Canovas Solbes, A.; Tomás Gironés, J.; Lloret, J.; García Pineda, M. (2013). Statistical speech translation system based on voice recognition optimization using multimodal sources of knowledge and characteristics vectors. Computer Standards and Interfaces. 1-17. doi:10.1016/j.csi.2012.09.003.

Statistical Speech Translation System based on Voice Recognition Optimization using Multimodal Sources of Knowledge and Characteristics Vectors

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

***Abstract*— Synergic combination of different sources of knowledge is a key issue for the development of modern statistical translators. Reconnaissance, and thus the translation, can be improved by adding new heuristic characteristics. In this work, a speech translation statistical system that adds additional other-than-voice information in a voice translation system is presented. The additional information serves as a base for the log-linear combination of several statistical models. We obtain the characteristics vectors using a statistical model that is based on the N-best reconnaissance list. We describe the theoretical framework of the problem, summarize the overall architecture of the system, and show how the system is enhanced with the additional information. Our real prototype implements a real-time speech translation system from Spanish to English that is adapted to specific teaching-related environments. Finally, we will provide and explain the system performance results. A tool like the one presented in this article may increase the participation rate of the foreign students to the lecture classes and talks.**

Keywords- speech recognition; speech translation; adaptation; pedagogical tool.

1 INTRODUCTION

The development of automatic real-time translation systems when the source is the voice constitutes a long term objective. However, recent advances in the statistical translation research field increase the chances of a widespread usage in the near future [1] [2]. One of the main issues in the deployment of statistical translators is the combination of different knowledge sources. Some of the last proposed systems use more than ten statistical models to guide the translator. Moreover, several research projects demonstrated that the system performance can be improved significantly when multimodal information is arranged in the translator/recognizer entry. Thus, when the system knows beforehand the topic of the speech it can be enhanced.

Current voice recognition and automatic translation systems are far from having satisfactory results. The tools usually have significant number of errors, so they are not adequate for their use in real environments. However, these systems have obtained excellent results in controlled environments such as hotel reception desks [3] and parliamentary speeches [1]. The same case also happens in the conference tutorials and keynote speeches, and in class lectures. Moreover, the speakers use some information as a support for the speech. This information is usually some slides and personal nodes, for his/her own guide, or even a text book.

The application environment, where we have detected the need of the system proposed in this paper, is the university lecture classes. An increasing number of foreign students in European countries, because of interchange programs, and alike, promoted by the bologna process, is requiring more effort to provide tools and means that help the integration of the students in the learning process while the new language skills are getting developed. The duration of the students is short, so it is difficult to learn enough language of the local country to follow the oral classes. So, a tool for simultaneously translate the lecture is very useful for the lecturers and students.

When a degree subject has students from different countries, that speak two different languages (local language and English), there are few ways to solve this problem. One of them is to split the class in two groups, one with the native students and other with the foreign students, but this is not always possible because it doubles the lecturer teaching hours and the university resources. Another one is to suggest the lecturer to translate the subject content into English, so the students can have the content that is going to be explained during the class in both languages. Because this is the lowest cost option, it is the most followed one. Bearing in mind the availability of these slides, with the lecturer notes, in both languages, this information can be used to improve the recognition/translation process.

We hence provide a prototype that demonstrates the viability of the real time speech translation in a real teaching class environment. Moreover, it can be used in conference tutorials, keynote speeches, and so on. Given the fact that current status-of-the-art products and techniques in the area of automatic real-time translation are far from perfect, the results are enhanced by providing beforehand material about the translation elements, e.g., specific vocabulary, texts, etc. With this purpose our speech translation system is fed with slides and class notes that are provided previous to the initial operation of the system. Often these sources, or information, are already translated and provided to the students. This offline information is used as input to the proposed system as well. We also detail how to adapt the real-time speech translation system to add this additional information and how it impacts positively in the accuracy of the results of the recognition, and, thus, the translation system. The implementation part and verification test of the system proposed in this work has been performed in the High Polytechnic School of Gandia, where there are 15% of foreigner students.

The remainder of this paper is as follows. Section 2 shows the works and projects published related with speech translation and some that add other information sources to the translation system. The system is overviewed in Section 3. Section 4 provides the analysis of our statistical model. The system architecture is detailed in Section 5. Section 6 explains the how the system is trained and adapted. Section 7 details the search algorithm. Section 8 provides the test results of our proposed system. Finally, section 9, draws the conclusion and gives our future works.

2 RELATED WORK

The related work section is split in two parts. The first one is focused on the speech translation and the second one is focused on the works that add other information sources to the translation system.

The development of automatic systems for simultaneous voice translation is one of the most pursued objectives in the language's technologies research field. Some good results are appearing in the last works published related with this topic. E.g. EuTrans [3] project demonstrated the feasibility of simultaneous translation in a restricted environment (in this case it was a hotel reception desk). Advances in the speech recognition research field and the statistical translation allowed creating translators/demonstrators in wide environments. An example of these systems is TALES project (developed by IBM), that allows the simultaneous translation of 4 TV channels. Another example has been developed by the University of Carnegie Mellon, in the TC-Star framework project, and allows translating simultaneously conferences from English to Spanish and German. The results of these projects are quite hopeful, but they are quite far from having a correct grade of accuracy to be viable in practice.

Nowadays there are several research lines of speech-to-speech translation systems. One of them is NESPOLE. It 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 receiver by means of sensitized speech. The paper also describes the system architecture. Here are other research projects such as VERBMOBIL [5], C-STAR, BABYLON and S2ST [6] that have also addressed speech-to-speech translation technology. The last one is quite interesting because it is focused on the translation between English and Asian languages (Japanese and Chinese). This requires technologies to overcome the drastic differences in linguistic expressions. The main issue is that existing systems are not of public domain, so they cannot be compared technically and we cannot know their technical features. Moreover, each group has its own scoring system.

In the voice recognition, most systems relay on two main statistical models: acoustic models and the origin language model. Acoustic models form different phonetic segments produced by the human voice started from a sequence of features taken from the signal voice. The acoustic model is usually performed using Hidden Markov Models, HMM [7]. The origin language model tries to distinguish between the entry phrases with high appearance probability and those that are not so expected. This type of model is usually performed using the n-gram concept [8]. An n-gram system estimates the probability of having a word, once it knows last n-1 words. Our speech recognition system is based on this hidden Markov Model and n-grams models.

The translation systems that are actually providing the best results are based on statistical methods. They are mainly supported by two models: the translation model and the destination language model. On one hand, the destination language model discern between output sentences with high appearance probability. Like in the origin language model, the most frequent solution is usually based in the n-gram concept. On the other hand, the translation model is responsible of informing about the most probable translations. Several statistical translation models have been published in the literature. Some of the most well-known are IBM model [9] and

the alignment templates model [10]. However, in the last years the phrase-based models have become very popular [11] [12] [13]. These models have big dictionaries, where there is the probability that a determined phrase of the origin language is translated by a phrase in the destination language.

The speech translation systems based on stochastic finite-state networks are also having high success. EUTRANS system, in [14], was made using the methodologies developed and the data collected during the project. The speech translation is built in a similar way as speech recognition. Stochastic finite-state transducers, which are specific stochastic finite-state networks, are 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. The authors in [14] also describe the advantages of using these networks. On one side confusion networks permit to effectively represent a huge number of transcription hypotheses, on the other side they lead to a very efficient search algorithm for statistical machine translation.

One of the proposals to improve the voice recognition systems and the automatic translation systems is the use of additional information to guide the system to choose the right output. The type of sources providing this additional information could be very different.

Between the projects that use additional information sources we find TransTalk project [15] and an IBM project [16]. Both projects are dictation systems for translators that have the destination language signal voice (provided by a human) and the text of the source language. We can find another example that integrates other information sources to the translation system in TransType 1 and TransType 2 projects [17]. In these projects a translation-aided system is proposed. Starting from the source text, and part of the destination text validated by the user, the system tries to complete the whole destination text. Moreover, in [18], the authors describe a translation system that allows combining source texts of different languages.

A statistical translation module mainly uses a translation module and a language module in order to perform its tasks. Nevertheless, in the last years additional models are added in the log-linear framework [19]. This approximation may be the most appropriate if we want to combine different statistical models with high flexibility. In this approach, each model has a weighed parameter that allows increasing or decreasing the importance of that model. The adjustment parameters can be estimated by using maximum entropy methods or minimizing the errors observed in a validation test.

In [20], 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 the authors try to combine distinct features derived from both modules: speech recognition and statistical machine translation. All the features of the speech recognition and machine translation module were combined by log-linear models seamlessly. They conclude the work letting us know that statistical acoustic and language models helped to improve 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. The same approach has been made in [21]. In this paper, they attempted to derive a suitable Bayes decision rule for speech translation and to present suitable implementations. Authors introduced specific modeling assumptions to convert Bayes decisions into a practical algorithm.

In the last years, the use of characteristics vectors in translation system has been appeared in some published works. Works like the ones presented in [22], [23], [24] are based on the use of confidence measures in statistical translation machines in order to improve the error in the translation. All these papers explain how we can use characteristics vectors from these measures, obtained from the "N-best list" and the first model of IBM, a translation output can be determined from a given input.

As far as we know there is not any real-time translation system used for lecture classes that uses extra information taken from the presentation slides. Moreover there is not any statistical speech translation system published that allows the end user to improve the translation by adding extra information.

3 SYSTEM OVERVIEW

The application environment of our research is a class room where a lecturer is teaching in Spanish and uses the system to translate in real time into English to foreign students, but it can be used in any type of environment with the same features (conference tutorials and keynote speeches, presentation talks, and so on) Our proposal uses a real-time speech translation system to support a classroom of native students with foreign language-speaking students. In our specific case, we use the Spanish as the local language and English as the foreign language.

