, many language pairs of interest can be adequately modeled by Stochastic Finite-State Transducers (SFST). Apart from their great simplicity, an important advantage of these models is that they can be easily integrated with the conventional (HMM – also finite-state) ASR models, thereby allowing the use of the simple, traditional Viterbi-Beam-Search techniques for very efficient translation of speech input (Jiménez et al., 1995; Vidal, 1997; Amengual et al., 2000). SFST can be also used to approach statistical alignment models (Knight and Al-Onaizan, 1998). All of these models, SAM and SFST, can be learnt automatically from bilingual corpora (Brown et al., 1993; Mäkinen, 1999; Ney et al., 2000; Bangalore and Riccardi, 2000b; Casacuberta, 2000; Vilar, 2000).

Different types of SFSTs have been applied with success for speech-input MT and ASR. Bangalore and Riccardi (2000a, 2001a,b) divided the speech-input MT problem into two subproblems: lexical selection and lexical reordering. They used SFST to solve both subproblems. In Alshawi et al. (2000a,b), the authors used head transducer models for speech-input MT. This type of models is a generalization of the finite-state transducer. In Mohri (1997), Mohri et al. (2002), a generalization of SFSTs, weighted finite-state transducers, permits a uniform representation for all knowledge levels in ASR and speech-input MT systems.

The various approaches presented in this paper follow these general ideas. Some of them focus on improving statistical alignments and the corresponding search techniques. Other, based on
finite-state technology and statistical word alignments, aim at increasing the simplicity and efficiency of fully integrated models for speech-input MT. All these developments were carried out within the framework of the European project EUTRANS (EuTrans, 2000). As an outcome of this project a number of speech-input MT demonstration prototypes were produced.

These prototypes were assessed through systematic experiments with three limited-domain translation tasks of increasing difficulty. An overall conclusion was that, for simple tasks and using a (relatively) large amount of training data, most techniques yield very good results; in particular the integrated models obtained using FS transducers. As training data shrinks, results degrade gradually; however, in this case, the advantage of integration becomes less apparent. Finally, for more complex tasks with (relatively) small training data sets, integration does not appear to help improving accuracy or robustness.

The remaining of this paper is organized as follows: Section 1 is devoted to the general statistical framework to speech translation. In Sections 2 and 3, statistical alignment and finite-state models are introduced. Two different types of statistical speech translation systems are presented in Section 5. Experiments and the corresponding results are described in Section 6. Finally, a discussion about the results achieved and some conclusions are presented in Section 7.

2. Statistical framework to speech translation

Let $x^T_1$ be the acoustic representation of an input sentence. The translation of $x^T_1$ into another language can be formulated as the problem of searching for a sequence of words $\hat{t}^I_1$ in the target language that maximizes 1

$$\hat{t}^I_1 = \arg\max_{t^I_1} Pr(t^I_1|x^T_1).$$  \hspace{1cm} (1)

The maximization is carried out over all possible sequences, $t^I_1$, of all possible lengths, $I$. For simplicity purposes, we do not present explicitly the maximization over $I$.

Speech translation can be seen as a two steps process

$$x^T_1 \rightarrow s'_I \rightarrow t'_I,$$ \hspace{1cm} (2)

where $s'_I$ is a possible decoding of $x^T_1$ in the source language that can be translated into a sequence of words, $t'_I$, in the target language. Consequently

$$\arg\max_{t'_I} Pr(t'_I|x^T_1) = \arg\max_{t'_I} \sum_{s'_I} Pr(t'_I, s'_I|x^T_1),$$ \hspace{1cm} (3)

We will assume that $Pr(x^T_1|s'_I, t'_I)$ does not depend on $t'_I$. Note that this assumption does not always hold, but, in practice, the assumption is reasonable.

---

1 By $y^j_i$ we will denote a sequence $y_i, y_{i+1}, \ldots, y_j$ of discrete symbols (words) or vectors.
Hidden Markov models (HMM) are the most adequate ones for modeling \( \Pr(x_{1}^{T}|s_{1}^{T}) \) (Knill and Young, 1997). There are different approaches to model \( \Pr(s_{1}^{T}, t_{1}^{T}) \). Stochastic finite-state transducers can directly approximate this probability distribution (see Section 4). On the other hand, the above joint probabilistic distribution can be decomposed as

\[
\Pr(s_{1}^{T}, t_{1}^{T}) = \Pr(s_{1}^{T}|t_{1}^{T}) \cdot \Pr(t_{1}^{T}).
\]

(5)

In this case \( \Pr(s_{1}^{T}|t_{1}^{T}) \) can be modeled with statistical alignments models (see Section 3) and \( \Pr(t_{1}^{T}) \) with N-grams (Ney et al., 1997) or stochastic regular (or context-free) grammars (Vidal et al., 1995).

3. Statistical alignment models and translation

A key issue in modeling the string translation probability \( \Pr(s_{1}^{T}|t_{1}^{T}) \) in Eq. (5) is the question of how we define the correspondence between the words of the target sentence and the words of the source sentence. In typical cases, we can assume a sort of pairwise dependence by considering all word pairs \((s_{j}, t_{i})\) for a given sentence pair \((s_{1}^{T}, t_{1}^{T})\). Here, we will further constrain this model by assigning each source word to exactly one target word. Later, this requirement will be relaxed. Models describing these types of dependencies are referred to as alignment models (Brown et al., 1993; Vogel et al., 1996).

