Speech Translation Statistical System for Teaching Environments and Conference Speeches

Jesus Tomás, Alejandro Canovas, Jaime Lloret, Miguel García
Instituto de Investigación para la Gestión Integrada de Zonas Costeras
Universidad Politécnica de Valencia, Spain
{jtomas, alcasol, jlloret, migarpi}@upv.es

Abstract— **The synergic combination of different sources of knowledge is a key aspect in the development of modern statistical translators. The effect and implications of adding additional other-than-voice information in a voice translation system for teaching environments and conference speakers is described in this work. The additional information serves as the bases for the log-linear combination of several statistical models. A prototype that implements a real-time speech translation system from Spanish to English is presented. In the scenario of analysis a teacher, or presenter, as speaker giving its presentation could use a real time translation system for foreign students or participants. The speaker could add slides or class notes as additional reference to the voice translation system. Should notes be already translated into the destination language the system could have even more accuracy. In this paper, first, we present the theoretical framework of the problem, then, we summarize the overall architecture of the system, next, we specify the speech recognition module and the machine translation module, then, we show how the system is enhanced with capabilities related to capturing the additional information, and, finally, we present the performance results of the developed system.**

Keywords-pedagogical tool; adaptation; speech recognition; speech translation; natural language processing.

I. Introduction

The development of automatic real-time translation systems from voice signals constitutes a long-term objective. However, recent advances in the field of statistical translation increase the possibility of an actual widespread usage in the near future [1, 2].

On one side, an ever-more increasing number of foreign students, interchange programs, and alike in Europe (reaching a 15% of the students in the Higher Polytechnic School of Gandia, Polytechnic University of Valencia), and, on the other side, multi language meetings, workshops and conferences, are requiring more efforts to provide tools and means that help the integration in the learning and speaker presentations processes while the new language skills are getting developed. A tool like the one presented in this article could increase the learning rate provided by the spoken classes and the attendance of foreign-language speakers.

We hence provide a prototype that demonstrates the viability of the real-time speech translation in the pedagogical student-teacher environment. Given the fact that current status-of-the-art products and techniques in the area of automatic real-time translation are far from perfect, we enhance the results by providing beforehand material about the elements of translation, e.g., specific vocabulary, texts, etc. With this purpose, our speech translation system is fed with slides and the class or presentation notes previous to the operation of the system. Often, these sources or information are already translated and handed over to the student. We make use of the offline translation as input to the system as well.

In this paper we explain how to adapt an existing real-time speech translation system to incorporate the usage of the above mentioned additional information, and how this impacts positively in the accuracy results of the translation. Notably, the system improves the alignment using off-line data taken from the tool output. A good environment to apply this system is in teaching, because the teacher's slides and notes could be used as additional information. Another good environment is multi-language meetings, workshops, and conferences. This, together with a very good real time response in the translation recognition, makes our proposal an ideal tool for this type of environments.

This paper is an extension of the paper presented in a conference [3].

The remainder of this paper is organized as follows. Section 2 presents some related works about the speech-to-speech translation systems. Section 3 explains our prototype in order to let the reader know how will work the final product. Statistical Spoken Language Translation used in our system is described in section 4. Section 5 shows the architecture of our system and introduces the modules used in it. The speech recognition module is explained in section 6. Section 7 explains the machine translation module. The adaptation system is described in Section 8. The system evaluation is shown in Section 9. Section 10 concludes the paper and gives our future work.

II. Related Works

Nowadays there are several lines of research in speech-to-speech translation systems. For example NESPOLE!, which is a speech-to-speech machine translation research project funded jointly by the European Commission and the US NSF [4]. The prototype system developed in NESPOLE! is intended to provide effective multi-lingual speech-to-speech communication between all pairs of four languages (Italian, German, French and English) within broad, but yet restricted domains. The idea of this project is to allow a communication online client-server on which both parties are expressed in different languages. The transmitter's phrases are translated and heard by the receiver by means of sensitized speech.

Many research projects have addressed speech-to-speech translation technology, such as VERBMOBIL [5], C-STAR [6], BABYLON [7] and S2ST [8]. The last is quite interesting because it is mainly focused on translation between English and Asian languages (Japanese and Chinese). This requires advanced technologies to overcome the drastic differences in linguistic expressions.

The speech translation system used in our work is based on hidden Markov models and n-grams as [9]. Nowadays, the most successful speech translation systems are based on stochastic finite-state networks. This paper was written using the methodologies developed and the data collected in "The EuTrans-I speech translation system" project. This speech translation is accomplished in using a procedure that is similar as the one used in our speech recognition. Stochastic finite-state transducers, which are specific stochastic finite-state networks, have proved very adequate for translation modeling. The acoustic, language and translation models are finite-state networks that are automatically learnt from training samples. Other interface between automatic speech recognition and machine translation are the confusion networks. In [9], the authors obtained next conclusions: "Confusion networks, from one side, permit to effectively represent a huge number of transcription hypotheses, and from the other side, lead to a very efficient search algorithm for statistical machine translation".

In the paper presented in [10], the problem boils down to the question of how to arrive to a suitable interaction between the recognition process and the translation process. In this study they try to combine distinctive features derived from both modules: speech recognition and statistical machine translation. All the features from the speech recognition and the machine translation modules were combined by the log-linear models seamlessly. It is very interesting for us their conclusions derived from speech recognition: likelihood of acoustic and language models, helped to improve the speech translation. The N-best recognition hypotheses are better than the single-best ones when they are used in translation. They show that N-best recognition hypothesis translation can improve speech recognition accuracy of incorrectly recognized sentences. This same approach has been made in [11]. In this paper, they attempt to derive a suitable Bayes decision rule for speech translation and to present suitable implementations. The authors introduce specific modeling assumptions to convert the Bayes decisions into a practical algorithm. We thought that to compare our work with the works in reference [9] and [11] might be interesting, but it is difficult. The tasks are different and the tools used in these tasks are not open source. In the beginning, we considered to work with one of these tools, but this tool does not compute in real time.

Our translation system is based on the Corpus of the European Parliament. We take this corpus as training data for statistical machine translation. In [12], the authors describe the acquisition of the corpus and its application to the statistical machine translation. These training corpuses are usable thanks to the work performed by some research groups such as "The statistical machine translation group" at the University of Edinburgh. They participated in the transcription and translation tasks for five language pairs, using only the supplied corpora, for example in [13].

In [14], the authors present an overview of their current out-of-the-box system. It includes a detailed treatment of models added over the last years, especially a novel lexicalized reordering model. The system employs a phrase-based statistical machine translation model that uses the Pharaoh decoder [15].