Initially, the lecturer adds the presentation slides or class notes as additional information of the voice translation system. That is, he/she provides the slides in a tool for creating multimedia presentations (e.g. MS PowerPoint, Impress, PDF, etc.) making sure that the notes area for each slide contains an explanation of the

slide. The information written should be as closest as possible to the speech. It should describe the current slide.

Previous to run the lecture, the lecturer must load the multimedia presentation file in the system. The system takes this information provided by the slides as an external information source and increases the probability to find the correct translated words when the slide is presented and all along the class.

When the lecturer is speaking, the voice signal is gathered by a microphone, the system recognizes the spoken sentence (the most probable sentences), writes the most probable into text, translates it, and displays a subtitle with the translation as a caption to the slide. We can see the blocks diagram in figure 1.

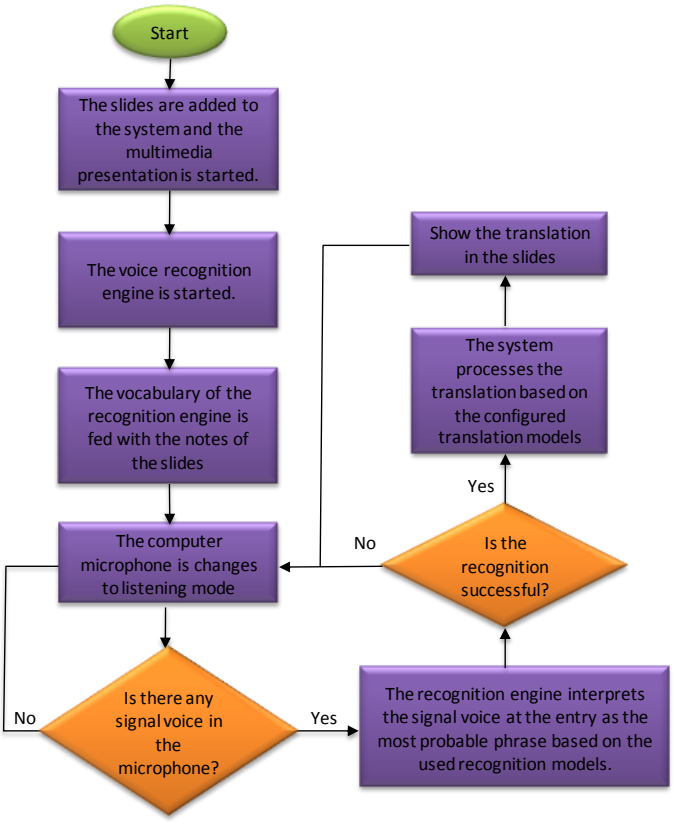

Figure 1. System block diagram.

The system result is seen in figure 2. Last line of presentation is superimposed to the projection of the slide. It shows the translation of each sentence said by the lecturer. In this way, an English-speaking student is able to relate the content with the Spanish representation through the displayed translation.

Figure 2. Demonstration example.

4 STATISTICAL MODEL ANALISYS

In this section we detail analytically the statistical speech language recognition system and the statistical language translation system.

4.1. STATISTICAL SPEECH RECOGNITION SYSTEM

The automatic speech recognition process provides to the machines the capacity of receiving voice messages. It is able to take the voice signal from a microphone as an input. The goal of the automatic speech recognition process is to decode the message obtained from the acoustic sound in order to take the appropriate actions. According to [25], from the beginning of the computer science, and concretely the artificial intelligence, researchers have tried to provide this media to the computers. However, they have split the synthesis from the recognition.

Automatic speech recognition problem can be modeled analytically from a statistical point of view. Given sequence of T measures from the signal voice, that is represented by $x_1^T = x_1 \dots x_T$ and a sequence of I words belonging to a known vocabulary that are represented by $s_1^J = s_1 \dots s_J$, according to [26], the conditional probability $P(s_1^J|x_1^T)$ is the probability of being pronounced the sequence of words s_1^J (from now called phrase) given the observation of the acoustic data x_1^T. Thus, the recognition system will provide as a result the phrase that maximizes that probability. It is given by equation 1.

$$\hat{s}_1^J = \underset{r_1^J}{\operatorname{argmax}}\ P(s_1^J|x_1^T) \qquad (1)$$

We consider equation 1 as the result of the most probable sentence given by the speech recognition system (in Windows Operative System it is performed by SAPI, Speech Application Programming Interface). Using the equation of Bayes we can write the conditional probability as it is shown in equation 2.

$$P(s_1^J|x_1^T) = \frac{P(x_1^T|s_1^J)P(s_1^J)}{P(x_1^T)} \qquad (2)$$

Where,

$P(s_1^J)$ is the probability of the phrase s_1^J or the "a priori" probability of the event s_1^J

$P(x_1^T|s_1^J)$ is the probability to observe the sequence x_1^T when the phrase s_1^J is pronounced. It can also be named as the "a posteriori" probability of x_1^T given s_1^J.

$P(x_1^T)$ is the probability of the phrase given the acoustic information x_1^T.

The probability of $P(x_1^T)$ is the same independently of the pronounced phrase in the maximization process. Thus this probability can be deleted because the phrase that provides the maximum does not vary. Then, we can estimate the automatic speech recognition as equation 3.

$$\hat{s}_1^J = \underset{r_1^J}{\operatorname{argmax}}\ P(x_1^T|s_1^J)P(s_1^J) \qquad (3)$$

The recognized phrase is that one that will maximize the product of two probabilities: $P(x_1^T|s_1^I)$ that relates the acoustic information with the phrase (we call it acoustic model), and $P(s_1^J)$ that only depends on the phrase (we call it language model). Moreover, inside the basic blocks of an automatic speech recognition system we distinguish the training and recognition steps. In the first step, the language and speech models learn from the voice and text. In the second step, the acoustic signal is transcribed to a phrase according to equation 3. But, we increase the number of recognition results that are taken into account. The system builds the "N-best list", based on the acoustic signal of the entry, and we obtain a series of characteristics for each hypothesis. Taking in to account this set of characteristic, as a result of the recognition given by equation 3, the final result of the voice recognition is determined.

4.2. LANGUAGE TRANSLATION SYSTEM

The goal of the Statistical Speech Language Translation is to translate a given acoustic observation vector $x_1^T = x_1 \dots x_T$ into a target sentence $t_1^I = t_1 \dots t_I$ [16] [27]. The methodology used in our proposal is based on the definition of a function $Pr(t_1^I | x_1^T)$ that returns the probability that t_1^I is a translation of a given acoustic observation [28] [29]. We can introduce a hidden variable that represents the source sentence, $s_1^J = s_1 \dots s_J$. Then, we can write equation 4.

$$\hat{t}_1^{\hat{I}} = \underset{t_1^I}{\operatorname{argmax}}\, Pr(t_1^I | x_1^T) = \underset{t_1^I}{\operatorname{argmax}} \sum_{s_1^J} Pr(s_1^J, t_1^I | x_1^T) =$$

$$= \underset{t_1^I}{\operatorname{argmax}} \sum_{s_1^J} Pr(s_1^J | x_1^T)\, Pr(t_1^I | s_1^J) \approx \underset{t_1^I,\, s_1^J}{\operatorname{argmax}} Pr(s_1^J | x_1^T) Pr(t_1^I | s_1^J) \qquad (4)$$

Following the log-linear approach [13] [14], $Pr(t_1^I | s_1^J)$ 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. Equation 5 provides the analytical expression.

$$Pr(t_1^I | s_1^J) = \sum_{m=1}^{M} \lambda_m\, h_m(t_1^I, s_1^J) \qquad (5)$$

This framework allows us a simple integration of several models in the translation system. Moreover, scaling factors allow us to adjust the relative importance of each model. For this objective, Och and Ney propose a minimum error rate criterion [30].

5 SYSTEM ARCHITECTURE

The proposed system architecture is based on two main modules: the speech recognition module (that uses a Speech Application Programming Interface) and the translation module. They are shown in figure 3. The speech recognition module is in charge of transcribing the signal voice from the entrance to a representation close to the natural language (typically a phrase). The translation module receives the information from the speech recognition and translates it to the destination language. The heuristics features are used in the log linear combination for the search module. In order to avoid the accumulation of the errors of both processes, both modules should work coordinated.

