EUTRANS: a Speech-to-Speech Translator Prototype.

M. Pastor, A. Sanchis, F. Casacuberta, E. Vidal

Institut Tecnològic d’Informàtica
Universitat Politècnica de València
Camí de Vera s/n 46071 València
{moises,asanchis,fcn,evidal}@iti.upv.es

Abstract
EUTRANS system is a telephone speech input translation prototype capable of translating telephone calls from one language to another. It assumes a human to human communication, with each one speaking a different language, assisted by a system with translation capabilities. The prototype has been developed as a demonstrator for the European project with the same name. EUTRANS achieves a response time close to real time for speaker-independent, medium complexity tasks (a few thousand words) and offers competitive accuracy. The acoustic, language and translation models are finite-state networks that are automatically learnt from training samples, this makes the system easily adaptable to new tasks. It runs on a standard PC with audio capability and a cheap modem. The system is currently available for two translation tasks: FUB task (Italian-English) and Traveler task (Spanish-English).

1. Introduction
EUTRANS is a telephone speech input translation engine capable of translating telephone calls from one language to another. It assumes a human to human communication, with each one speaking a different language. From an ideal point of view, this process should be bi-directional. Both users would feel as if they were speaking the same language. Nevertheless, the EUTRANS project is aimed at improving Automatic Speech-to-Speech Machine Translation technology and, as a first step, we only model the translation in one direction. Therefore, in the current prototype, just for demonstration purposes, the caller will receive a synthesized translation for his uttered sentence through the standard telephone line. Additionally, the entire process can be monitored in more detail using front-ends in the host computer or using a web page, for remote visualization.

EUTRANS works on limited domains [3] where it achieves high performance. Currently, it is completely available for two tasks involving two language pairs and two translation directions for Italian-English and Spanish-English. It is possible to have as many tasks as the computer memory can hold and one can remotely change from one task to another by simply using the telephone keypad. Using a standard Pentium PC computer, the system achieves response times from 2.5 times real time factor to less than real time, depending on the task.

This system is based on the ATROS (Automatically Trainable Recognizer Of Speech) engine [8, 7, 6, 5]. ATROS is a continuous-speech recognition/translation system which uses stochastic finite-state models [2] at all its levels: acoustic-phonetic, lexical and syntactic/translation. All these models can be learnt automatically from speech and/or text data [7, 11]. This makes the system easily adaptable to different recognition/translation tasks. The use of finite-state models allows the system to obtain the translation synchronously with the recognition process [9]. The ATROS system is completely coded in the C programming language. It only needs a general-purpose CPU without the help of any digital signal processor. This lower demand for computational requirements is crucial for building low cost real-time systems and to allow for great portability and hardware independence.

2. Prototype Architecture
The EUTRANS prototype uses a Linux PC computer with a standard audio card and a voice-modem.

The system is remotely controlled using the telephone keypad: allowing us to change one task for another, to translate a new utterance, to repeat the translated output utterance, etc. When the system is in command state (after each call), if no key is pressed, after a while, it hangs up automatically. The EUTRANS system also provides an HTML page with the output of recognition/translation, just for remote monitoring. All the input calls are stored to be used in system refinements.

Most parts of EUTRANS have been programmed in C language with a few parts in TCL/TK and shell scripts. It is composed of several logic parts (see Fig. 1). These parts will be described in following sections.
2.1. Control subsystem

The Control subsystem carries out the coordination between the different modules. EUTRANS is composed of. This module also includes the Control Modem Manager, which uses the Linux command `vmcp`. It is a simple way to drive a modem through a program that allows interaction of the type “send-string, wait-string”. A command string is sent to the modem and then the program waits for a specified answer from the modem itself.

The Control subsystem works as follows: when the EUTRANS system receives an incoming call (which is continuously waiting for) the Control process plays a short greeting utterance and asks for the speech input from the caller. Then, it puts the recognition engine ATROS in recognition mode and keeps waiting for the speech signal. When the recognition/translation is performed, the Control process passes the translation string to the TTS, which synthesizes it and plays the resulting synthetic utterance into the telephone line. If recognition/translation fails, an apologetic message is produced. At this point, it puts itself into remote control mode. If after a short time period no telephone key has been pressed, it hangs the telephone up.

2.2. Recognition Module: ATROS

ATROS (Automatically Trainable Recognition Of Speech) is a continuous-speech recognition/translation system based on stochastic finite state acoustic-phonetic, lexical and syntactic/translation models. All these models can be learnt automatically from speech and/or text data [7, 10, 6]. This makes the system easily adaptable to different recognition/translation tasks.

The ATROS system has been developed in the C programming language. It only needs a general-purpose CPU, without the help of any digital signal processor. This low demand for computational requirements is crucial to building real-time systems and it allows for great portability and hardware independence.

For medium complexity tasks (roughly a thousand words) with low perplexity, speech recognition can be performed in less than real-time, achieving very competitive accuracy in speaker-independent experiments.

2.2.1. System Design

ATROS is basically composed of two different modules: the feature extraction module and the decoding module. The first module computes a sequence of feature vectors (cepstral coefficients) from the input speech signal. From this sequence, the second module computes a string of words as a hypothesis of the words that have been uttered. The feature extraction module and the decoding module are described in more detail in the following subsections.