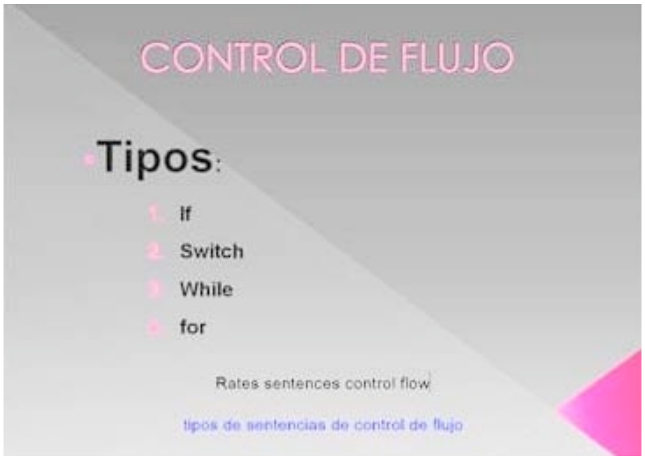
III. PROTOTYPE DESCRIPTION

The prototype presented in this paper implements a real-time speech translation system to support a Spanish speaker with English-speaking speech listeners.

First, the speaker provides the slides in a MS PowerPoint format making sure that the notes area of each slide contains an explanation of the slide. The explanation in the notes area should be very close to what the speaker is going to say explaining the slide.

Before starting the speech, the speaker must load the PowerPoint file in the system. At this point, the system gets adapted to reflect the text within the slides, increasing the probability to guess the right words to appear when each slide is presented. The procedure is used all along the duration of the presentation.

When the presenter is speaking, the system recognizes the sentence, translates it, and is able to display the subtitle translation as caption to the slide. An example is seen in Figure 1. The last two lines are superimposed to the projection of the slide. In the bottom, what the presenter said appears written (in Spanish); while the line above it corresponding translation in English is written. In this way, an English-speaking student is able to relate the content with the Spanish representation through the displayed translation.



CONTROL DE FLUJO

Tipos:

1. If
2. Switch
3. While
4. for

Rates sentences control flow

tipos de sentencias de control de flujo

Figure 1. Slide example.

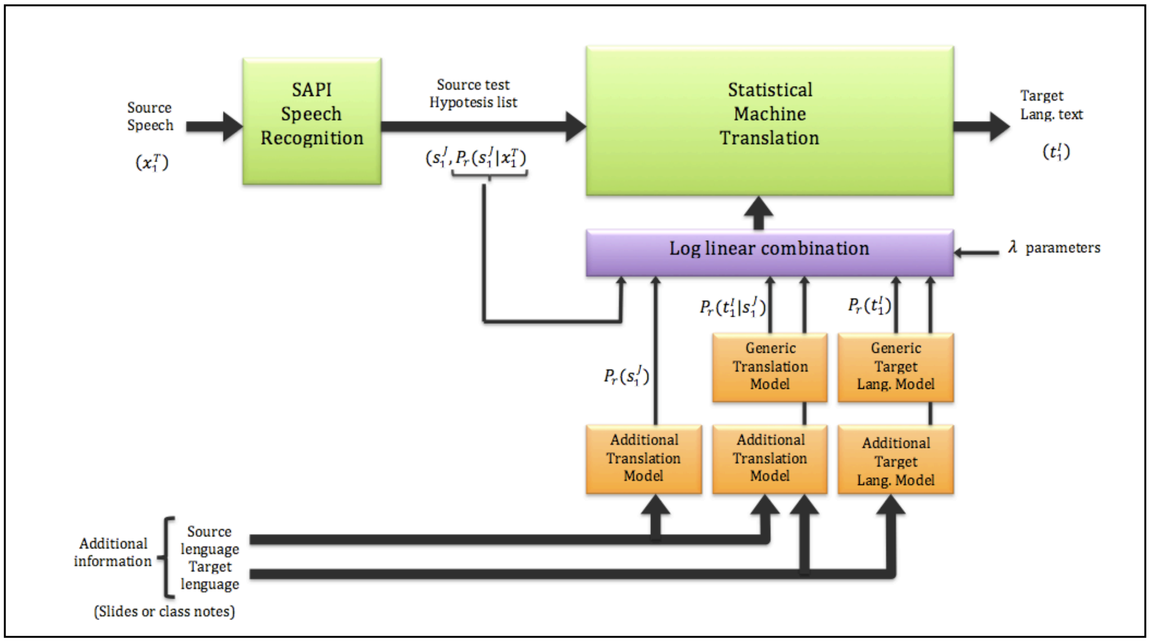


Figure 2. System architecture.

IV. Statistical Spoken Language Translation

The goal of Statistical Spoken Language Translation [14, 16] is to translate a given acoustic observation vector $x_1^T = x_1 \ldots x_T$ into a target sentence $t_1^I = t_1 \ldots t_I$.

The methodology used in [17, 18] is based on the definition of the function $Pr\,(t_1^I|x_1^T)$ that returns the probability that t_1^I is a translation of a given acoustic observation. We can introduce a hidden variable that represents the source sentence, $s_1^J = s_1 \ldots s_J$.Then, we can write equation 1.

$$\hat{t}_1^{\hat{I}} = \underset{t_1^I}{\text{argmax}}\ Pr(t_1^I|x_1^T) = \underset{t_1^I}{\text{argmax}} \sum\nolimits_{s_1^J} Pr\left(s_1^J, t_1^I \middle| x_1^T\right) =$$

$$= \underset{t_1^I}{\text{argmax}}\ \sum\nolimits_{s_1^J} Pr\left(s_1^J \middle| x_1^T\right) Pr\left(t_1^I \middle| s_1^J\right) \approx$$

$$\approx \underset{t_1^I,\ s_1^J}{\text{argmax}}\ Pr\left(s_1^J \middle| x_1^T\right) Pr\left(t_1^I \middle| s_1^J\right) \qquad (1)$$

Following the log-linear approach [10, 11], $Pr\left(t_1^I \middle| s_1^J\right)$ can be expressed as a combination of a series of feature functions, $h_m(t_1^I, s_1^J)$, that are calibrated by scaling factors, λ_m, as it is shown in equation 2.

$$Pr\left(t_1^I \middle| s_1^J\right) = \sum_{m=1}^{M} \lambda_m\, h_m(t_1^I, s_1^J) \qquad (2)$$

This framework allows us a simple integration of several models in the translation system. Moreover, scaling factors let us adjust the relative importance of each model. Bearing in mind this objective, Och and Ney propose a minimum error rate criterion in [19].

V. Architecture

Our system architecture is based in two modules, as Figure 2 shows.

A. *Speech Recognition Module (SRM)*

The Speech Recognition Module (SRM) takes the audio input stream from a microphone and obtains an N-best output text. In the N-best list, each hypothesis is scored according to the equation $Pr\left(s_1^J \middle| x_1^T\right)$.