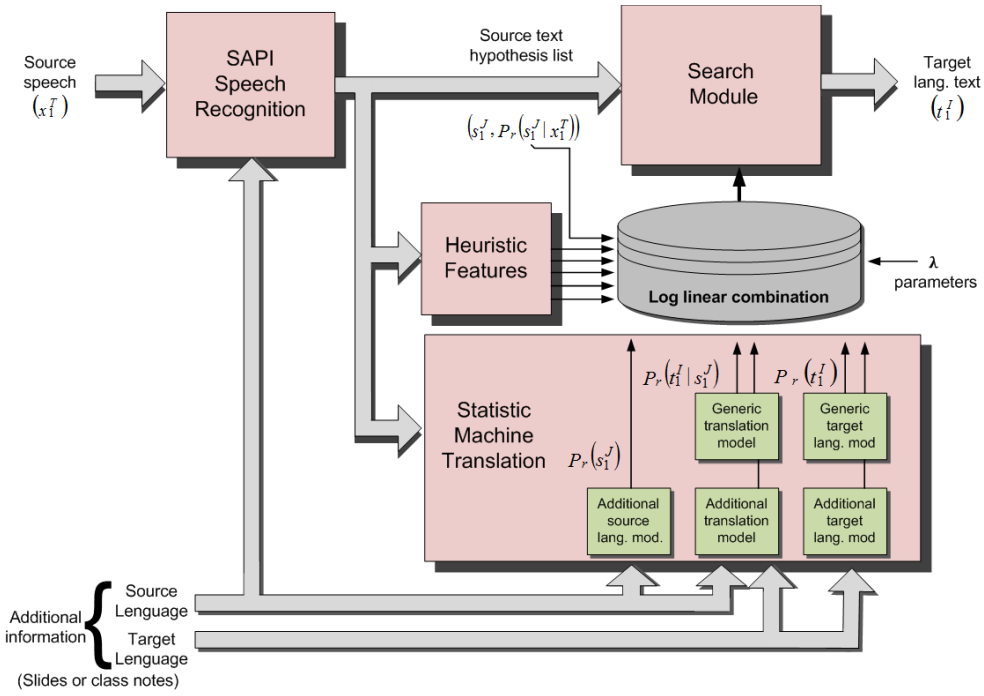

Figure 3. System architecture

5.1. SPEECH RECOGNITION MODULE (SRM)

The speech recognition module (SRM) gets the audio input stream from a microphone and obtains an N-best output list. Each hypothesis in the N-best list is scored according to the $Pr(s_1^J|x_1^T)$. Although there are several available open source speech recognition systems, like Sphinx or HTK, we have used the standard system provided by MS Windows Vista OS, as it seems to be the only one incorporating acoustic models for Spanish. The communication with this engine is based on the SAPI interface [31].

In addition this engine has a few interesting capabilities that make it well suited for a real-time applications like ours. For example, its functionality can be customized for a specific speaker and task which allows us to work with multiple output hypotheses simultaneously. These hypotheses will be used later by the translation module and to extract heuristic features. In the heuristic features module, we will take the appropriate features to classify the hypothesis probabilistically. In the translation module, we also use these hypotheses to probabilistically classify them based on the used models. Once it has made this classification, the system selects the most probable recognition output.

The output of this module is a list of hypothesis. Each hypothesis is represented by (s_1^J, ($p(s_1^J|x_1^T)$). Where s_1^J is a source word sequence and $p(s_1^J|x_1^T)$ is the probability given by the Speech Application Programming Interface. Moreover we also took into account the confidence obtained from each hypothesis. The total number of words of a phrase is also provided by the Speech Application Programming Interface. This value is also a part of a set of heuristic features used for classification.

5.2. MACHINE TRANSLATION MODULE (MTM)

The machine translation module (MTM) is based on a previous work described in [17]. In order to estimate $Pr(t_1^I|s_1^J)$ a log-linear combination of several statistical models is used.

3.4.1 PHRASE-BASED TRANSLATION MODELS

Statistical translation models typically assume that the input and the output sentences can be divided in smaller units. These units are related to each other by means of an alignment and they are translated independently. When the units are words, we make use of the conventional word-based translation models. These include the well known IBM models [9], the HMM based models [32] or the template models [10] (under this model, the lexicon is still word-based and the alignments are restricted by the available templates). On the other hand, phrase-base models divide the sentence in segments, each one composed by a series of words. The translation probabilities now relate a sequence of words in a source sentence ($\tilde{s}$) with another sequence of words in the target sentence ($\tilde{t}$). The simplest formulation with such models is based on monotone models [11]. In this model, the source sentence s_1^J is segmented into *K* phrases ($\tilde{s}_1^K$) and the target sentence t_1^J into other *K* phrases ($\tilde{t}_1^K$). A uniform probability distribution over all possible segmentation is assumed. The monotonicity assumption implies that the target phrase in position *k* is produced by the source phrase in the same position *k*. This can be expressed in equation 6.

$$Pr(s_1^J|t_1^T) \propto \sum_{K,\tilde{t}_1^K,\tilde{s}_1^K} \prod_{k=1}^{K} p(\tilde{s}_k|\tilde{t}_k) \qquad (6)$$

The distribution $p(\tilde{s}|\tilde{t})$ can be interpreted as a dictionary that returns the probability of translating phrase $\tilde{t}$ into phrase $\tilde{s}$. A phrase can be a single word. A conventional word-to-word statistical dictionary can be considered as part of the model. If monotonicity is not admissible, a hidden variable α can be introduced. This represents the fact that the target phrase in position *k* is produced by the source phrase in position α_k (see equation 7).

$$Pr(s_1^J|t_1^T) \propto \sum_{K,\tilde{t}_1^K,\tilde{s}_1^K,\alpha_1^K} p(\alpha_1^K) \prod_{k=1}^{K} p(\tilde{s}_k\,|\tilde{t}_{\alpha_k}) \qquad (7)$$

This model allows efficient search algorithms.

3.5.1 LOG-LINEAR MODEL COMBINATION

The above approach has two problems. The first one is the difficulty of coming up with good models using a generative approach, and the second one is the difficulty to introduce other sources of knowledge in the process. Those problems can be solved by using a log-linear combination of models. In the experiments, we adopted the following log-linear model combination in the monotone search for a given segmentation of $(s_1^J|t_1^I)$ into *K* segments $\sigma = (\tilde{s}_1^K|\tilde{t}_1^K)$. It is shown in equation 8.

$$\begin{aligned} p(x_1^T, s_1^J, t_1^I; \sigma) = {} & \lambda_{10} \log p(s_1^J|x_1^T) + \sum_{j=1}^{J} \lambda_1 \log p'^{(s_j|s_{j-2}^{j-1})} \\ & + \sum_{i=1}^{I} \left[c_1 + \lambda_2 \log p(t_i|t_{i-2}^{i-1}) + \lambda_3 \log p(T_i|T_{i-4}^{i-1}) + \lambda_4 \log p'(t_i|t_{i-2}^{i-1}) + \lambda_5 \log \sum_{\mathrm{j}=1}^{\mathrm{J}} \mathrm{p}(t_i|s_j) \right. \\ & \left. + \lambda_6 \log \sum_{\mathrm{j}=1}^{\mathrm{J}} \mathrm{p}(s_j|t_i) \right] \\ & + \sum_{k=1}^{K} [c_2 + \lambda_7 \log p(\tilde{t}_k|\tilde{s}_k) + \lambda_8 \log p(\tilde{s}_k|\tilde{t}_k) + \lambda_9 \log p'(\tilde{t}_k|\tilde{s}_k)] \end{aligned} \qquad (8)$$

This integrates the following knowledge sources:

- *Speech recognition model*. The probability obtained from the SRM: $p(s_1^J|x_1^T)$.
- *Additional language models for the source language*. A conventional trigram model is used: $p'(s_j|s_{j-2}^{j-1})$. Commonly this model is introduced in the voice recognition phase. In fact, the SAPI SRM

includes a generic Spanish language model. We use additional information that is trained from the text that is provided by the teacher. This model is trained using this data.

- *Generic language models for the target language*. There are two models, a conventional trigram model, $p(t_i|t_{i-2}^{i-1})$, and a five-gram class model, $\log p(T_i|T_{i-4}^{i-1})$. As explained above, the aim of these models is to guarantee that the resulting sentence is correct in the target language. Word classes are obtained using the software *mkcls* [33].
- *Additional language models for the target language*. A conventional trigram model is used: $p'(t_i|t_{i-2}^{i-1})$. This model is trained using the additional target information provided by the teacher.
- *Generic Translation model.* We use the combination of four models. On one hand the simple translation models (like IBM model 1) both direct $(\mathrm{p}(t_i|s_j))$ and inverse $(\mathrm{p}(s_j|t_i))$. These models act as "smoothers"' for the translation probabilities. On the other hand, direct and inverse phrase based translation models: $p(\tilde{t}_k|\tilde{s}_k)$ and $\log p(\tilde{s}_k|\tilde{t}_k)$. These are the most complex models and should capture the main bulk of the work.
- *Additional Translation model*. A direct phrase based translation model is used: $p'(\tilde{t}_k|\tilde{s}_k)$. Trained from the additional information.

Each of the sources is controlled by a weight (a scaling factor) and the λ_i. Two penalties c_1 and c_2 are included to control the values of *I* and *K*.

Summarizing, we used a standard model based on phrases and n-grams, which has been widely analyzed in previous works ([17] and [29]). Moreover, in this paper we have used new methods to improve the translation model by combining different models obtained from various sources, such as slides.

3.6.1 ADDING NEW HEURISTIC FEATURES

Starting from the output hypothesis obtained in the previous subsection, which form the N-best List, the heuristic features can be estimated. These features have been obtained from the knowledge, such as Levenshtein distance and confidence values, and from inference, such as obtaining the most probable hypothesis based on the number of words of the phrase.

A scheme with the modules used in the recognition process, which uses characteristics vectors, is shown in figure 4. First, the recorded sentences are captured using the Speech Application Programming Interface. Then, the information is processed and interpreted using the voice recognition engine. The hypotheses are obtained based on the N-Best List. Next, the characteristics of each hypothesis are estimated and are stored in vectors. These vectors are used to estimate which hypothesis is the most probable and uses it as the output phrase. Figure 4 shows the described modules.