When aligning the words in parallel texts, we typically observe a strong localization effect. Fig. 1 illustrates this effect for the language pair Spanish–English. In many cases, although not always, there is an additional property: over large portions of the source string, the alignment is monotonic.

Fig. 1. Word-to-word alignment.
3.1. Basic models

To arrive at a quantitative specification, we define an alignment mapping: \( j \rightarrow i = a_j \), which assigns a word \( s_j \) in position \( j \) to a word \( t_i \) in position \( i = a_j \). We rewrite the probability for the translation model by introducing the ‘hidden’ alignments \( a_j' := a_1 \cdots a_j \cdots a_J \) for each sentence pair \( (s'_1, t'_1) \). To structure this probability distribution, we factorize it over the positions in the source sentence and limit the alignment dependencies to a first-order dependence:

\[
Pr(s'_1|t'_1) = p(J|I) \cdot \sum_{a_1'} \prod_{j=1}^{J} [p(a_j|a_{j-1}, I, J) \cdot p(s_j|t_{a_j})].
\]

(6)

Here, we have the following probability distributions:

- the sentence length probability: \( p(J|I) \), which is included here for completeness, but can be omitted without loss of performance;
- the lexicon probability: \( p(s|t) \);
- the alignment probability: \( p(a_j|a_{j-1}, I, J) \).

By making the alignment probability \( p(a_j|a_{j-1}, I, J) \) dependent on the jump width \( a_j = a_{j-1} \) instead of the absolute positions \( a_j \), we obtain the so-called homogeneous hidden Markov model, for short HMM (Vogel et al., 1996).

The optimal word alignment is given by following maximization:

\[
\hat{a}'_j = \underset{a_j'}{\text{argmax}} \prod_{j=1}^{J} [p(a_j|a_{j-1}, I, J) \cdot p(s_j|t_{a_j})].
\]

(7)

The alignment \( \hat{a}'_j \) is called Viterbi alignment of this model. This equation allows us to produce a word alignment for each given bilingual sentence pair. These alignments have proven quite helpful in the learning process of other translation models (see Vilar, 2000; Casacuberta, 2000).

We can also use a zero-order model \( p(a_j|I, J) \), where there is only a dependence on the absolute position index \( j \) of the source string (Brown et al., 1993). Assuming a uniform alignment probability \( p(a_j|I, J) = 1/I \), we arrive at one of the most simple models (Brown et al., 1993).

These models can be extended to allow for source words having no counterpart in the translation. Formally, this is incorporated into the alignment models by adding a so-called ‘empty word’ at position \( i = 0 \) to the target sentence and aligning all source words without a direct translation to this empty word.

In Brown et al. (1993), more refined alignment models are introduced by using the concept of fertility. The idea is that often a word in the target language may be aligned to several words in the source language. Using, in addition, first-order alignment probabilities along the positions of the source string leads us to more complex models. Although these models take one-to-many alignments explicitly into account, the lexicon probabilities \( p(s|t) \) are still based on single words in each of the two languages. In systematic experiments, it was found that the quality of the alignments determined from the bilingual training corpus has a direct effect on the translation quality (Och and Ney, 2000a).

In this paper, we use the so-called Model 4 from Brown et al. (1993), which is referred to later on as single-word based (SWB) search. In Model 4 the statistical alignment model is decomposed into five sub-models:
• the lexicon model \(p(s|t)\) for the probability that the source word \(s\) is a translation of the target word \(t\),
• the distortion model \(p_{\|j}^- (j - j'|C(s_j), T)\) for the probability that the translations of two consecutive target words have the position difference \(j - j'\) where \(C(s_j)\) is the word class of \(s_j\) and \(T\) is the word class of the first of the two consecutive target words,
• the distortion model \(p_{\|=1}^- (j - j'|C(s_j))\) for the probability that the words aligned to one target word have the position difference \(j - j'\),
• the fertility model \(p(\phi|t)\) for the probability that a target language word \(e\) is aligned to \(\phi\) source language words,
• the empty word fertility model \(p(\phi_0|t_0)\) for the probability that exactly \(\phi_0\) words remain unaligned to.

The final probability \(p(s_1^*, a_1^*|t_1^*)\) for Model 4 is obtained by multiplying the probabilities of the sub-models for all words.

We use Model 4 in this paper for two reasons. First, it has been shown that Model 4 produces a very good alignment quality in comparison to various other alignment models (Och and Ney, 2000b). Second, the dependencies in the distortion model along the positions \(j\) of the target language words make it quite easy to integrate standard \(n\)-gram language models in the search process. This would be more difficult in the HMM alignment model (Vogel et al., 1996). Yet, many of the results presented in the following are also applicable to other alignment models.

### 3.2. Search

The task of the search algorithm is to generate the most likely target sentence \(t_1^*\) of unknown length \(I\) for an observed source sentence \(s_1^*\). The search must make use of all three knowledge sources as illustrated by Fig. 2: the alignment model, the lexicon model and the language model. All three of them must contribute in the final decision about the words in the target language.

Unfortunately, the search problem in statistical machine translation is \(\text{NP-complete}\) (Knight, 1999). Therefore, suitable approximations in search need to be performed. An important point is to find a good trade off between translation quality and efficiency. To this end, several search algorithms have been proposed, such as A* algorithm (Och et al., 2001), stack decoding (Berger et al., 1996), greedy (Berger et al., 1994; Germann et al., 2001), or beam-search algorithms (Tillmann and Ney, 2000; García-Varea et al., 1998). We use a beam-search algorithm as it offers very good possibilities to adjust the optimal tradeoff between quality and efficiency.