Although there are several open source speech recognition systems available, like Sphinx [20] or HTK [21], we have used the standard system provided by the MS Windows Vista OS, because it is the only one that incorporates acoustic models for Spanish. They are available to non-restricted tasks. The communication with this engine is based on a SAPI interface [22].

In addition, this engine has a few interesting capabilities that makes it well suited for a real-time application like ours. It let us customize its functionality for a specific speaker and task, which allows us to work with multiple output hypotheses simultaneously.

B. *Machine Translation Module (MTM)*

The Machine Translation Module (MTM) is based on a previous work (described in [23]). Basically, in order to estimate $Pr\left(t_1^I \middle| s_1^J\right)$, a log-linear combination of several statistical models is used. In our application, two models are needed, one for the translation itself and one for the target language selected. A new important feature is introduced in this work, the output score of the speech recognition module.

Therefore the machine translation module integrates the following knowledge:

- Translation model. It is based on monotone phrase-base models. Phrase-base models divide the sentence into segments, each one of them composed by a series of words. Now, the translation probabilities relate a sequence of words in a source sentence with another sequence of words in the target sentence. The simplest and fastest formulation in such models is based on monotone models [23]. However, to operate in real time the speed of the translation search is a critical factor. Thus, we have to select a monotone phrase-based model.
- Target language model: It is comprised of two sub-models, a conventional trigram model, $p(t_i|t_{i-2}^{i-1})$, and a five-gram class model: $p(T_i|T_{i-4}^{i-1})$.
- Speech recognition score: It is the output of the speech recognition module.

VI. SPEECH RECOGNITION MODULE

Generally, a speech recognition application involves the detection of the user's voice and the interpretation of what he/she has said. In our case, the speech recognizer is a fundamental tool of the application. Once the recognizer detects and interprets what the user says, the information is passed to the translation module and, then, it is displayed on the application screen.

The voice detection of our prototype is performed by the SAPI voice recognition engines. Specifically, we use the recognition engine included in Spanish Windows SDK for Windows Vista. Figure 3 shows a diagram of the structure of the speech recognition module. The figure describes and explains how the speech recognition module is integrated into our system.

As it is shown in the previous scheme, the information provided by the speaker's speech flows from the top to the bottom, but the configuration information is from the bottom to the top. For the text to the speech translation scheme, the procedure is more or less the same, but starting from the information based on applications and ending at the speakers.

The high-level interface is implemented in Microsoft COM objects in order to call in a simple way to a low-level interface. It is essential to load the libraries of Microsoft speech recognition in order to access these objects. On the other hand, the low-level interface is implemented by the TTS and SR engines. SAPI interface allows the exchange of these engines without needing to reprogram the application. Therefore, an application can choose between different voice recognition engines, such as the speech recognizer Windows SDK, IBM or the Dragon, among others.

The main interfaces for speech recognition in ISAPI are ISpRecoContext, ISpRecognizer and ISpGrammar. What follows is a brief explanation of a thread of different interfaces in the SAPI speech recognizer.

A. Speech recognition interface

1) ISpRecoContext

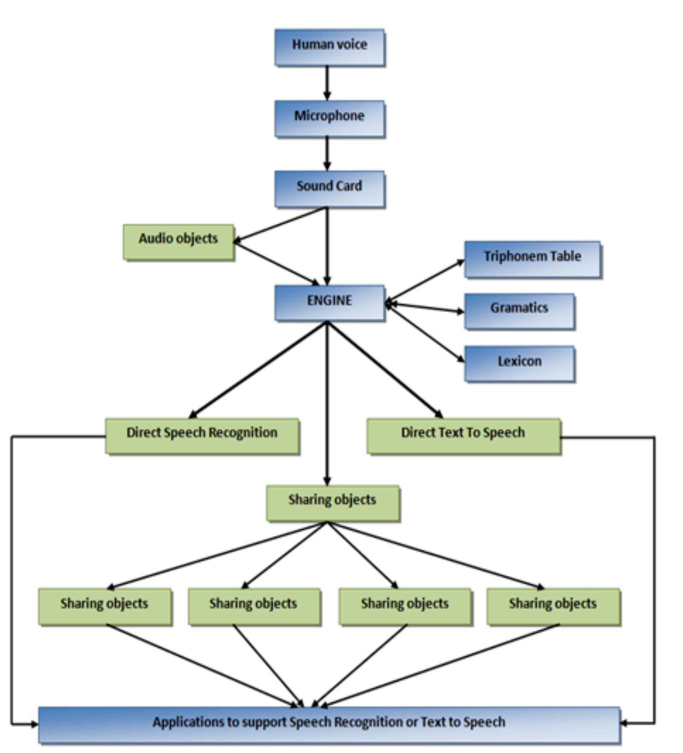


Figure 3. Speech recognition module.

ISpRecoContext is the main interface in the Windows Vista OS speech recognition.

ISpRecoContext interface allows the application to create different functional points of views or context of the SR (SpeechRecognition) engine. ISpRecoContext, as well as ISpVoice, is ISpEventSource. ISpRecoContext is used by the voice application to receive notifications from different events during the voice recognition. Therefore, this object allows an application to start and stop the recognition, receive results from the recognition, analyze the words and sentences pronounced by the user and other events. For example, it will not be the same for an application to use the world "close" in a dictate, being referred to close a door, that "close" to close a desktop application. Both belong to different contexts.

An application could have different contexts. Those words that are not inside the context could be included.

In order to convert the acoustic entrance into a stream in SAPI, first we call the object ISpRecognizer and create a RecoContext using the method CreateRecoContext. Therefore, when we create an ISpRecoContext from an ISpRecognizer, SAPI is converting the voice to text.

2) ISpRecognizer

ISpRecognizer interface allows the application to control some the voice recognition engine (SR Engine) and its audio input. Each ISpRecognizer interface represents a unique SREngine.

There are two possible implementations of the ISpRecognizer in SAPI. One is for the “in-process” (InProc) recognition, where only our application could connect to the recognizer. It is used in the situations in which a maximum performance, lower response time or high quality recognition is being searched. And another implementation is "shared-recognizer", where all voice applications work simultaneously using a shared engine connected to the same recognizer. In this way, when a user talks, the recognition engine will decide which context is the most convenient between all possibilities.

3) ISpRecoGrammar

ISpRecoGrammar interface allows the application to manage the words and sentences recognized by the voice recognition engine.

A SpRecognizer object can have several SpRecoContext objects associated to it and a SpRecoContext object could have several SpRecoGrammar associated to it. This let us create voice recognition applications with several grammars. Therefore, a SpRecoGrammar object could have a contact free grammar and a dictate grammar simultaneously.