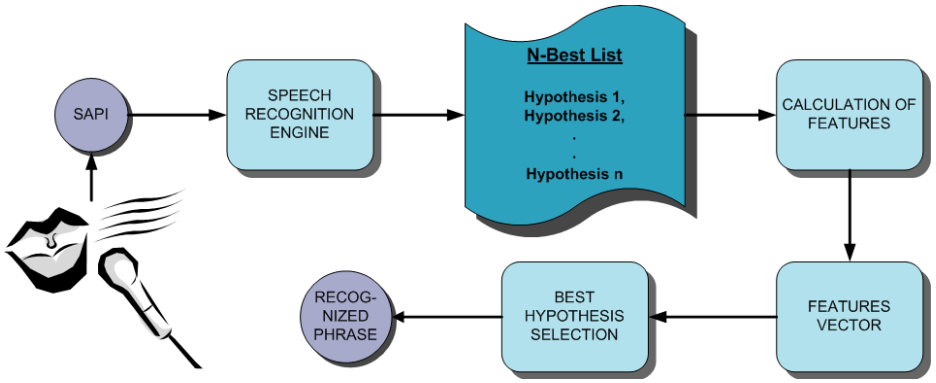

Figure 4. Modules used to extract the heuristic features.

We have added 7 characteristics vectors. They are the following ones:

- Probability based on the number of words.

In order to obtain this characteristic, first we store in a vector the number of words that has each hypothesis. If there are many hypothesis with the same number of words, the most probable recognized phrase is one of them, not other with different number of words. Based on this, those few phrases with different number of

words will be penalized, and those ones with the same number of words will be rewarded. In order to estimate this characteristic, the size of each phrase is compared with the mean value. In the algorithm we write one when the number of words is equal to the mean value and zero when it is different. This is obtained by using equation 9.

$$C_{0,i} = \frac{\sum_{n=1}^{H^*} \delta(J_n = J_i)}{H^*} \qquad (9)$$

Where, J_n is the hypothesis to be compared, J_i is the rest of hypothesis, and H^* is the total number of hypothesis.

At code level, we estimate the number of words of each phrase and the result will be stored as a characteristic of a characteristics vector. Then we compare this value with the rest of hypotheses, so we obtain a probability value. The closer it is to the rest of hypotheses, the higher probability.

- Confidence value given by the recognition engine.

Voice recognition engine can also provide the confidence value for each hypothesis belonging to the "N-Best List" (Windows SAPI provides it). "N-Best List" is obtained from the recognition process. The method used in this process is explained in [34] and its analytical demonstration in [35]. The confidence level of the phrases let us know detect those phrases that are not recognized properly but they are included in our system [36]. Those phrases that have high confidence level will be more preferred tan the low level ones. This confidence values is a floating point value. The *EngineConfidence* feature of the *ISpeechPhraseRule* interface [34] provides this feature.

- Confidence value given by SAPI

This confidence is different of the recognition engine confidence. It averages the confidence level of each Word of the phrase and sets 0 or 1 values. Zero value is given to a very high confidence level, while 1 value is given to a medium-low confidence level. This value will determine the most appropriate sentences because there will be very few with 0 value, while most of them will have 1 value. This feature also has an analytical explanation in [35]. Moreover, [34] explains how this method can be used. Based on the obtained confidence, the hypothesis will be ordered from the highest to the lowest probability value. *ISpeechPhraseRule* interface [34] provides this feature.

- Levenstein distance

Levenshtein distance (also called edition distance or distance between words) is the minimum number of operations required to transform a chain of words to another. It is an operation, insertion, deletion or substitution of a character [37]. We developed Levenshtein algorithm to estimate this distance. Then, we added it to the characteristics vectors as a new feature.

It is obtained performing the following steps. First, we compare the recognition output phrase with each hypothesis. This phrase matches the result of the partial hypotheses provided by the voice recognition engine, so it is the most probable phrase. Then, we estimate the Leveshtein distance from each hypothesis to the most probable. The lower is this value; the more similar are the phrases. The result is added to the characteristics vectors.

- Probability at word level given a language model

In this step, we start with a corpus that has been preciously trained and validated. It provides us the language model. This model is based on trigrams and a linear interpolation smoothing type. This model is included in the system. Using some libraries created by us in [11] we are able to estimate the probability of a phrase given a language model based on n-grams. Finally it is added to the characteristics vectors.

- Probability at phrase level given a language model

The process used to obtain this characteristic is similar to the process explained in the last point, but in this case the probability is obtained at word level. Now, instead of calculating the probability of the whole phrase, we estimate the probability of each isolated word. Then, we add these results and obtain the characteristic value

for the whole phrase. In this case we do not obtain the probabilistic value which is dependent to the n-grams, or to the accompanying words, but of each isolated word. The process to get this probabilistic value is the following one: the system goes through each word of the phrase and estimates the probabilistic value at world level according to our language model. It is performed for all words of the phrase and this value is added to the result obtained for each word of the phrase. The final result is saved as its characteristic value.

- Joint Probability

Finally, we use joint characteristic based on the probability of the phrase according to a language model and the confidence value. This confidence value is given by the recognition engine. We selected these two characteristic because we appreciated in several tests that both characteristics have much effect in the evaluation results. The result of this combination is given by the following expression 10.

$$\mathrm{P}(\mathrm{c_j}) = (1-\alpha)_{1\ldots\mathrm{T}} \times Pr\left(s_1^J \middle| x_1^T\right)^{\beta} \qquad (10)$$

Where, $(1-\alpha)_{\mathrm{n}}$ is the confidence level of the *n* hypothesis, and $\mathrm{P}(\mathrm{X_n}|\mathrm{W})^{\beta}$ is the probability of the *n* hypothesis given the W language model. β let us vary the weight for this factor. In our case we set up β= 1.

6 SYSTEM TRAINING AND ADAPTATION

The models described in the previous section are composed of millions of parameters that must be learned from the training corpus. If available, we make use of additional information, such as text, closely related to the speech that we are going to translate. This text can be written in the source language, in the target language or in both.

Two types of models are used. Generic models, which are learned from a large Spanish/English corpus that does not correspond to a particular task; and adaptation models, learned from corpus provided by the teacher. The training of the generic models can be previously performed. Adaptive models are learned in the adaptation phase.

6.1. SPEECH RECOGNITION ADAPTATION

There are two main ways for adapting the speech recognition module (SRM) using the SAPI interface [16]. SAPI adaptation uses specific calls to the SAPI interface. On the one hand, we extract each word from source adaptation data and use it with the SAPI call *AddVocabulary* to extend the SRM vocabulary. These words, which initially were not recognized by the system, were later recognized by the speech recognition engine in many cases when we added to the vocabulary. On the other hand, we are able to adapt the language model used by the speech recognition engine using *SetAdaptationData* method. This method is included in SAPI libraries. In this way, we are increasing the recognition engine lexicon and language model by adding these words or phrases which are difficult to recognize by the engine. We have appreciated that this adaptation method is good, but it is not as effective as last method.

6.1.1. MACHINE TRANSLATION TRAINING

There are different approaches to estimate the parameters of previous equations. Details of the estimation of monotone and no-monotone phrase-based models can be found in [37]. Some of these techniques correspond to a direct learning of the parameters from a sentence-aligned corpus using a maximum likelihood approach [11][39]. Other techniques are heuristics based on previous computation of word alignments in the training corpus [40][41]. Word alignments are the basis of the most widely used methods for finding bilingual segments. However, the word alignment models usually adopted do not permit the alignment of one source word to many target words [9]. The strategy proposed in [33] and [42] deals with this problem in two steps. In the first step, *symmetrized alignments* are computed from the alignments obtained in a translation direction (s→t) and the alignments obtained in the opposite translation direction (t→s). Different combinations of these two types of alignments were proposed in [43] (*intersection*, *union* and *refined*). From these symmetrized alignments, the bilingual segments are built following different criteria in the second step [43]. These criteria consider that a segment from a source sentence and a segment from a target sentence give way to a bilingual segment. This happens if all the words in the source segment are aligned (according to the symmetrized

alignments) with one word in the target segment and vice versa. Adjacent or internal (source or target) words that are not aligned with any (target or source) word can also be added to the bilingual segment.

In this work, an alternative strategy is proposed. It also consists in two steps but they are different from the steps proposed in [42]. In the first step, two sets of bilingual segments were obtained: separate PB models (bilingual segments and the corresponding probabilities) were built, one model from word-alignments in one direction (s→t) and another model from word-alignments in the opposite direction (t→s). The bilingual segments are obtained following a similar procedure as the one in the second step of the method proposed in [42]. In the second step of our strategy, these two models are combined using log-linear interpolation.

6.2. MACHINE TRANSLATION ADAPTATION

The MTM (Machine Translation Module) adaptation is hence used as follows. Using the source language text, an additional source language model is trained, and, using the target language text, a second additional target language model is trained. Finally, using both source and target text an additional third translation model is trained.

These three new models are then incorporated to the system using the loglinear framework. In this framework each model needs a scaling factor parameter.

6.3. ESTIMATION OF MODEL SCALING FACTORS

The scaling factors can be estimated by optimizing the value of a training criterion over a development corpus [44]. In our case, the optimization consisted in minimizing the difference between the translation word error rate and the BLEU (Bilingual Evaluation Understudy) scores. The optimization was carried out using the downhill simplex algorithm [45].

7 SYSTEM SEARCH

Given a source speech x_1^T, the aim of the search in statistical translation is to obtain a target sentence $\hat{t}_1^I$ that maximizes equation 11:

$$\hat{t}_1^I = \underset{J,s_1^J,I,t_1^I;\sigma}{\operatorname{argmax}}\ p\left(x_1^T, s_1^J, t_1^I; \sigma\right) \qquad (11)$$

The search algorithm is a crucial part in a real time statistical speech translation. Its performance directly affects the quality and efficiency of the translation. In this section, we describe two search algorithms which are based on multi-stack-decoding [46] for the monotone (equation 6) and for a non-monotone version (equation 7) [47].