To illustrate the specific details of the search problem, we slightly change the definitions of the alignments:
• we use inverted alignments, which define a mapping from target to source positions rather the other way round.
• we allow several positions in the source language to be covered, i.e., we consider mappings \(B\) of the form:
  \[B : i \rightarrow B_i \subset \{1, \ldots, j, \ldots, J\}\].

This inverted direction for the alignments has the advantage that word sequence hypotheses for the target language can be conveniently constructed: starting with the first position \(i = 1\), we will build up hypotheses by extending the hypotheses for positions \(1, \ldots, i - 1\) to positions \(1, \ldots, i\).
The original direction of the alignments, i.e., from source positions to target positions, had been introduced to allow efficient training algorithms such as the EM algorithm. This direction, however, is not well suited for the search procedure generating the target sentence.

We replace the sum over all alignments by the best alignment, which is referred to as maximum approximation in speech recognition. Using a trigram language model \( p(t_i | t_{i-2}, t_{i-1}) \), we obtain the following search criterion:

\[
\max_{B'_i,t'_i} \prod_{i=1}^{j} \left[ p(t_i | t_{i-2}, t_{i-1}) \cdot p(B_i | B_{i-1}, J) \cdot \prod_{j \in B_i} p(s_j | t_i) \right]
\]

with \( p(t_i | t_{i-2}, t_{i-1}) \) being the trigram language model probabilities and \( p(B_i | B_{i-1}, J) \) being the distortion probabilities. Considering this criterion, we can see that we can build up hypotheses of partial target sentences in a bottom-to-top strategy over the positions \( i \) of the target sentence \( t'_i \) as illustrated in Fig. 3. An important constraint for the alignment is that all positions of the source sentence should be covered exactly once. This constraint is similar to that of the traveling salesman problem where each city has to be visited exactly once. Details on various search strategies can be found in Tillmann et al. (1997), Nießen et al. (1998) and Ney et al. (2000).

In order to take long context dependencies into account, we use a class-based five-gram language model with backing-off. Beam-search is used to handle the huge search space. To normalize the costs of partial hypotheses covering different parts of the input sentence, an (optimistic) estimation of the remaining cost is added to the current accumulated cost as follows. For each word
in the source sentence, a lower bound on its translation cost is determined beforehand. Using this lower bound, it is possible to achieve an efficient estimation of the remaining cost (Tillmann and Ney, 2000).

3.3. Alignment template approach (ALTEMP)

A general shortcoming of the baseline alignment models is that they are mainly designed to model the lexicon dependencies between single words. Therefore, we have extended the approach to handle word groups or phrases rather than single words as the basis for the alignment models (Och et al., 1999). In other words, a whole group of adjacent words in the source sentence may be aligned to a whole group of adjacent words in the target language. As a result, the context of words tends to be explicitly taken into account, and the differences in local word orders between source and target languages can be learned explicitly. Fig. 4 shows some of the extracted alignment templates for the language pair Spanish–English. The training algorithm for the alignment templates extracts all phrase pairs which are aligned in the training corpus up to a maximum length of seven words. To improve the generalization capability of the alignment templates, the templates are determined for word classes rather than words directly. These word classes are determined by an automatic clustering procedure (Och, 1999). For this model we use a similar beam-search algorithm as presented for the single-word based alignment models.

4. Finite-state translation models

Stochastic finite-state transducers allow to model $Pr(s'_i, t'_j)$ in Eq. (4). Finite-state models are interesting for their simplicity and potentially high parsing efficiency.

4.1. Stochastic finite-state transducers

A stochastic finite-state transducer (SFST) is a finite-state network whose transitions are labeled by three items:
(a) an input symbol (a word from the input vocabulary);
(b) an output string (a sequence of words from the output vocabulary) and
(c) a transition probability.

Fig. 5 shows a small fragment of a SFST for Italian to English translation.

More formally, a SFST (or a stochastic regular syntax-directed translation scheme), \( \mathcal{T} \), is a tuple \( (Q, \Sigma, \Delta, R, q_0, F, P) \), where

(a) \( Q \) is a finite set of or states;
(b) \( q_0 \) is the initial state;
(c) \( \Sigma \) is a finite set of input symbols;
(d) \( \Delta \) is a finite set of output symbols (\( \Sigma \cap \Delta = \emptyset \));
(e) \( R \) is a set of rules or transitions of the form \( (q, a, \omega, q') \) for \( q, q' \in Q \), \( a \in \Sigma \), \( \omega \in \Delta^* \) and
(f) \( P : R \to \mathbb{R}^+ \) (transition probabilities) and \( F : Q \to \mathbb{R}^+ \) (final-state probabilities) are functions such that:

\[
\forall q \in Q : F(q) + \sum_{(q,a,\omega,q') \in R} P(q,a,\omega,q') = 1. \tag{9}
\]

A particular case of SFST is the subsequential transducer (SST). This is a finite-state transducer with the basic restriction of being deterministic. This implies that if both \( (p,a,\omega,q) \) and \( (p,a,\omega',q') \) belong to the set of rules, \( R \), then \( \omega = \omega' \) and \( q = q' \). In addition, SSTs can emit a target substring when the end of the input string has been detected.