B. Voice recognition process execution thread.

In order to create an ISpRecoContext object, first we create an ISpRecognizer "in-process" object from the application. Then, we call ISPRecognizer::SetInput to activate the audio input and an ISpRcognizer::CreateRecoContext to obtain an ISpRecoContext object. Next, we activate the notifications of the events generated by the voice recognition. After that, we create an ISp_Recognition object. This object will inform us if the ISpRecognizer has recognized a voice.

Last, a voice application should create, load and activate a ISpRecoGrammar. This object gives us information basically about the model, that is, if it is a dictate or control commands. In order to create this interface, we use the following call ISpRecoContext::CreateGrammar. Then, the application loads the appropriate grammar, through calling an ISpRecoGrammar::LoadDictation for dictate or an ISpRecoGrammar::LoadCmdxxx for control commands. In order to activate these grammars or models, the application calls to ISPRecoGrammar::SetDictationState for a dictate or to ISPRecoGrammar::SetRuleState or ISpRecoGrammar::SetRuleIdState for control commands.

It is important to highlight that the applications based in voice synthesis use grammars. They should be specified and loaded when the application is starting. The dictate grammar is freer because it allows using higher number of language words. However, the grammar word list of control commands is limited.

Figure 4 summarizes the process previously explained.

It is important to say that voice recognition only happens when a RecognizeStream event is produced by the recognition engine, that is, when a stream is processed by the engine. Then, the result of this event is sent to SAPI. This reply is not sent until the stream recognition is fully finished. This synchronization between SAPI and the recognition engine is controlled by the engine, because until there is not a reply from the engine, there is no recognition.

C. Several events in the application execution.

One of the issues to take into account during the development of our project has been the analysis performed of the produced events when the application is being executed. The application follows the execution thread shown in figure 5. We can see in this figure that the first produced event is sent by the engine when a sound is detected (*On sound Start*). If the sound input set is recognized by the engine, it will be processed as a *stream* until a sentence is formed, creating the state *recognizer State Change* in the recognition. If it the sounds are not recognized, the engine will send an *interference event* in the recognition.

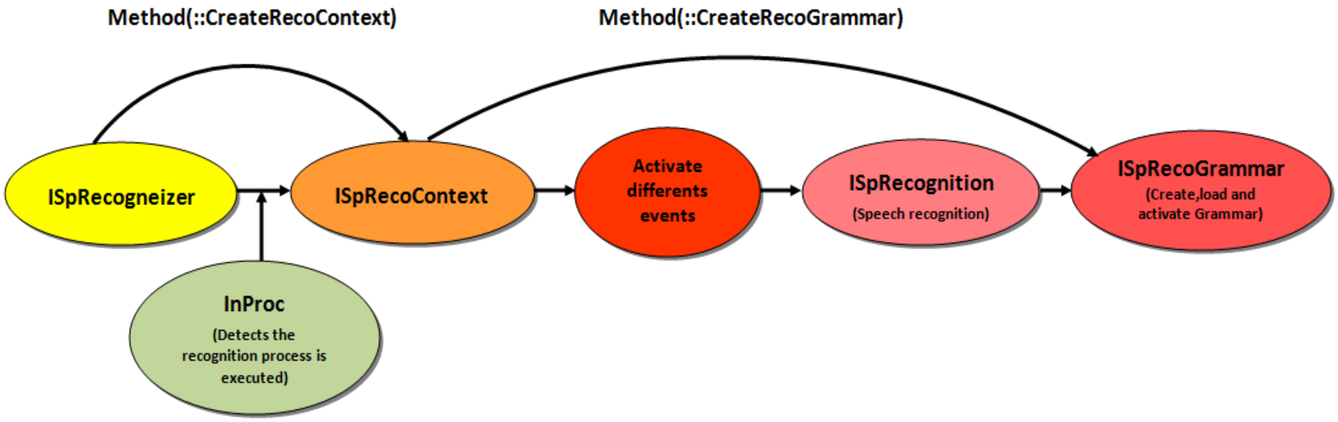

Method(::CreateRecoContext)
Method(::CreateRecoGrammar)
ISpRecogneizer
ISpRecoContext
Activate differents events
ISpRecognition
(Speech recognition)
ISpRecoGrammar
(Create,load and activate Grammar)
InProc
(Detects the recognition process is executed)

Figure 4. Windows interface execution thread.

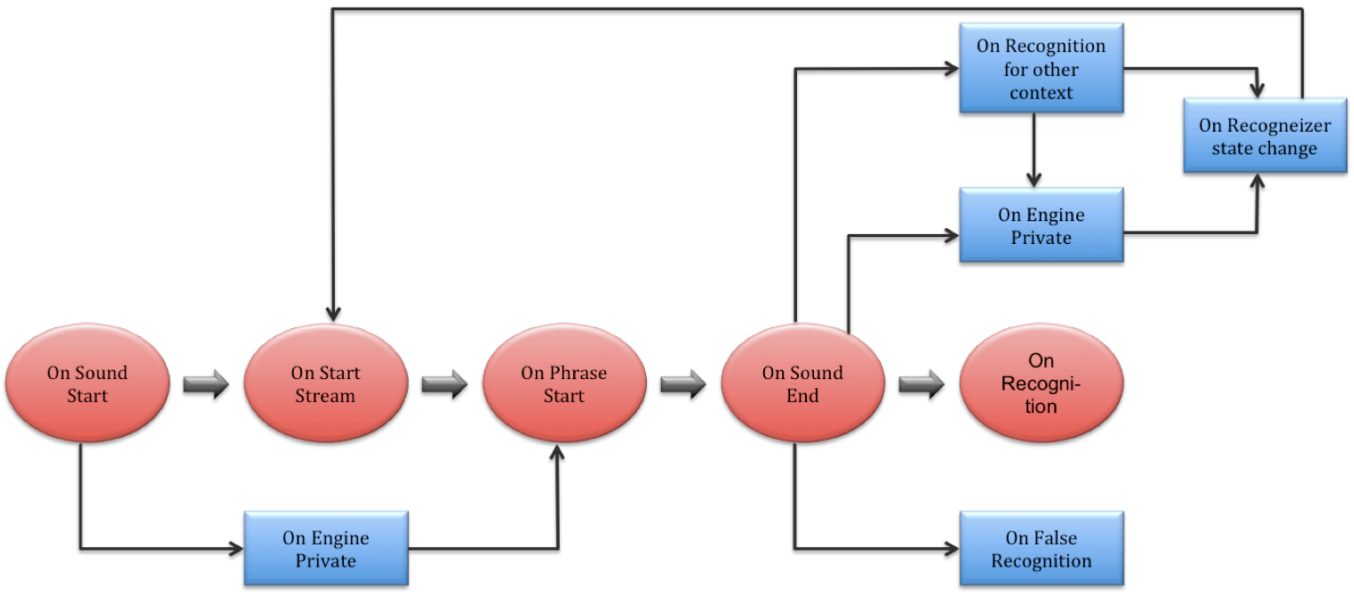


Figure 5. Application execution thread.