The most common statistical decoder algorithms use the concept of partial translation hypothesis to perform the search. In a partial hypothesis, some of the source words have been used to generate a target prefix. Each hypothesis is scored according to the translation and language model. In our implementation for the monotone model, [11] we define a hypothesis as the triple (J', $t_1^{I'}$, g), where J' is the length of the current source prefix (i.e., that prefix is $s_1^{J'}$), $t_1^{I'}$ is its translation and g is the score of that translation computed from equation 11.

7.1. MONOTONE SEARCH

The translation procedure can be described as follows. The system maintains a large set of hypotheses, each of them with its translation score. The set is divided in lists so that each hypothesis in the list covers the same number of source words.

Within each list the hypotheses are sorted according to the translation score. The algorithm consists in an iterative process.

In each iteration, the system extracts from each list the best scored hypothesis and extends it. The extension consists on selecting one or more untranslated source words and to attach one or more target words to the current output prefix. The extension of a hypothesis can generate hundreds of new hypotheses. The process is iterated *Max-iter* times. Thus, at most *Max-iter* hypothesis are extended from each list. At most *Max-iter*

extensions are done (they can be less if in a given moment there are not enough hypotheses to extend). The output of the search is the hypothesis with highest score and with no untranslated source words.

7.2. NON-MONOTONE SEARCH

If a non-monotone model is used, the search can be made by using *target-word reordering* (TWR) [47]. Here, we define a hypothesis like in the monotone algorithm, and each hypothesis is also stored in a separate list according to the source-length prefix. In contrast to the monotone case, we can introduce the special token <nul> in the target hypothesis. The meaning of this token is that, in a future expansion, the token <nul> must be replaced by a sequence of words. In our implementation, we allow only one token <nul>. Therefore, we can distinguish between two classes of hypotheses. A hypothesis is closed if it does not contain the token <nul>, and it is open if it contains this token.

In the process of extending a partial hypothesis, those bilingual phrase-pairs ($\tilde{s}$, $\tilde{t}$), in which $\tilde{s}$ matches the source segment after the last translated word, are considered. On the one hand, if the hypothesis to be extended is closed (it has no <nul> token), two new hypotheses are created by adding $\tilde{t}$ and <nul>$\tilde{t}$, respectively, to the target prefix. On the other hand, if the hypothesis is open, four new hypotheses are created: one closes the hypothesis by replacing the token <nul> by $\tilde{t}$; and three new open hypotheses are obtained by putting $\tilde{t}$ to the left or to the right of <nul> and at the end of the target prefix. We have a different parameter distortion for each type of extension. If the hypothesis is closed, we use the probability p_0 to open it, and $1- p_0$ to keep it closed. If the hypothesis is open, we use the probabilities p_c to close it. (1- p_0)/3 is used for the other three extension types. A decoding example using this algorithm is shown in Figure 5. The Spanish sentence "A la bruja verde Mario dio un bofetada" is translated into the English sentence "Mary slapped the green witch". Partial hypotheses are stored in sorted list with numbers from 0 to 8. In each hypothesis, the first J' words of the source sentence have been translated.

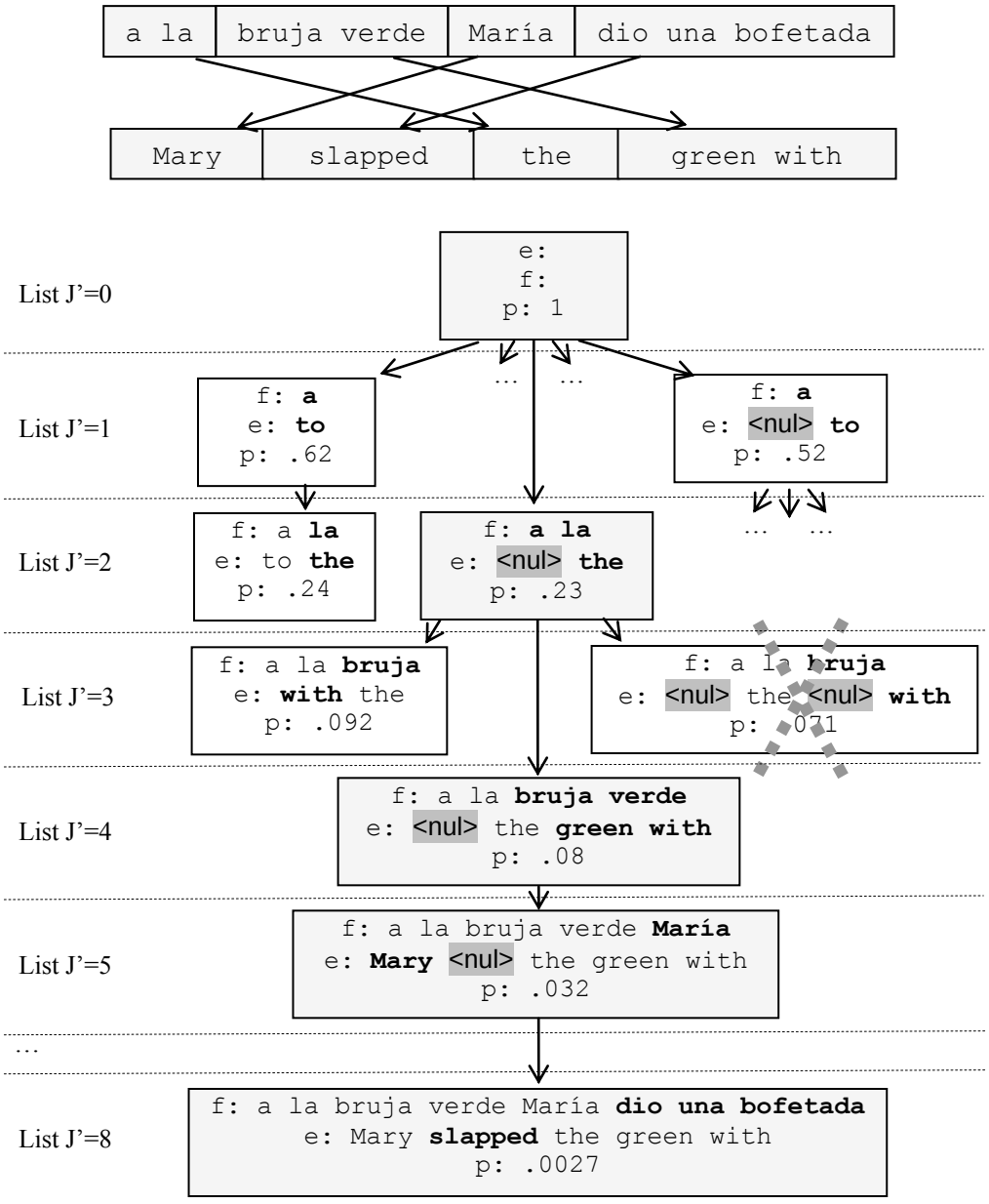

Figure 5. Decoding example using TWR algorithm.

The restriction to at most one <nul> token implies very similar practical costs of the monotone and non-monotone search algorithms. In practice, the parameter *Maxiter* can been used to increase the speed of the translation.

The language model causes another problem. In an open hypothesis we cannot calculate the language model contribution of the right part of the prefix after the <nul> token. In order to solve this problem, we compute an estimation of the language model contribution. It consists of assigning the probability of its unigram to the word at the right of <nul> times the probability of the bigram for the next word. When a hypothesis is closed, this estimation is replaced by the true language model contribution.

There are some proposals ([48],[49],[10]) that try to solve this problem by selecting source words in different positions (SWR) and generating the target words left to right. In this approach, a partial hypothesis is a triple

$(C,t_1^{l'}, g)$, where C is the coverage set (the source positions that have been translated). Hypotheses with the same number of elements in C are stored in the same list. As it happens in TWR approach, J such lists are needed.

Compared with TWR, this approach has different limitations. First, each partial hypothesis needs a coverage set, which increases spatial costs. Second, hypotheses with different C can be stored in the same list. In order to properly compare the hypothesis that covers different parts of the source sentence, estimation (usually heuristic) of the contribution to the score of the parts, that are not yet covered, can be introduced [50]. Third, in order to reduce the computational cost of the algorithm, re-ordering restrictions must be introduced. For example, [46] proposes that only the first l yet uncovered word positions can be translated. Fourth, many different paths lead to hypotheses that have the same coverage set and target segment. In order to reduce the search space hypothesis, recombination is used. In our approach this phenomenon is less frequent, and hypothesis recombination is not essential. Moreover, the main limitations of the TWR approach are the following ones. First, in an open hypothesis, an estimation of the language model is used. Second, only one token <nul> is allowed. Then, we do not cover all possible reordering.

8 SYSTEM EVALUATION

In order to evaluate the system we have implemented an assessment tool (see figure 6). It allowed us to perform various tests in order to improve the recognition system. The tool has been programmed in Pascal using a Delphi programming framework. With this tool we analyzed specifically:

- The recognition result of the microphone input given by the recognition engine.
- Partial hypotheses during the speech phrase.
- The N-best list with possible phrases together with the characteristics vector values for each sentence.
- The resulting sentence and its translation depending on the used translation models.
- Result of the WER

The hardware used to perform this task has been a 2Quad CPU Q9400@2.66GHz microprocessor-based PC Intel® Core™, with 4GB of RAM and Microsoft Windows 7 Operative System.