A translation form, \( \phi \) is a sequence of transitions in a SFST \( \mathcal{T} \):

\[
\phi : (q_0, s_1, \tilde{t}_1, q_1), (q_1, s_2, \tilde{t}_2, q_2), \ldots, (q_{l-1}, s_l, \tilde{t}_l, q_l). \tag{10}
\]

\( \Delta^* \) and \( \Sigma^* \) we denote the set of finite-length strings on \( \Delta \) and \( \Sigma \), respectively.
where $\tilde{t}_j$ denotes a substring of output symbols, such that $\tilde{t}_1 \tilde{t}_2 \cdots \tilde{t}_f = t'_1$ and $q_f \in F$. A pair $(s'_1, t'_1) \in \Sigma^* \times \Delta^*$ is a translation pair if there is a translation form $\phi$ in the SFST $\mathcal{T}$.

The probability of a translation form $\phi$ is defined as:

$$Pr_{\mathcal{T}}(\phi) = P(q_0, s_1, \tilde{t}_1, q_1) \cdot P(q_1, s_2, \tilde{t}_2, q_2) \cdots P(q_{f-1}, s_f, \tilde{t}_f, q_f) \cdot F(q_f).$$  \hspace{1cm} (11)

Finally, the probability of a translation pair is

$$Pr_{\mathcal{T}}(s'_1, t'_1) = \sum_{\phi \in d(s'_1, t'_1)} Pr_{\mathcal{T}}(\phi).$$  \hspace{1cm} (12)

where $d(s'_1, t'_1)$ is the set of all translation forms for the pair $(s'_1, t'_1)$. These models have implicit source and target language models in their own definition. The source language model, $\mathcal{S}_{I}$, can be defined from Eq. (12) as

$$Pr_{\mathcal{S}_{I}}(s'_1) = \sum_{t'_1} Pr_{\mathcal{T}}(s'_1, t'_1).$$  \hspace{1cm} (13)

In practice, this source language model can be obtained by removing the output symbols in each rule/transition. From Fig. 5, the source language model is presented in Fig. 6.

On the other hand, the target language model, $\mathcal{S}_{O}$, can be defined from Eq. (12) as

$$Pr_{\mathcal{S}_{O}}(t'_1) = \sum_{s'_1} Pr_{\mathcal{T}}(s'_1, t'_1).$$  \hspace{1cm} (14)

Similarly to the source language model, this target language model can be obtained by removing the input symbols in each rule/transition. From Fig. 5, the target language model is presented in Fig. 7.

4.2. Search

Similarly to Section 3.2, the task of the search algorithm is to generate the most likely target sentence $t'_1$ of unknown length $f$ for an observed source sentence $s'_1$, but in this case, according to a SFST as a translation model. Formally,
This is a difficult computational problem (Casacuberta and de la Higuera, 2000), but it admits a computationally efficient approximation by substituting the sum operator in Eq. (15) by a maximization:

$$\arg\max_{r_t^1} \Pr_T(s_{J1}, t_{I1}) = \arg\max_{r_t^1} \sum_{d(s_{I1}, t_{I1})} \Pr_T(d(s_{J1}, t_{I1})). \quad (15)$$

The maximizations in Eq. (16) can be performed by using an adaptation of the well known Viterbi algorithm (Picó and Casacuberta, 2001), which yields an optimal sequence of states for the observed source sentence $s_{J1}$. From this optimal sequence of states, an approximation to the most likely target sentence can be obtained by concatenating the output strings of the transitions involved in this sequence.

## 4.3. Learning stochastic finite-state transducers

A SFSTs has two basic components: the structural component (states and transitions) and the probabilistic component (probabilities associated to the transitions). These components can be learned from training pairs in a single process or in a two steps-process. In the latter case, first the structural component is learned and next the probabilistic components are estimated from training samples. The GIATI $^3$ is a technique of the first type and OMEGA $^4$ is a technique for learning the structural component of a SFST.

The GIATI technique is based on building a training string for each training pair. Then a (stochastic) regular (e.g., a $n$-gram) grammar is inferred from the set of training strings. Finally, the inferred regular grammar is transformed into a stochastic finite-state transducer. The details of the GIATI technique were presented in (Casacuberta, 2000).

The OMEGA algorithm (Vilar, 1998) can be seen as an improvement over OSTIA (Oncina et al., 1993), hence the name (OSTIA Modified for Employing Guarantees and Alignments). The

---

$^3$ Grammatical Inference and Alignments for Transducer Inference previously known as Morphic Generator Translator Inference (MGTI).

$^4$ Onward subsequential transducer inference algorithm Modified for Employing Guarantees and Alignments.
algorithms of this kind lead to *identification in the limit* of subsequential transducers from sequences of training pairs. The details of the OMEGA algorithm were presented in Vilar (2000).

To estimate the probabilistic component in the two-step approaches, a *maximum likelihood* technique or other possible criteria can be used (Picó and Casacuberta, 2001). One of the main problems associated with the learning process is the modeling of events not seen in the training set. This problem can be tackled, in a similar way as in language modeling, by using smoothing techniques; either in the estimation of the probabilistic components of the SFST (Llorens, 2000) or within of the process of learning both components (Casacuberta, 2000). Other (similar) approaches to learn finite-state transducers can be found in Bangalore and Riccardi (2000b) and Mäkinen (1999).