When one of these events has been produced, the system re turns an end of sound event and the recognition result. This result can be basically classified in a correct recognition, in a false recognition, or may be the recognition is not considered coherent or may belong to another context.

VII. *Machine Translation Module*

Machine translation module is based in Statistical Machine Translation (SMT). The goal of SMT is to translate a given source language sentence, $s_1^J = s_1 \dots s_J$, into a target sentence $t_1^I = t_1 \dots t_I$. In order to achieve this goal, a function $Pr\left(t_1^I|s_1^J\right)$ is defined. It estimates the probability of obtaining t_1^I as a translation of a given s_1^J. Following the log-linear approach [19], this function can be expressed as a combination of a series of feature functions, $h_m(t_1^I, s_1^J)$, as it is shown in equation 3.

$$\hat{t}_1^{\hat{I}} = \underset{t_1^I}{\operatorname{argmax}}\, Pr\left(t_1^I|s_1^J\right) = \underset{t_1^I}{\operatorname{argmax}} \sum_{m=1}^{M} \lambda_m\, h_m\left(t_1^I, s_1^J\right) \qquad (3)$$

A. *Phrase-Based Models*

For many SMT systems, the most important feature function in equation 3 is the PB model. The main characteristic of this model is that it attempts to calculate the translation probabilities of word sequences (phrases) rather than only single words. These methods explicitly estimate the probability of a sequence of words in a source sentence (s_1^J) to be translated as another sequence of words in the target sentence (t_1^I).

To define the PB model, we segment the source sentence s_1^J into K phrases $(\tilde{s}_1^K)$ and the target sentence t_1^I into K phrases $(\tilde{t}_1^K)$. A uniform probability distribution over all possible segmentation is assumed. If we assume a monotone alignment, the target phrase in *k* position is produced only by the source phrase in the same position [24]. Then we get equation 4.

$$Pr\left(t_1^I|s_1^J\right) \propto \max_{K,\tilde{t}_1^K,\tilde{s}_1^K} \prod_{k=1}^{K} p(\tilde{t}_k\,|\tilde{s}_k) \qquad (4)$$

where the parameter $p(\tilde{t}|\tilde{s})$ estimates the probability of translating the phrase $\tilde{s}$ into the phrase $\tilde{t}$. A phrase can be comprised of a single word (but empty phrases are not allowed). Thus, the conventional word-to-word statistical dictionary is included. If we permit the reordering of the target phrases, a hidden phrase level alignment variable, α_k, is introduced. In this case, we assume that the target phrase in k position is produced only by the source phrase in position α_k. Then, we obtain equation 5.

$$Pr\left(t_1^I|s_1^J\right) \propto \max_{K,\tilde{t}_1^K,\tilde{s}_1^K,\alpha_1^K} p(\alpha_1^K) \prod_{k=1}^{K} p\left(\tilde{t}_k\,|\tilde{s}_{\alpha_k}\right) \qquad (5)$$

where the distortion model $p(\alpha_1^K)$ establishes the probability of a phrase alignment. Usually a first order model is used, assuming that the phrase-based alignment depends only on the distance of a phrase to the previous one [19].

There are different approaches for the parameter estimation. The first one corresponds to a direct learning of the parameters, in equation 4 or equation 5, from a sentence-aligned corpus using a maximum likelihood approach [25, 21]. The second one is heuristic, and tries to use a word-aligned corpus [26, 14]. These alignments can be obtained from single-word models [17] using the available public software GIZA++ [27]. The phrase-based models used in our project have been trained using the second approach.

VIII. ADAPTATION SYSTEM

If available, we make use of additional information, such as text closely related to the speech we are going to translate. This text can be written in the source language, in the target language or in both languages.

The SRM adaptation is performed via the SAPI interface [22]. SAPI adaptation uses specific calls to the SAPI interface. Specifically, we extract each word from the source adaptation data and we use it with the SAPI call *AddVocabulary* to extend the SRM vocabulary.

The MTM adaptation is hence performed as follows: first we train an additional source language model using the source language text, and, then, we train a second additional target language model using the target language text. Finally, we train additional statistical models using both source and target text.

These new models are then incorporated to the system using the loglinear framework. In this framework, each model needs a scaling factor parameter that is estimated by using a minimum error rate criterion [19]. A development corpus is needed for this propose. We can create these adaptation models from a corpus of labeled samples. They will have greater or lesser weight in the system according to the error rate configured for each model.

From a practical point of view, in a first step, in the system adaptation, we consider the choice of the subject on which the presentation will be taught, and what material we can use to improve the training process. The material that can be introduced is classified into three levels:

a) Texts of the slides: The speaker has to provide the slides, that are going to be used during the presentation, to the system. In this case, we used the Power Point format, so the application was programmed to be able to extract text from each slide.

b) Class Notes: Power Point application allows the speaker to introduce in each slide text notes on what is going to be explained. If these notes are introduced, the system will read and use them to guide the system. For this reason, we recommend to include them using a text as equal as possible to the one used in the presentation. For example, if the teacher starts the class telling "welcome to this speech", this sentence should be added in the first slide.

c) General information on the subject: The speaker can include the notes offered to the speech participants or students, reference books and other materials related to the subject. This third level of information has to be introduced to the system using plain text files.

This information may be supplied in the source language (Spanish) and in the target language (English). In addition, for levels a) and b), the system let us introduce this information in a bilingual format. That is, both the slides and the lecture notes can be translated and added into the system.

Additional information is used to adapt both modules of our system. The main adaption is performed in the machine translation module. New statistical models are trained and combined with the models described before using this new information and the log-linear approach [19].

A. Speech recognition module adaptation

1) Add exclusivity to the recognition in our application

One of the problems we encountered during the development of this project is the conflict created between the recognition commands connected to Windows and the ones related to our application. In other words, if we say a word that corresponds to a system command such as "start", then a conflict appears because it is recognized as a command and it is executed.

In order to solve this problem we have added the code shown in figure 6 in the developed application. It gives us exclusivity to our grammar and thus does not recognize any voice-recognition command in Windows Vista.

2) Add new words to the recognizer

In this sub-section we describe how we can add a new word recognizer from the application process. First, the application creates a token object. It assigns several constants and, then, a user lexicon object is created. Secondly, if the user wants to add a word, the application creates an object from the token and the user lexicon object. Finally, the application adds the new word to the user lexicon using the *AddPronuntation*. There are two ways to add new words to the dictionary of the speech recognition engine using the user interface (UI). One is making a call from the same application UI object (running again the screen setup), and the other one is to go directly to the option of "Speech Dictionary" IU recognition engine and select "add a new word". The code to add a new word or phrase into the SR engine in Delphi language is shown in figure 7.