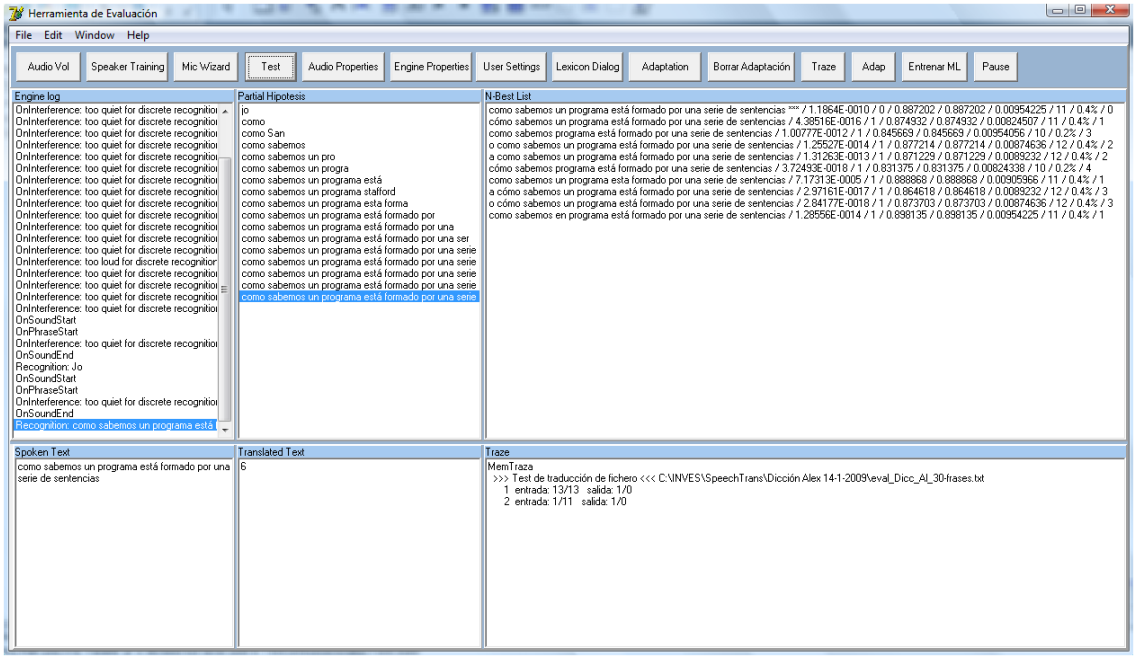

Figure 6. Interface evaluation tool used to avaluate our system.

After having integrated the tool in the system, and having performed a recognition test, we seek to improve the error related with each recognition process and have it into account in the prototype. In order to improve the

recognition error we have focused our efforts in two main tasks: the adaptation of the system and the use of an optimum characteristics vector.

We performed a series of experiments in order to test the system in different situations and using one or three different speakers depending on the situation. Next subsections show the obtained results.

8.1. RECOGNITION ERROR EVALUATION USING DIFFERENT SYSTEM ADAPTATION METHODS

8.1.1. MICROPHONE VOLUME AND GAIN ADAPTATION

After integrating the tool into the system, and performing the recognition, we investigate the way to improve the recognition error to create a prototype with the best performance. In order to perform this recognition error improvement we primarily focus on two tasks: to adapt the system and to use the optimal characteristics vector.

First, in the adaptation process, we analyze the external variables influence such as the microphone volume, or gain, during the recognition process. For this experiment, we realize a test corpus based on a series of phrases. Next, a speaker reads these phrases, which are recorded. Then, they are played by the tool and the Word Error Rate (WER) is estimated based on the test corpus. The WER results are obtained by comparing the output of the recognition system with the test corpus. In order to perform this test process, the system plays each recorded phrase and estimates the instant WER for each phrase. Finally, after finishing all recorded phrases, the average instantaneous value of all instantaneous values is estimated. The WER results obtained in this experiment are shown in figure 7.

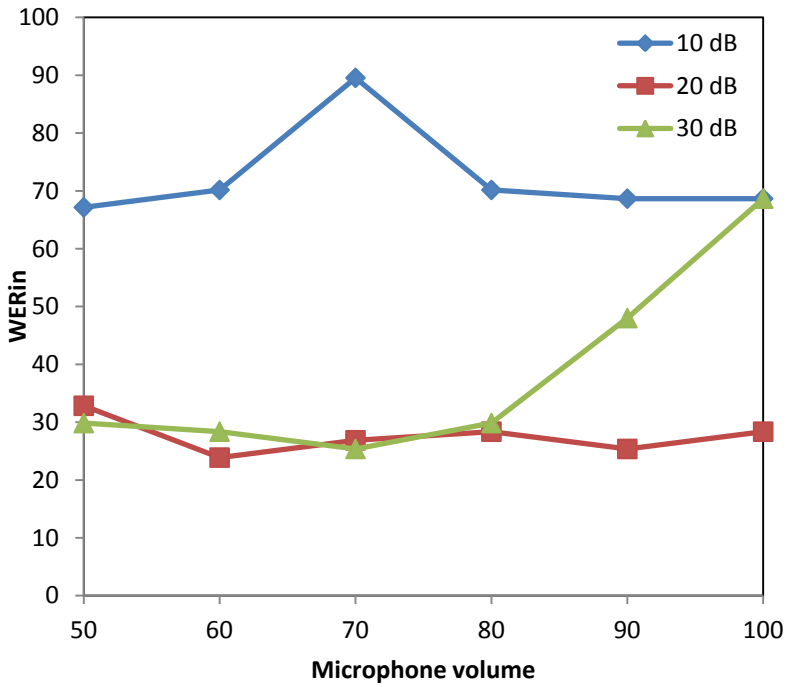

Figure 7. Speech recognition WER with different microphone gains and volumes.

These tests have been performed with a specific computer and, therefore, if the characteristics of the hardware change these values will change too. The purpose of this analysis is to demonstrate the influence of microphone volume, and gain, in the recognition process. Therefore, the volume is a variable that must taken into account before using the recognition tool.

8.1.2. DIFFERENT SPEAKERS ADAPTATION

Experiments have been run in a scenario that reproduces the regular conditions of a university class. In this scenario, a teacher taught class during 20 minutes supported by some slides and lecture notes which where

beforehand translated from Spanish to 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 were recorded in Spanish, and then were segmented, transcribed and translated into English. The obtained phrases were divided into two parts: the test corpus (made of 240 sentences) and the development corpus (with 120 phrases). Phrases from the test corpus were also recorded later by two additional speakers. Table 1 represents the different quality features for each speaker.

In Table I, spontaneous speech column is used to inform us if the speaker has made the diction of the sentences of the test corpus spontaneously, or, otherwise, if he/she has read the test. The speaker adaptation column refers to whether the speaker has made the pre-process using Microsoft Windows' speech recognizer adaptation, or not. In this case, the speaker previously reads a series of sentences in order to adjust the system.

TABLE I. QUALITY FOR THREE SPEAKERS

	spontaneous speech	speaker adaptation	genre
speaker 1	yes	no	male
speaker 2	no	yes	male
speaker 3	no	no	female

In Table II, the speech recognition performance of three test speakers is compared. The best WER result is 16.5 (obtained by the speaker 2). This speaker is the one who has performed an adaptation preprocess to the system by using Microsoft Windows' speech recognizer. In order to do it, the speaker reads the sentences displayed by the application. The system self-adapts to the reader dynamically. Then, it is evident that the speaker adaptation capabilities are crucial to obtain good speech recognition rates.

TABLE II. SPEECH RECOGNITION PERFORMANCE FOR THREE SPEAKERS

	WER Speech Recognition
speaker 1	30.75
speaker 2	16.5
speaker 3	34.5

8.1.3. **SAPI ADAPTATION**

Another process that can be performed in order to adapt Microsoft Windows' speech recognition system is to add to the recognition engine vocabulary new words and also phonetically adapt the system with new words and phrases. The process is explained in the next subsection. The average best results are shown in table III. The WER result for different phrases compared with other experiments is shown later (in figure 13).

8.1.4. **LENGUAGE MODEL ADAPTATION**

In the previous experiments we used a common language model based on "Europarl" [51]. In this subsection we will test how the language model affects to the recognition system. In order to achieve this goal, we performed the following tasks.

In order to train the language model, based on the lecture notes corpus, we performed the following training process. It has been performed by creating a language model that is used later in our evaluation tool.

First we added the teacher's lecture notes into a text file. This will be our reference corpus. This corpus is loaded in a software tool, called Stat Trans, implemented by our research group (see figure 8). It is very easy to use it, we just have to click the button marked in figure 8 and select teacher‘s notes text file.

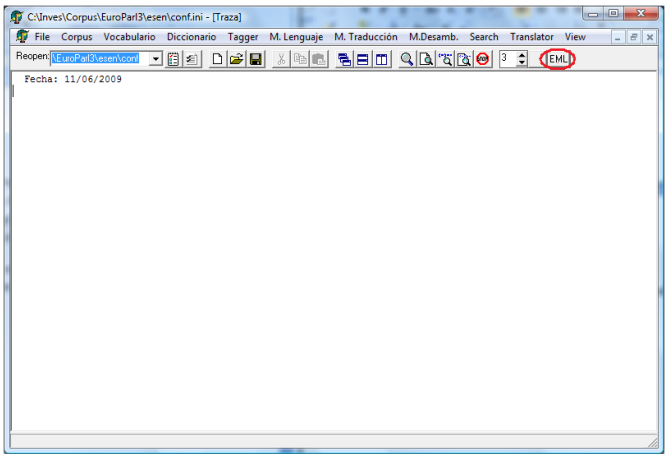

Figure 8. Tool Stat Trans

Then, the tool displays the window shown in figure 9. It let us set the type of language model and the size of the n-grams that will be used to estimate the probability.