5. Search for speech translation

In Sections 3.2 and 4.2, the task of the search algorithm was to compute a most likely target sentence $t_1'$ of unknown length $I$ for the observed *text* source sentence $s_1'$. For speech translation, the problem is different in that the task now is to obtain a most likely target sentence $t_1'$ of unknown length $I$ for the observed *acoustic* source sentence $x_{1T}$. From Eq. (4),

$$\text{argmax}_t \frac{Pr(t_1|x_{1T})}{Pr(s_{1T}|t_1)} = \text{argmax}_t \sum_{s_{1T}} Pr(s_{1T}, t_1') \cdot Pr(x_{1T}|s_{1T})$$

By approximating the sum by the max-operator, a source sentence can be associated to the acoustic signal whose translation is the target sentence searched for:

$$\text{argmax}_t \frac{Pr(t_1'|x_{1T})}{Pr(s_{1T}|t_1')} \approx \text{argmax}_{s_{1T}} \text{max}_{t_1'} Pr(s_{1T}, t_1') \cdot Pr(x_{1T}|s_{1T})$$

(18)

Different approaches can be proposed for searching in Eq. (18). In this section we will describe the “integrated” architecture that is very appropriate if $Pr(s_{1T}, t_1')$ is modeled by means of adequate SFSTs. In this case, the acoustic models are integrated into the translation model and the search can explicitly maximize the product $Pr(s_{1T}, t_1') \cdot Pr(x_{1T}|s_{1T})$.

Unfortunately, the interesting statistical models discussed in Section 3 do not allow such an easy integration (Ney, 1999). On the other hand, in any case, the search for the optimal target sentence using the integrated architecture may require a high computational effort. A reduction of this effort can be achieved by a further approximation consisting in decomposing the search into a “two-step” or “serial” architecture. In this case, a conventional source speech decoding is followed by a text-input translation of the decoded sentence into the target sentence.

5.1. Integrated architecture to SFST speech translation

The maximization of Eq. (18) can be performed by searching for the optimal target sentence in an *integrated network*, similar those used for speech recognition. Each transition of the SFST is expanded into the concatenation of HMMs of the phone units that define the input word of the transition (Fig. 8).
As in the case of standard speech recognition, the general search process for translation is a difficult computational problem (Casacuberta and de la Higuera, 2000). Nevertheless, quite adequate approximations can be obtained by using the Viterbi algorithm on a trellis associated to the input acoustic sequence and the integrated network. This algorithm yields an “optimal” state sequence which is in fact an approximation to the join maximization of $s_{1}^{J}$ and $t_{1}^{I}$ in Eq. (18).

5.2. Serial architecture to speech translation

Using $\Pr(t_{1}^{I}, s_{1}^{J}) = \Pr(t_{1}^{I}|s_{1}^{J}) \cdot \Pr(s_{1}^{J})$ in Eq. (18), the maximization becomes

$$\arg\max_{t_{1}^{I}} \Pr(t_{1}^{I}|x_{1}^{T}) \approx \arg\max_{t_{1}^{I}} \max_{s_{1}^{J}} \left\{ \Pr(t_{1}^{I}|s_{1}^{J}) \cdot \Pr(x_{1}^{T}|s_{1}^{J}) \right\}. \quad (19)$$

The search for an optimal target sentence in Eq. (19) can be approximated as follows:

1. **Word decode** $x_{1}^{T}$ by searching for a sequence of words $\hat{s}_{1}^{J}$ such that

$$\hat{s}_{1}^{J} \approx \arg\max_{s_{1}^{J}} \left\{ \Pr(s_{1}^{J}) \cdot \Pr(x_{1}^{T}|s_{1}^{J}) \right\}, \quad (20)$$

where $\Pr(x_{1}^{T}|s_{1}^{J})$ is modeled by acoustic models and $\Pr(s_{1}^{J})$ by a source language model.

2. **Given** $\hat{s}_{1}^{J}$, **translate** $\hat{s}_{1}^{J}$ by searching for a sequence of words $\hat{t}_{1}^{I}$ such that

$$\hat{t}_{1}^{I} \approx \arg\max_{t_{1}^{I}} \Pr(t_{1}^{I}|\hat{s}_{1}^{J}) = \arg\max_{t_{1}^{I}} \Pr(\hat{s}_{1}^{J}, t_{1}^{I}), \quad (21)$$

where $\Pr(\hat{s}_{1}^{J}, t_{1}^{I})$ is modeled by a SFST. Alternatively,

$$\hat{t}_{1}^{I} \approx \arg\max_{t_{1}^{I}} \Pr(t_{1}^{I}|\hat{s}_{1}^{J}) = \arg\max_{t_{1}^{I}} \Pr(\hat{s}_{1}^{J}|t_{1}^{I}) \cdot \Pr(t_{1}^{I}), \quad (22)$$

---

**Fig. 8.** Example of the integration process of the lexical knowledge (b) and the phonetic knowledge (c) in a FST (a). $\lambda$ denotes the empty string.
where $Pr(s^t_1|t^s_1)$ can be modeled by statistical alignment models and $Pr(t^s_1)$ by a target language model (Brown et al., 1993; Dagan et al., 1993; Vogel et al., 1996).