First we added the elements that we need from the palette of elements of the development environment Delphi 7. These objects are SpSharedRecoContext and SpLexicon. They belong to the SAPI interface found in ActiveX. SpLexicon object is used with the *AddPronunciation* method. The way we used *AddPronunciation* method is shown in figure 8. The constant indicating the type of unknown word is *SPSUnknown*. In this case, the pronunciation of the word is added with phonemes.

The procedure in figure 7 let us add the words from the speaker's presentation notes to the user lexicon dictionary in order to improve the recognition. *GetPronunciations* method let us know the words added to the user lexicon.

```
var GramarFija:TOleEnum;
begin
   GramarFija:=SGSExclusive;
   SRGrammar.State:= GramarFija;
end;
```

Figure 6. Added code for having exclusivity.

```
Procedure AddPronunciation(const FileName:string;
          SpLexicon:TSpLexicon);
var
  Ref,fic1,fic2,salida: textFile;
  sAdap_Ref,FileNameSalida,fileWav,palabra:string;
  SREF: TStringList;
  l,posPalabra:integer;
begin
  numRep:=0;sRef:=TStringList.Create;
  assignFile(Ref,FileName); reset(Ref);
  while not eof (Ref) do begin sRef.Free;
   readSoundReferences(Ref,fileWav,sAdap_Ref,sRef);
    posPalabra:=1;
    repeat
        palabra:=readFromStr(sAdap_Ref, posPalabra,
                     ' ,.:;/{}()\"+=');
        if palabra <>'' then
        // We will read every word of the file.
        // txt (sAdap_Ref) and adding to the
        // dictionary SR motor using the method
        // AddPronunciation()
        SpLexicon.AddPronunciation(palabra, 3082,
                      SPSUnknown, '');
    until palabra='';
   end;
end;
```

Figure 7. Code to add a new word into the SR engine.

```
SpObjectToken1.AutoConnect:=true;
SpLexicon1.AddPronunciation('new word',
                         3082, SPSUnknown,'');
```

Figure 8. *SpLexicon* object with the *AddPronunciation* method.

```
SpeechRecoContext.SetAdaptationData(
                       AdaptationString As String)
```

Figure 9. *setAdaptationData* with *SpeechRecoContext* method.

```
procedure  AdaptationData(const FileName: string;
     iteration: integer;
     SpSharedRecoContext: TSpSharedRecoContext);
var Ref,fic1,fic2,salida:textFile;
    sAdap_Ref:string;
    sRef: TStringList;
    n,numRep:integer;
    FileNameSalida,fileWav,srclang,trglang:string;
    valorEdit:string;
begin
  numRep:=0;
  nivelFichTraza:=0;
  sRef:=TStringList.Create;
  assignFile(Ref,FileName); reset(Ref);
  while not eof (Ref) do begin
     sRef.Free;
     readSoundReferences(Ref, fileWav, sAdap_Ref,
                                          sRef);
     for n :=0 to iteration do begin
        SpSharedRecoContext.SetAdaptationData(
                                     sAdap_Ref);
     end;
  end;
end;
```

Figure 10. Code to add a new phrase into the SR engine.

3) Adapt phrases to the recognizer

We utilized *setAdaptationData* method to improve the application. This method is used in the interface ISpeechRecoContext which recognizes different contexts. Generally, setAdaptationData method sends a data string to the voice recognition engine. It is often used to improve the recognition of words (or groups of unfamiliar words), by training the SR engine. Its structure is shown in figure 9.

The code used in the program to adapt different phrases is shown in figure 10.

B. Machine Translation Module Adaptation

In order to adapt our system to a specific topic, additional information must be supplied. As we have described at the beginning of this section, the information is structured in three levels: texts of the slides (level a), presentation notes (level b) and general information on the subject (level c). The following lines explain how we have added the statistical models in the MTM.

1) Specific matter source language model: A trigram language model that is trained using the source training text which has been supplied in level c. It is expressed in equation 6.

$$\mathrm{Pr_m}\ (s_1^J) \propto \prod_{j=1}^{J} p\left(\mathrm{s_j} \middle| \mathrm{s_{j-2}^{j-1}}\right) \tag{6}$$

This model may seem redundant because it is also defined in the SRM. However, our experiments show that it helps the recognizer to have better performance.

2) Specific target language matter model: A trigram language model that is trained using the target training text which has been supplied in level c. It is given by equation 7.

$$\mathrm{Pr_m}\ (t_1^I) \propto \prod_{i=1}^{I} p\left(\mathrm{t_i} \middle| \mathrm{t_{i-2}^{i-1}}\right) \tag{7}$$

This model is vital to guide the translator in selecting the most likely output.

3) Slides source language model: A bigram language model that is trained using the text from the PowerPoint file (levels a and b) in the source language.

$$\mathrm{Pr_s}\ (s_1^J) \propto \prod_{j=1}^{J} p\left(\mathrm{s_j} \middle| \mathrm{s_{j-1}}\right) \tag{8}$$

It is similar to $\mathrm{Pr_m}\ (s_1^J)$, but it is trained on texts that have high probability of being delivered (transparencies and presentation notes).

4) Slides target language model: A bigram language model that is trained using the text from the PowerPoint file (levels a and b) in the target language.

$$\mathrm{Pr_s}\ (t_1^I) \propto \prod_{i=1}^{I} p(\mathrm{t_i}|\mathrm{t_{i-1}}) \tag{9}$$

It is similar to $\mathrm{Pr_m}(t_1^I)$, but trained with slides and lecture notes.

5) Slides translation model: A phrase based translation model that is trained using the bilingual text from two PowerPoint files: the source language and the target language.

$$\Pr_s\left(t_1^I \middle| s_1^J\right) \propto \max_{K,\tilde{t}_1^K,\tilde{s}_1^K} \prod_{k=1}^{K} p(\tilde{t}_k \,|\tilde{s}_k) \qquad (10)$$

In order to train this model we need a bilingual corpus (source and target languages). In our case we have used Spanish-English corpus. For this corpus the presenter has to create two PowerPoint files with the same information in each one of them. The number of sentences in each one of the slides or presentation notes should be equal. In our application, if the presenter loads the slides with different number of sentences, the system displays an error message. The presenter must indicate which file is in Spanish and which one is in English.

IX. System Evaluation

The system described in this paper was assessed through a series of experiments stressing the system in different situations and using three different speakers. The experiments were carried out in a scenario that reproduces the regular conditions of a university class.

The subject selected for the experiments was "Programming", which was taught in the first year of the "Technical Engineer in Telecommunications" degree (at the Higher Polytechnic School of Gandia, Polytechnic University of Valencia). This course describes the principles of C++ object-oriented programming language. This choice was motivated because this subject is being taught in several groups and has a lot of specific information in several languages that allow us to adapt the system. Specifically we used several programming books in both languages and the teacher notes of the subject in Spanish.