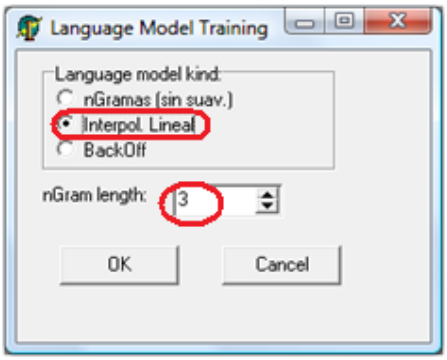

Figure 9. Type of language mode and n-grams lenght value.

The type of language model selected in our case is the linear interpolation [5]. In this method, and also in the backoff method, a smoothing is performed. Smoothing is a technique for solving null probability problems and low estimations when there are few occurrences. It allows us to modify the estimations of the Maximum Likehood Estimator (MLE) to avoid having an n-gram with zero probability. In the interpolation, the probabilities of different n-grams are combined in order to obtain the new probability. In our case, we have selected 3 n-grams. The validation values that we have considered optimal, after different tests or experiments, for these n-grams are: λ_0=0.01, λ_1=0.09, λ_2=0.3, and λ_3=0.6.

The formula used for linear interpolation is shown in equation 12.

$$\hat{P}(wn|w\,n-2\,w\,n-1) = \lambda_3 P(wn|w\,n-2\,w\,n-1) + \lambda_2 P(wn|\,w\,n-1) + \lambda_1\,P(wn) + \lambda_0 \quad (12)$$

Where $\sum_i \lambda_i = 1$.

Once the corpus training and validation has been ended, we obtain the language model. The program will give us information such as the vocabulary size, number of n-grams observed, number of words and number of sentences. Figure 10 shows the result of an example provided by our software tool.

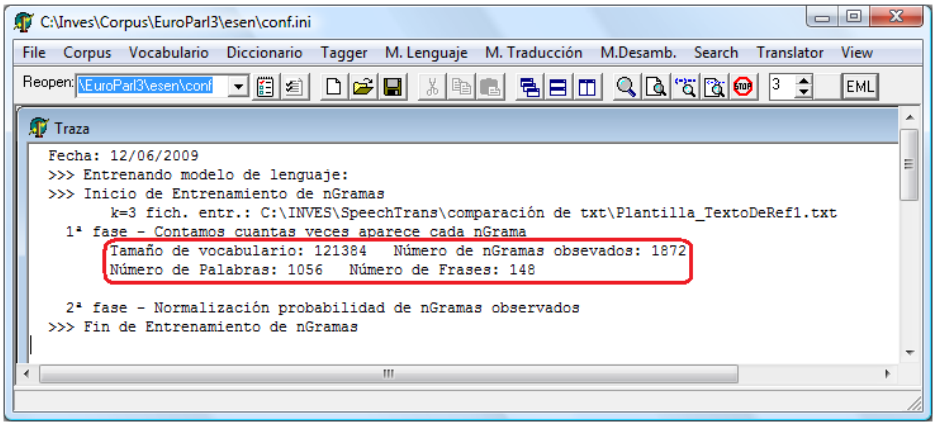

Figure 10. Language model train result.

Finally, in order to load the new language model training corpus, we must specify the directory where the corpus configuration file is going to be saved. The file is called "conf.ini" and the saved values are available from the same application (see figure 11). Among the most interesting values are the *ScalingFactor* and *pAdjustLong*. The first value gives the weight to have the probability of the language model in the translator. The effect of this value will be analyzed in the speech recognition phase.

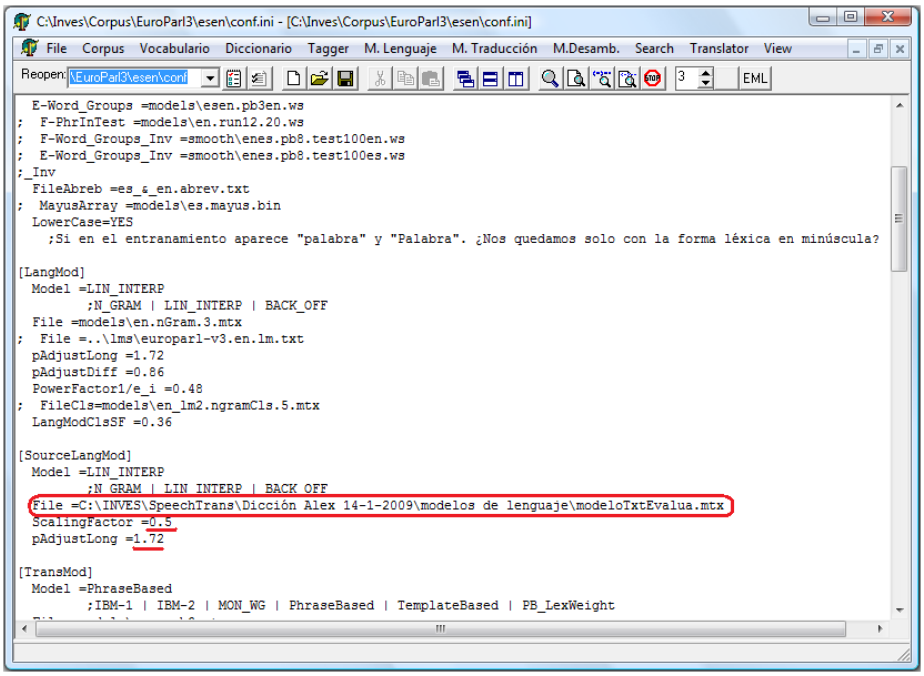

Figure 11. Corpus configuration screen.

The generic models of the MTM were initially trained by the Europarl corpus [51]. It was used to train the generic models of MTM. Moreover, slides and lecture notes was also used to train the additional models of MTM. The developed corpus is used to estimate the lambda parameters of the n-grams using the minimum error rate criteria.

In Table III, different adaptation mechanisms are compared. As baseline there was no adaptation used. The SAPI adaptation mechanism uses specific calls to the SAPI interface. Specifically, we extract each word from the source slides and the lecture notes to extend the SRM vocabulary. In order to evaluate the MTM adaptation, we considered two sources of knowledge. The first one just uses the slides or the slides with class notes (combined with the source language), and the second case uses both source and target language.

TABLE III. PERFORMANCE ARCHIEVED DIFFERENT ADAPTATION SOURCES FOR SPEAKER 1

	Speech Recognition (WER)	Machine Translation (WER)	(BLEU)
base line	17.5	54.2	34.8
+ SAPI adaptation	16.5	53.8	35.1
+ source slides	15.4	53.3	35.6
+ target slides	15.4	42.1	45.7
+ source lecture notes	9.7	48.4	40.1
+ target lecture notes	9.7	35.0	56.4

As it is shown in the above table, the results obtained, when the system is fed by the lecture notes, are greatly improved in terms of WER. This happens because we feed the system with information that is very similar to that spoken by the speaker. This information makes also to improve the quality of the translation and thus the result of BLEU.

We have also analyzed the influence of the ScalingFactor (S_F) in the recognition result. This variable determines the weight to the probability of each phrase in the speech recognition result according to our language model. That is, the more the S_F value of a phrase is, the higher is the dependence of the speech recognition result on the obtained probability (according to our language model) than on the Microsoft Windows speech recognition engine. Figure 12 shows the input WER (WERin) versus different S_F values. We can see in this figure that the optimal S_F value to improve the recognition is S_F=0.5.

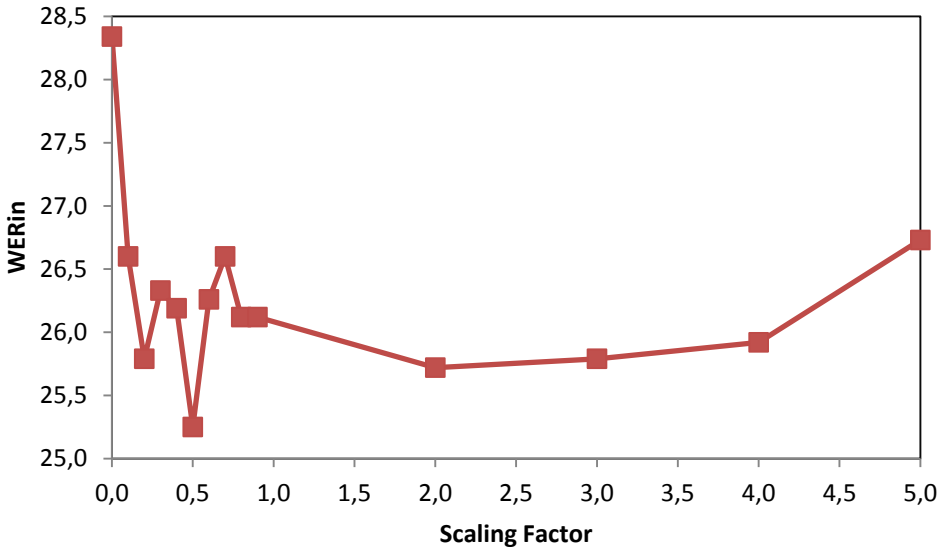

Figure 12. WERin values for different S_F values.

Figure 13 shows the comparison between a text without adaptation, a text using an adaptation type based on feeding the vocabulary of the speech recognition engine and applying an adjustment of Scaling factor equal to 0.5, and a text using the same language model than the previous case of adaptation.

We can see in the figure that to apply a new language model improves the recognition. In addition, with the appropriate S_F value, the results are even better. It not only happens in some particular phrases but also all along the test.

When we analyzed the comparison of several phrases, some interesting results were obtained. For example, in the 60th phrase the system adaptation performed efficiently because the system recognized an English word in the Spanish text. It weren't able to recognize it without adaptation. Table IV shows how was recognized in each case.