6. The EuTRANS systems

The EuTRANS project was aimed at developing machine translation systems to assist human to human (speech) communications in specific domains (EuTrans, 2000). As a result, telephone speech input translation systems capable to translate limited-domain telephone calls from one language into another have been developed.

These systems are based on the ATROS engine, a continuous-speech recognition/translation system which uses stochastic finite-state models at all its levels: acoustic-phonetic, lexical and syntactic/translation (Llorens et al., 1999; Casacuberta et al., 2001).

ATROS supports both types of architectures described in Section 5: integrated and serial. In the first case an integrated finite-state network is used to translate an input utterance. In the second architecture the translation is performed in two steps (see Section 2). The first step only needs a finite-state language model (e.g., a $n$-gram), rather than a FST, and ATROS produces an optimal source-language decoding of the input speech signal. In the second step, a FST (Section 4) or a statistical translation system (Section 3) can be used to translate the decoded source-language sentence into the target translation.

6.1. Speech-input translation prototypes

Three types of speech-to-speech translation prototypes have been implemented for Spanish to English and for Italian to English. In any case, the general application was the translation of queries, requests and complains made by telephone (or microphone) to the front desk of a hotel. Three tasks of different degree of difficulty have been considered. In the first one (EuTRANS-0), Spanish-to-English translation systems were learned from a big and well controlled training corpus (about 170k different pairs). In the second one (EuTRANS-I), also from Spanish to English (EuTRANS-I), the systems were learned from a random subset of 10k pairs from the previous corpus, which was established as a more realistic training corpus for the kind of application considered. In the third and last one, from Italian to English (EuTRANS-II), the systems were learned from a small training corpus (about 3k pairs) that was obtained from a transcription of a spontaneous speech corpus.

Two translation models based on (pure) statistical methods were tested in these prototypes: SWB and ALTEMP. In these systems only the serial architecture was adopted, using trigrams for the front-end speech decoding.

In the systems based on FSTs both architectures, integrated and serial, were considered, using the same translation models in both cases. These models were learnt by the OMEGA and GIATI transducer inference algorithms. In the serial architecture trigrams were also used for the front-end speech decoding.
For demonstration purposes, the output English speech is obtained by using an open source Text-To-Speech synthesizer (Festival system) which offers understandable speech and reasonably good quality (Taylor et al., 1998).

6.1.1. **EuTRANS-0: Spanish–English translation based on a large training corpus**

These prototypes were developed to show the feasibility of the finite-state technology to implement speech-input translation systems for limited domains.

A text training corpus was generated in a semi-automatic way using travel booklets as a seed corpus (Amengual et al., 2000). This resulted in a very large set of 490,000 pairs, including many repetitions of simple sentences; in fact, only approximately 170k pairs were different. Such a large corpus was adopted in this first prototype in order to ensure that performance was not impaired because of lack of training data. The (test-set) perplexity (bigrams) for the source language was 6.8 and 5.6 for the target language. The size of the Spanish/English vocabularies were 686/513.

From a selected subset of these text data, a multi-speaker Spanish speech corpus was produced. The utterances were acquired using a microphone. More specifically, speech signals were limited to a 7.8 kHz bandwidth and sampled at 16 kHz with 16 bits linear resolution. Another selected subset of the text data was used to produce telephone speech signals. In this case, speech signals were limited to 3.8 kHz and sampled at 8 kHz, 10 bits per sample.

The acoustic models of phone units were left-to-right continuous-density HMMs. 26 Spanish monophones were adopted and the corresponding models were trained with the HTK Toolkit (Young et al., 1997). The speech corpus was composed of 11,000 running words. This amounted to approximately 3.8 h of speech.

The microphone-input speech test set consisted of 84 sentences not included in the training corpus. These sentences were uttered several times by four speakers (different from those used for acoustic training), yielding 336 Spanish utterances (3000 running words and 0.5 h of speech). The telephone-input test-set was similar except that the number of test speakers were 10 rather than 4.

A summary of the main features of the EuTRANS-0 corpus is presented in Table 1. The systems trained with this corpus were used to build a prototype that is fully operational for both microphone and telephone input.

6.1.2. **EuTRANS-I: Spanish–English translation**

Clearly, 170k different training pairs is an unrealistically large training corpus for a simple limited-domain task as the one here considered. In order to approach more realistic conditions, a

<table>
<thead>
<tr>
<th>Table 1</th>
<th>The EuTRANS-0 corpus</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>Spanish</td>
</tr>
<tr>
<td>Training text</td>
<td>Sentence pairs</td>
</tr>
<tr>
<td></td>
<td>Different sentence pairs</td>
</tr>
<tr>
<td></td>
<td>Running words</td>
</tr>
<tr>
<td></td>
<td>Vocabulary</td>
</tr>
<tr>
<td></td>
<td>Bigram test-set perplexity</td>
</tr>
<tr>
<td>Training speech test</td>
<td>Running words</td>
</tr>
<tr>
<td></td>
<td>Speech utterances</td>
</tr>
<tr>
<td></td>
<td>Running words</td>
</tr>
</tbody>
</table>
subset of EuTRANS-0 was randomly selected, yielding a reduced text training corpus of 10,000 paired sentences.

The acoustic models and speech test data were identical as those of EuTRANS-0, both for the microphone and telephone input systems.

A summary of the main features of the EuTRANS-I corpus is presented in Table 2. The prototype based on the systems trained with this corpus is also fully operational for both telephone and microphone input.