In the scenario, a teacher provided a 20 minutes class supported with projected slides and class notes which where beforehand translated into Spanish and English. The class was recorded in an empty room without students for the sake of comparing output results with the same background noise conditions. Sentences from the recording in Spanish were then segmented, transcribed and translated into English.

The sentences obtained where divided into two parts: the test corpus (made by 240 sentences) and the development corpus (made by 120 sentences). The sentences from the test corpus were also recorded later by two additional speakers. Table I represents the different quality features for each speaker.

The generic models of MTM were initially trained by the Europarl corpus [24]. Slides and class notes have been used to train the additional models of MTM. The developed corpus has been used to estimate the lambda parameters using the minimum error rate criteria.

A. Speech Recognition Module Evaluation

First, we performed a set of experiments to assess the SRM module individually. In this set of experiments, we analyzed the influence of the speaker in each type of environment, the type of computer used and the different adaptation methods applied to the SRM.

1) Speaker and speech rate

Three different speakers were used in the test phase. In all three cases the sentences were the same. The first one spoke spontaneously. The other two read a transcript of the class. One of these speakers made an adaptation to his pronunciation. Table I shows the most important features of the three speakers.

Table II shows the speech recognition performance comparison for three test speakers. It is obvious that the speaker adaptation capabilities are crucial to obtain good speech recognition rates. They can be used to analyze the words error rate recognized by the system (WER).

2) Used Hardware

During the development phase we noted that the type of computer used has a very important factor for proper operation of the SRM. In order to evaluate the influence of the hardware at this stage, we performed the test using different hardware equipment. Their details are shown in Table III. Table IV shows the WER obtained from different experiments. We can state that if we have a system with more memory and faster processing capacity, the recognition is improved significantly. The remaining of experiments carried out in this paper were performed using the server.

3) Type of adaptation in SRM

In section VIII we proposed different methods for the SRM adaptation. Its evaluation was performed through a series of experiments. We used the presentation slides and class notes in Spanish as the information provided for the adaptation. The results are shown in Table IV. When new vocabulary words (*AddPronunciation*) are added, the results are significantly improved. Statistical significance is calculated using paired bootstrap [29]. It is better than the baseline with a confidence of 99%. Otherwise, to retrain internal language models (*SetAdaptationData*) does not show any improvement. We have seen that if we repeat this process several times (10 and 100 times), we obtained a slight improvement. But if we use our language model, the results are considerably better. This leads us to the conclusion that the implementation of the SAPI *SetAdaptationData* method is not satisfactory.

B. Machine Translation Module Evaluation

In Table V, different adaptation mechanisms have been compared. In the baseline case, there was no adaptation. The SAPI adaptation mechanism uses specific calls to the SAPI interface. Specifically, we extract each word from the source slides and the class notes to extend the SRM vocabulary. To evaluate the MTM adaptation we considered two cases of sources of knowledge. The first one was only the slides, and the second one was the slides with class notes. Moreover, in each case we tested it just using the source language, and using both source and target language.

TABLE I. SYSTEM QUALITY FOR THREE SPEAKERS

	spontaneous speech	speaker adaptation	gender
speaker 1	yes	No	male
speaker 2	no	Yes	male
speaker 3	no	No	female

TABLE II. SPEECH RECOGNITION PERFORMANCE FOR THREE SPEAKERS

	Speech Recognition (WER)
speaker 1	30.75
speaker 2	15.38
speaker 3	34.5

TABLE III. FEATURES OF THE COMPUTERS USED IN EXPERIMENTS

Model	MacBook Pro	Server
Processor	Intel(R) Corel(TM)2 Duo	Intel® Core™2 Quad Processor Q9400
Processing Speed	2.40 GHz+2.40 GHz	4x2.66 GHz
Operating System	Windows Vista Enterprise (32 bits). Service Pack 1	Windows 7 Ultimate (32 bits)
Memory (RAM)	2,00 GB	4,00 GB

TABLE IV. WER OBTAINED FOR DIFFERENT METHODS OF ADAPTATION AS A FUNCTION OF TYPE OF COMPUTER

Adaptation method	MacBook Pro	Server
Base-line	20.2	17.5
AddPronunciation	19.3	16.7
AddPronunciation+SetAdaptationData (10 times)	19.2	16.8
AddPronunciation+SetAdaptationData (100 times)	19.0	16.4
AddPronunciation + external LM	18.3	15.3

TABLE V. ADAPTATION RESULTS FOR SPEAKER 1

	speech Recognition	Machine Translation	
	(WER)	(WER)	(BLEU)
Base line	17.5	54.2	34.8
+ SRM adaptation	16.5	53.8	35.1
+$\mathbf{Pr}_{\mathrm{m}}$ (s_1^J)	15.3	53.3	35.6
+ $\mathbf{Pr}_{\mathrm{m}}$ (t_1^I)	15.4	42.1	45.7
+ $\mathbf{Pr}_{\mathrm{m}}$ (t_1^I)	9.7	48.4	40.1
+ $\mathbf{Pr}_{\mathrm{s}}$ (t_1^I)	9.7	35.0	56.4
+ $\mathbf{Pr}_{\mathrm{s}}$ $\left(t_1^I \middle| s_1^J\right)$	9.6	34.8	56.5

In order to measure the quality of the translation machine, we analyzed both, the well-recognized words and the well translated, in WER and BLEU measures respectively. We can see in this table that there is an important difference between translation WER and recognition WER. These results depend on the type of task.

The experiments demonstrate how the use of an additional information source really improves significantly the overall results (by a 12%). Particularly, it is most improved when we make use of the class notes in both languages before we start the system. In this case, the accuracy rate increases by a 35%. To provide translations is obviously an extra effort for the teachers, but is often worth doing it when the number of foreign students in the class is high.

X. CONCLUSIONS AND FUTURE WORK

A real-time speech translation system specific to pedagogical environments has been presented in this paper. The main innovation of this work is the way in which additional sources of knowledge are used to improve the accuracy of the system, while remaining practical.

To train the system with other text sources related to the class helps very much the translation (even if they are not the exact notes to the slides). But, we should provide texts related to the same concepts developed in the speech (e.g. reference books of the topic that is going to be presented).

Finally we listed a series of benefits and drawbacks associated with this system. In contrast, we show that a powerful PC microprocessor is required in order to have a good tool performance. Moreover, in order to carry out the training a data collection process is necessary. This data collection process is somewhat laborious for the teacher. This is why we developed a tool to perform this process in real time. Only small breaks between phrases are needed to collect data. This was one of the main objectives, and it has been achieved successfully.

The system described in this work can be used in any pair of languages allowing their translation.

As future work we will improve the system by using models of confusion networks as interfaces between the automatic speech recognition and machine translation modules. Moreover, we are going to add a third language to the system in order to have a system with real-time translation to several languages.