ATROS uses thread technology. This technology allows having some threads of execution over the same code. In our case, it consists of four threads working at the same time: Control Thread, Parsing Thread, Acquisition/Preprocessing Thread and Signal Management Thread. In this way, ATROS can take advantage of the use of multiprocessors, if available, and can easily perform the module coordination. All the threads are started by the Control Thread, which waits for user control commands. The Parsing Thread corresponds to the routines involved in the decoding module; the Acquisition/Preprocessing Thread corresponds to the routines involved in the feature extraction module.

In order to achieve real-time computation, there exists a close interaction between Acquisition/Preprocessing and Parsing threads. Each new frame of cepstral coefficients which is obtained by the acoustic preprocessing is immediately supplied to the linguistic decoder. This works in a so-called frame-synchronous manner, thereby allowing computation to be performed in parallel with acquisition/preprocessing, without waiting until the end of the utterance.

2.2.2. Feature Extraction Module

Speech is acquired and sampled at 8KHz with 16 bits of precision. Samples are grouped into successive frames and then passed through a Hamming window. Each frame is 25ms. long and the interframe distance is 10ms. A filter bank formed by 18 trapezoidal filters with increasing widths according to mel-frequency scale, is applied to the 256-point FFT, producing 18 spectral weighted mean values. A discrete cosine transform is applied to these coefficients producing 11 mel-frequency cepstral coefficients. First and second derivatives of the cepstrum coefficients and energy are also added.
2.2.3. Decoding Module

The Decoding module uses different knowledge sources to model the successive mappings from the acoustic signal into phonemes, words and syntactically-correct sentences [6]. The acoustic processor is composed of two levels: the acoustic level and the lexical level. The acoustic level consists of phone-like models represented as CDHMMs. The lexical level consists of word acoustic models which are obtained by the concatenation of acoustic-phonetic models according to orthographic-phonetic rules. These models are represented by stochastic finite state networks whose transitions are labelled with phone-like units.

An integrated network was dynamically constructed for decoding. This integrated network is obtained by replacing each arc in the syntactic/translation model by the lexical model corresponding to the associated word, and then replacing each arc in the lexical models by the acoustic model corresponding to the associated acoustic model (see Fig. 2). The decoding process is performed using the beam-search Viterbi algorithm [1] through the integrated network. In order to achieve the maximum spatial efficiency, the integrated model is not fully expanded in memory, only those states which are not pruned by the beam search are expanded for each frame. In ATROS system, by following the edges in the optimal path through the syntactic/translation model, we can recover not only the optimal sequence of words in the input language but also the corresponding translation.

2.3. Graphical Front-End Interface

Two graphical interfaces have been developed, one for local control and another for remote monitorization. The local interface is a Tcl/Tk application [4] which provides a friendly user interface not only for visualization but for controlling the engine (see Fig. 3). The remote interface consists of a web page where the translated sentence and its associated input language recognized sentence are shown, as well as some information on how to use the prototype (see Fig. 4).

2.4. Text to Speech Subsystem (TTS)

A Text-To-Speech (TTS) synthesizer is a computer-based system that is able to read any text aloud. The Festival synthesizer 2 from the University of Edinburgh (UK) is used as the TTS for the EU TRANS Prototype. Currently, the Festival system is freely available for use in research projects.

In EU TRANS, the TTS subsystem gets the result of the translation process, which is a string of words in the target language. The subsystem takes their result as input and produces the corresponding speech utterance.

3. Tasks

Currently there are two fully operative tasks:

- **Traveler task**: The general framework aims at covering common sentences that might be needed

2 http://www.cstr.ed.ac.uk/projects/festival/
by a traveler visiting a hotel in a foreign country. For example asking for rooms, making complaints, asking for the bill, etc. For this purpose, a corpus of paired bilingual sentences (Spanish-English) that belong to this semantic domain has been collected and a subset of it used to train the different models. This subset consists of 10,000 pairs of sentences with a vocabulary of 683 Spanish words, 514 English words, and 2,792 speech sentences [10].

- **FUB task:** This task is similar to the Traveler task although, significantly more complex. The corpus used to train the system consisted of the acquisition, transcription and translation of real phone calls to the front desk of a hotel, simulated using Wizard of Oz techniques. It was highly spontaneous and contained many non-speech artifacts. This corpus consists of 3,038 pairs of sentences (Italian-English) with a vocabulary of 2,459 Italian words and 1,701 English words [12].

### 4. Conclusions

A prototype for speech-to-speech translation has been presented. This prototype works close to real-time and requires a low amount of computational resources. The use of finite-state models allows us to perform the translation synchronously with the recognition process. All the models used by the prototype can be learnt automatically. The system is fully available through a normal telephone line. After a year of intensive use (more than 7,000 calls, received), the system behavior has been stable and robust. From a subjective analysis of a random sample set from the 7,000 calls, we found that for the Traveler task, 70% of sentences were correctly translated, 10% were not correct but acceptable enough and 20% were incorrect. Similar results were achieved for the FUB task.

### 5. References