6.1.3. EuTRANS-II: Italian–English translation

The Italian–English task was significantly more complex and closer to a real situation than the Spanish–English ones. In this case, the speech corpus consisted of acquisitions of real phone calls to the front desk of a hotel, simulated using Wizard of Oz techniques (Aiello et al., 1999). This corpus is highly spontaneous and contains many non-speech artifacts. The text corpus was obtained by manually transcribing the acquired Italian utterances and translating them into corresponding English sentences. From this text corpus, 3,038 pairs of sentences were used for training the translation model.

The utterances from which this text corpus was obtained were used to train the (Italian) phone models. This amounted to approximately 7.9 h of speech, uttered by 276 speakers. The acoustic models of phone units were also left-to-right continuous density HMMs. Since the amount of available training data was very limited and the task was more spontaneous than the previous ones, more powerful techniques had to be used to train the models of the phone units. In particular, linear discriminant analysis (LDA) was used to improve the feature representation of the speech signal. On the other hand, decision-tree clustered generalized triphones (CART with 1,500 tied states plus silence) were adopted as phone-units. The models were trained using a Viterbi approximation (Ney et al., 1998).

The speech test set consisted of 278 different Italian sentences, corresponding to the utterances of 24 speakers (0.8 h of speech). Both the test sentences and the speakers were different from those involved in the training corpus.

A summary of the main features of the EuTRANS-II corpus is presented in Table 3. The prototype based on the systems trained with this corpus is operational through standard telephone lines for remote (or local) operation.

Table 2

The EuTRANS-I corpus

<table>
<thead>
<tr>
<th>Data</th>
<th>Spanish</th>
<th>English</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training text</td>
<td>Sentence pairs</td>
<td>10,000</td>
</tr>
<tr>
<td></td>
<td>Different sentence pairs</td>
<td>132,198</td>
</tr>
<tr>
<td></td>
<td>Running words</td>
<td>134,922</td>
</tr>
<tr>
<td></td>
<td>Vocabulary</td>
<td>513</td>
</tr>
<tr>
<td></td>
<td>Bigram test-set perplexity</td>
<td>6.3</td>
</tr>
<tr>
<td>Training speech test</td>
<td>Running words</td>
<td>11,000</td>
</tr>
<tr>
<td></td>
<td>Speech utterances</td>
<td>336</td>
</tr>
<tr>
<td></td>
<td>Running words</td>
<td>3,000</td>
</tr>
</tbody>
</table>
6.2. Prototype assessment

To assess the performance of the systems, several error criteria were used. On the one hand, the (Recognition) Word Error Rate (WER) was adopted, as usual, to assess speech input decoding results. This performance criterion, widely used in speech recognition, is basically the minimum number of substitution, insertion and deletion operations that have to be performed to convert the word string produced by a system into a given reference word string.

On the other hand, the Translation Word Error Rate (TWER) was adopted as a rather crude measure of the accuracy of translation results. TWER is just the WER obtained by comparing each automatically translated sentence with a single reference target sentence. Because most source-language sentences allow for many correct target translations, TWER is clearly a pessimistic error estimation. This is particularly true in the case of the Italian–English task, due to the free-form, human-produced test set reference translations. The Translation Position-independent word Error Rate (TPER) is similar to TWER, but it compares the words in the two sentences without taking the word order into account.

Finally, a Subjective Sentence Error Rate (SSER) has been used in some experiments. To this end, the translations were scored by classification into a number of error classes, ranging from “perfect” to “absolutely wrong”. The SSER is the mean score on a test set (Nießen et al., 2000). In comparison to the TWER, this criterion is more reliable and conveys more information. However, measuring the SSER is expensive, since it is not computed automatically but as result of laborious evaluation by human experts.

In each table of results, there are different columns:
- The type of speech input (microphone -Mic- or telephone -Tel-).
- The translation models (ALTEMP, SWB, GIATI or OMEGA).
- The architecture (serial or integrated).
- The source language model (trigrams or the implicit source language models of GIATI or OMEGA).
- The WER of the (implicit -integrated- or explicit -serial-) speech decoding.
- The TWER of the translations.
- The TPER of the translations.
- The SSER of the translations (only for some experiments).

It is worth noting that a specific column appear in the tables for the type of source language model adopted. For the serial experiments, source language models were always standard
trigrams. But for the integrated experiments the source language model is in fact embedded in the FST and the speech decoding is a subproduct of the translation process itself. Consequently, in these cases, the WER is associated to the intrinsic input language model of the transducer (see Section 4.1).

The Italian–English EuTRANS-II prototype, running on a standard Intel PC Pentium III 450 MHz machine, achieves quite acceptable response time (about three times real time or less), while the Spanish–English EuTRANS-0 and EuTRANS-I prototypes often run in less than real time, even on low-cost Pentium machines. Assessment results of the EuTRANS-0 and EuTRANS-I prototypes are presented in Tables 4 and 5, respectively. Assessment results of the EuTRANS-II prototype are presented in Table 6.