ACKNOWLEDGMENT

This work has been partially supported by the Generalitat Valenciana (Project GV/2009/038) and by the Universidad Politécnica de Valencia.

REFERENCES

[1] Loof, J., Gollan, C., Hahn, S., Heigold, G., Hoffmeister, B., Plahl, C., Rybach, D., Schluter, R., and Ney, H.: The rwth 2007 tc-star evaluation system for europeanenglish and spanish. Interspeech, Antwerp, Belgium (2007). Pp. 2145–2148.

[2] Casacuberta, F., Ney, H., Och, F.J., Vidal, E., Vilar, J.M., Barrachina, S., Garca-Varea, I., Llorens, D., Martnez, C., Molau, S., Nevado, F., Pastor, M., Pic´o, D.,Sanchis, A., and Tillmann, C.: Some approaches to statistical and finite-state speechto-speech translation. Computer Speech and Language 18 (2004) 25–47.

[3] A. Canovas, J. Tomás, J. Lloret, M. García, Speech Translation Statistical System Using Multimodal Sources of Knowledge, The Fifth International Multi-Conference on Computing in the Global Information Technology (ICCGI 2010), Valencia (España), september 20–24, 2010.

[4] Lavie, A., et al. 2001. Architecture and Design Considerations in NESPOLE!: a Speech Translation System for E-Commerce Applications. In Proceedings of the Human Language Technology Conference (HLT 2001), San Diego, CA. (2001).

[5] Verbmobil website, at http://verbmobil.dfki.de/overview-us.html [Last access January 31, 2011]

[6] Hervé Blanchon, and Christian Boitet. Speech Translation for French within the C-STAR II Consortium and Future Perspectives. Sixth International Conference on Spoken Language Processing (ICSLP 2000). Beijing, China. October 16-20, 2000. Vol. 4/4. Pp. 412-417.

[7] Baylon Sppech Engine Website. http://babylon-speech-engine.fyxm.net/ [Last access January 31, 2011]]

[8] Nakamura, S., et al.: The ATR multi-lingual speech-tospeech translation system. IEEE Transactions on Speech and Audio Processing, Vol. 14, No. 2, (2006) Pp. 365–376.

[9] F. Casacuberta, D. Llorens, C. Martíınez, S. Molau, F. Nevado, H. Ney, M. Pastor, D. Picó, A. Sanchis, E. Vidal, and J. M. Vilar: Speech-to-speech Translation Based on Finite-State Transducers. Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing, Salt Lake City, UH, (2001) Pp. 613–616.

[10] Zhang, R., Kikui, G., Yamamoto, H., Watanabe, T., Soong, F., Lo, W. K.: A Unied Approach in Speech-to-Speech Translation: Integrating Features of Speech Recognition and Machine Translation. The 20th International Conference on Computational Linguistics (COLING 2004), Geneve, Switzerland, August 23-27, (2004).

[11] Ney, H. Speech translation: Coupling of recognition and translation. In Proc. Int. Conf. on Acoustics, Speech, and Signal Processing, Phoenix, AR (1999) Pp. 517-520.

[12] Brown, P., Chen, S., Pietra, V. D., Pietra, S.D., Keller, A., and Mercer, R.: Automatic speech recognition in machine translation. Computer Speech and Language 8. (1994) Pp. 177–187.

[13] J. Schroeder and P. Koehn: The University of Edinburgh System Description for IWSLT 2007, Proc. of the International Workshop on Spoken Language Translation, Trento, (2007).

[14] Koehn, P., Axelrod, A., Birch, A., Callison-Burch, C., Osborne, M., and Talbot, D.. Edinburgh System Description for the 2005 IWSLT Speech Translation Evaluation. International Workshop on Spoken Language Translation (IWSLT 2005), Pittsburgh, PA, USA, (2005).

[15] Koehn, P. Pharaoh: a beam search decoder for phrase-based statistical machine translation models. Sixth Conference of the Association for Machine Translation in the Americas. Pp. 115-124. 2004.

[16] J.C. Amengual, J.M. Benedl, F. Casacuberta, A. Castellanos, V.M. Jimenez, D. Llorens, A. Marza, M. Pastor, F. Prat, E. Vidal, J. M. Vilar. The EuTrans-I speech translation system. Machine Translation. vol. 15, Pp. 75 –103, 2000.

[17] Brown, P.F., Della Pietra, S.A., Della Pietra, V.J., and Mercer, R.L.: The mathematics of statistical machine translation: Parameter estimation. Computational Linguistics 19 (1993) Pp. 263–311.

[18] Tomás, J., Lloret, J., and Casacuberta, F.: Phrase-based alignment models for statistical machine translation. In: Pattern Recognition and Image Analysis. Volume 3523 of Lecture Notes in Computer Science. Springer-Verlag (2005). Pp. 605–613.

[19] Och, F.J., Ney, H.: Discriminative training and maximum entropy models for statistical machine translation. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), Philadelphia, PA (2002).

[20] Sphinnx Website. At http://cmusphinx.sourceforge.net/ [Last access January 31, 2011]]

[21] HTK Hidden Markov Model Toolkit home page. At http://htk.eng.cam.ac.uk/ [Last access January 31, 2011]

[22] Hao Shi, A.M.: Speech-enabled windows application using microsoft sapi. International Journal of Computer Science and Network Security 6 (2006) Pp. 33–37.

[23] Tomás, J., Vilar, J., and Casacuberta, F.: The ITI statistical machine translation system. Proceedings of the TC-Star Speech to Speech Translation Workshop, Barcelona, Spain (2006) Pp. 49–55.

[24] Tomás, J., Casacuberta, F.: Monotone statistical translation using word groups. Proceedings of the Machine Translation Summit VIII, Santiago, Spain (2001) Pp. 357-361.

[25] Marcu, D., Wong, W.: A Phrase-Based, Joint Probability Model for Statistical Machine Translation. Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-2002), Philadelphia, PA (2002) Pp. 6 -7.

[26] Zens, R., Och, F. J., Ney, H.: Phrase-Based Statistical Machine Translation. Lecture Notes in Computer Science, Volume 2479, Pp. 35-56. (2002)

[27] Och, F. J., Ney, H.: Improved statistical alignment models. Proceedings of the 38th Annual Meeting on Association for Computational Linguistics (2000).

[28] Koehn, P.: Europarl: A parallel corpus for statistical machine translation. Proceedings of the Machine Translation Summit X, Phuket, Thailand (2005) Pp. 12-16.

[29] Bisani, M., Ney, H.: Bootstrap estimates for confidence intervals in asr performance evaluation. In: IEEE International Conference on Acustics, Speech, and Signal Processing. Volume 1., Montreal, Canada (2004) 409-412