The results with EuTRANS-0 (easy task, large training corpus) show that the finite-state approach OMEGA in the integrated architecture was the best approach. The next successful approach was ALTEMP that is based on the statistical alignments. An interesting result, clearly observed with OMEGA, is that in the integrated architecture the TWER can be lower than the WER implicitly produced by the transducer. This is consistent with the relative source-target perplexities of the sentences in this task (see Table 1). By comparing the integrated and serial architectures, using OMEGA, we can see that the WER is higher for the integrated approach than for the serial one. This means that the implicit input language model of OMEGA is in fact worse than the corresponding trigram, which is probably due to the lack of an appropriate smoothing for the OMEGA models. Nevertheless, the overall translation model is yet better for the integrated architecture.

The results with EuTRANS-I (easy task, medium-size training corpus) show that ALTEMP and GIATI (integrated architecture) run better than the other approaches for microphone input and for telephone input, respectively. The behavior of lower TWER than WER is also observed here for OMEGA in the integrated architecture. With the exception of GIATI with telephone-input, in this task the results with integrated and serial architectures were very similar.

<table>
<thead>
<tr>
<th>Input</th>
<th>Models</th>
<th>Architecture</th>
<th>Source LM</th>
<th>WER</th>
<th>TWER</th>
<th>TPER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mic</td>
<td>ALTEMP</td>
<td>Serial</td>
<td>Trigrams</td>
<td>2.4</td>
<td>4.7</td>
<td>4.3</td>
</tr>
<tr>
<td></td>
<td>SWB</td>
<td>Serial</td>
<td>Trigrams</td>
<td>2.4</td>
<td>12.1</td>
<td>11.6</td>
</tr>
<tr>
<td></td>
<td>GIATI</td>
<td>Serial</td>
<td>Trigrams</td>
<td>2.4</td>
<td>6.0</td>
<td>6.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Integrated</td>
<td>GIATI</td>
<td>2.3</td>
<td>6.0</td>
<td>6.0</td>
</tr>
<tr>
<td></td>
<td>OMEGA</td>
<td>Serial</td>
<td>Trigrams</td>
<td>2.4</td>
<td>4.5</td>
<td>4.4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Integrated</td>
<td>OMEGA</td>
<td>4.1</td>
<td>3.5</td>
<td>3.5</td>
</tr>
<tr>
<td>Tel</td>
<td>ALTEMP</td>
<td>Serial</td>
<td>Trigrams</td>
<td>8.6</td>
<td>9.9</td>
<td>9.6</td>
</tr>
<tr>
<td></td>
<td>SWB</td>
<td>Serial</td>
<td>Trigrams</td>
<td>8.6</td>
<td>17.2</td>
<td>16.8</td>
</tr>
<tr>
<td></td>
<td>GIATI</td>
<td>Serial</td>
<td>Trigrams</td>
<td>8.6</td>
<td>11.6</td>
<td>11.4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Integrated</td>
<td>GIATI</td>
<td>7.5</td>
<td>10.7</td>
<td>10.5</td>
</tr>
<tr>
<td></td>
<td>OMEGA</td>
<td>Serial</td>
<td>Trigrams</td>
<td>8.6</td>
<td>9.4</td>
<td>9.3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Integrated</td>
<td>OMEGA</td>
<td>8.4</td>
<td>7.6</td>
<td>7.5</td>
</tr>
</tbody>
</table>
The results with EUTRANS-I prototype (difficult task, small training corpus) show that the best ones were achieved with ALTEMP and GIATI (with serial architecture). The integrated architecture did not help improving results in this task and the TWER was worse than the source WER in all the cases.

A few words are in order about the comparison between the two statistical translation approaches, namely the single-word based approach (SWB) and the alignment template approach (ALTEMP). As the results in all experiments show, the alignment template approach is always significantly better than the single-word based approach. The same result was also found on other translation tasks such as the Verbmobil task and the Canadian Hansards task. Our interpretation of this result is that the alignment template approach is better able to take the context into account during the translation process and this context is needed for high-quality translations.

### 7. Discussion and conclusions

From the results of the previous section it can be seen that for simple tasks and using sufficient training data, all the techniques we have tried yield very good results, including in particular the integrated FS transducers trained with OMEGA. As training data shrinks, results degrade gradually; however, in this case, the advantage of integration becomes less apparent. Finally, for
more complex tasks, with (relatively) small amount of training data, results become generally worse. Nevertheless, for the ALTEMP and GIATI techniques, and according to the subjective evaluation tests, performance levels can still be sufficient for some applications. In this last case, however, integration does not appear to be helpful to improve quality or robustness.

It is somewhat unexpected that integration does not always help to improve the overall translation accuracy. While we do not have a perfectly clear understanding of this empirical fact, we would like to discuss under which conditions an integrated approach is expected to significantly outperform a serial approach.

To better understand these conditions, Table 7 shows a summary of the results presented in the previous section, accompanied with the corresponding “Text Translation Word Error Rate” (TTWER); that is the WER of pure text-to-text translation of correct transcriptions of the test-set speech utterances. According to these results, serial-coupling performance seems to degrade rather gracefully as training/task conditions become increasingly hard. In contrast, the integrated approach shows a sharper degradation behaviour. It starts with a very good performance, significantly better than that of the serial approach, when conditions are favourable. But, after certain point, performance degrades rather quickly, making integration even worse than the serial approach.

In conclusion, the simple serial architecture proves useful given the relatively low performance of current state-of-the-art text-to-text translators (and to some degree also of current speech decoders). Nevertheless, as accuracy of the systems will improve, we think that the need of integration will become increasingly apparent in the future.

Acknowledgements

This work has been partially supported by the European Union under Grant IT-LTR-OS-30268.
References