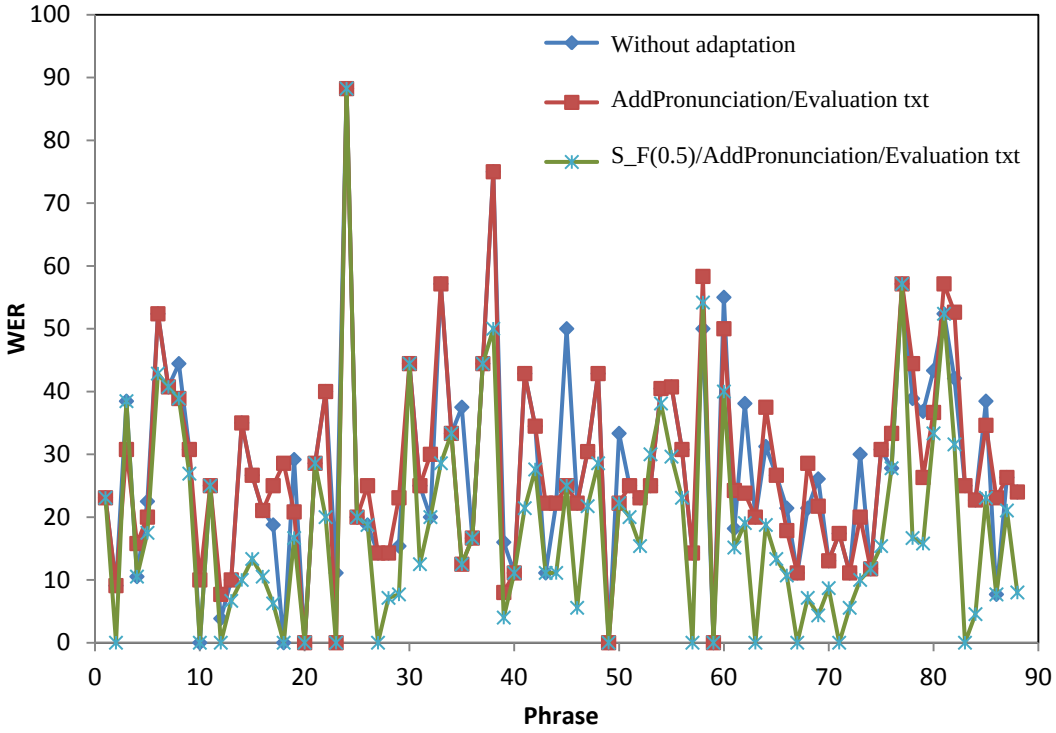

Figure 13. Different adaptation measures comparison.

TABLE IV. EXAMPLE: RECOGNITION OF THE 60TH WORD FOR SEVERAL ADAPTATION CASES.

	Translated word
Without adaptation	bucle Ford
AddPronunciation / txt evaluation	bucle foro
S_F (0.5) / AddPronunciation / txt evaluation	bucle for

Finally, another issue that is analyzed in this section is the language model used in the adaptation or training system. In this test, first, different language models have been trained, and, then, they have been adapted to the system. For the test bed described in this work, we have used four corpuses to perform the language models. They are the following ones:

- Thinking in C++: It is a book to learn to program in C++ language. Therefore, this book is very much related with the experiment topic.
- Lecture Notes: They are the teacher's lecture notes. This case is very small because it is an abstract about one day class.
- Evaluation Test: They are the sentences exactly as they have been said by the teacher in the simulation of a class.
- Europarl: This corpus is quite much larger than the others, but is not related to the experiment topic.

In order to perform this experiment, we have adapted the system with each corpus previously described using the procedure explained in this section. After training the system with each corpus, we performed the test. Then, we obtained WER values. Figure 14 shows the recognition error results using different language models. In this figure we can observe that if the corpus content is more similar to the content said by the speaker, and if it is large and there is some work on it, the recognition results are better.

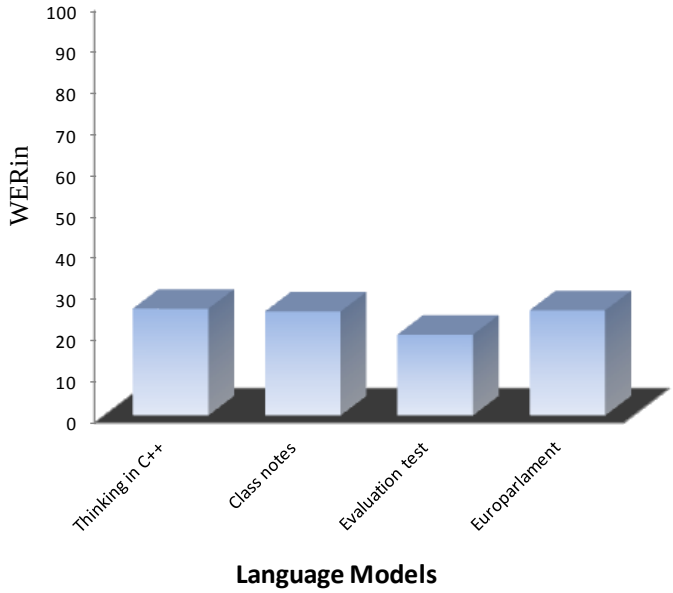

Figure 14. WERin for different language models

8.2. RECOGNITION ERROR EVALUATION USING CARACHTERISTICS VECTORS

In our last experiment, we optimize the recognition error using characteristics vectors based on the baseline of the adaptation task. After the adaptation process, the recognition result will be used as baseline. The evaluation process of this task is divided into the next 3 phases.

8.2.1. EVALUATION OF EACH CHARACTERISTIC INDIVIDUALLY

First, we have implemented the code needed to create the characteristics vector. Once we completed it, we have evaluated each characteristic individually. The feature extraction is performed as it has been explained in sub-section 3.6.1. The recognition system output depends on the highest or the lowest value, depending of the characteristic, for each hypothesis. For this, the evaluation tool selects one N-bestlist phrase depending of the selected characteristic. For example, the system estimates the confidence value given by the recognition engine of each hypothesis or N-bestlist. These values are saved in the characteristics vector. Then, we go through the vector. The hypothesis with highest value will be selected as the speech recognition output. In order to do it, the code checks each characteristic value and saves it in the memory if the value is greater (or lower, depending on the characteristic). The phrase corresponding to this value is saved in memory too. At the end of the process we get the best value and the phrase corresponding to it is saved, thus we optimize the application.

8.2.2. EVALUATION OF ALL CHARACTERISTICS TOGETHER

In this case instead of evaluating each characteristic individually, we evaluate all characteristics at the same time. In order to do it, we add (or subtract, depending of the characteristic) all characteristic values for each hypothesis. The subtract is performed when minor characteristic values provide better recognition result (e.g. characteristics 1 and 3, that correspond to SAPI confidence value and Levenshtein distance) and we add in opposite cases (other characteristics). The highest value provides the system speech recognition output.

8.2.3. EVALUATION OF ALL CHARACTERISTICS TOGETHER WITH DIFFERENT WEIGHTS

The process is similar as the one described in the previous subsection. But in this case different a weight is assigned depending on the characteristic. Therefore, for each hypothesis, we estimate all characteristics providing a specific weight and we add them. Finally, we select the hypothesis with maximum value. Equation 13 follows the described procedure.

$$\hat{s}_1^J = arg\max_{s_1^J} \sum_{n=0}^{h} \sum_{i=0}^{c} \lambda_i \times P(c_j) \qquad (13)$$

Where, W is the words sequence (phrase or hypothesis) that maximizes the argument value, h is the total number of hypotheses, λ_i is the weight of each characteristic and $P(c_i)$ is the probability value of each hypothesis according to the characteristic set c_i.

The weight we give to each characteristic is respect to the others, therefore it follows a probability distribution whose sum is 1 and depends on the results of the first evaluation. The best results have greater weight than the others. In order to have a consistent distribution, some characteristics may need a pre-process. This pre-process mainly consists on normalizing the characteristic values (that is, to have values between 0 and 1) and to change the tendency to those ones that having low values is better. Now we can analyze all positive being the higher value the best one for all characteristics. Once we have normalized the characteristic values, we evaluate the system as we have explained before. The obtained results are shown in table V.

The table shows that in four tests we obtain better results than in the test bench. One of them is when we take into account the probability characteristic that depends on the number of words. There is other that takes into account the probability characteristic at phrase level given a language model. Another adds all the characteristic values. Finally, the best result is provided by adding all characteristic values taking into account their weights. We highlight that there is a significant improvement (4.66%). We obtained a negative in the probability at word level given a language model. Now, we can state that this characteristic should not be used in this case. It is mainly because a language model based in n-grams performs well with more than one n-gram, but not with isolated words.

TABLE V. PERFORMANCE ACHIEVED WITH ADAPTED SYSTEM AND USING CHARACTERISTICS VECTORS FOR SPEAKER 1

	WER Speech Recognition
Baseline	15.4
Words number	15.18
Engine confidence	17.72
SAPI confidence	17.72
Levenshtein distance	17.26
ML (word) probability	20.05
ML (phrase) probability	13.07
Joint Probability	15.86
Sum of features	11.67
Sum of features with weights	10.74

On the other hand, we also evaluated the instantaneous WER. That is, the WERin of each one of the phrases provided by the recognition system. The results are shown in figure 15.

We can see that, generally, in the very long phrases WER results are very similar for all tests. Even in the 5th phrase, the result is the same. On the other hand, the WER is higher in short phrases (such as 19, 20 and 24). We have noticed that the characteristic related to Levenshtein distance is quite irregular, and the word level probability given a language model has poor performance in general. Finally, we think that the most important result is the WER values obtained from the evaluation of all characteristics with different weights, because it has the lowest value in all phrases.

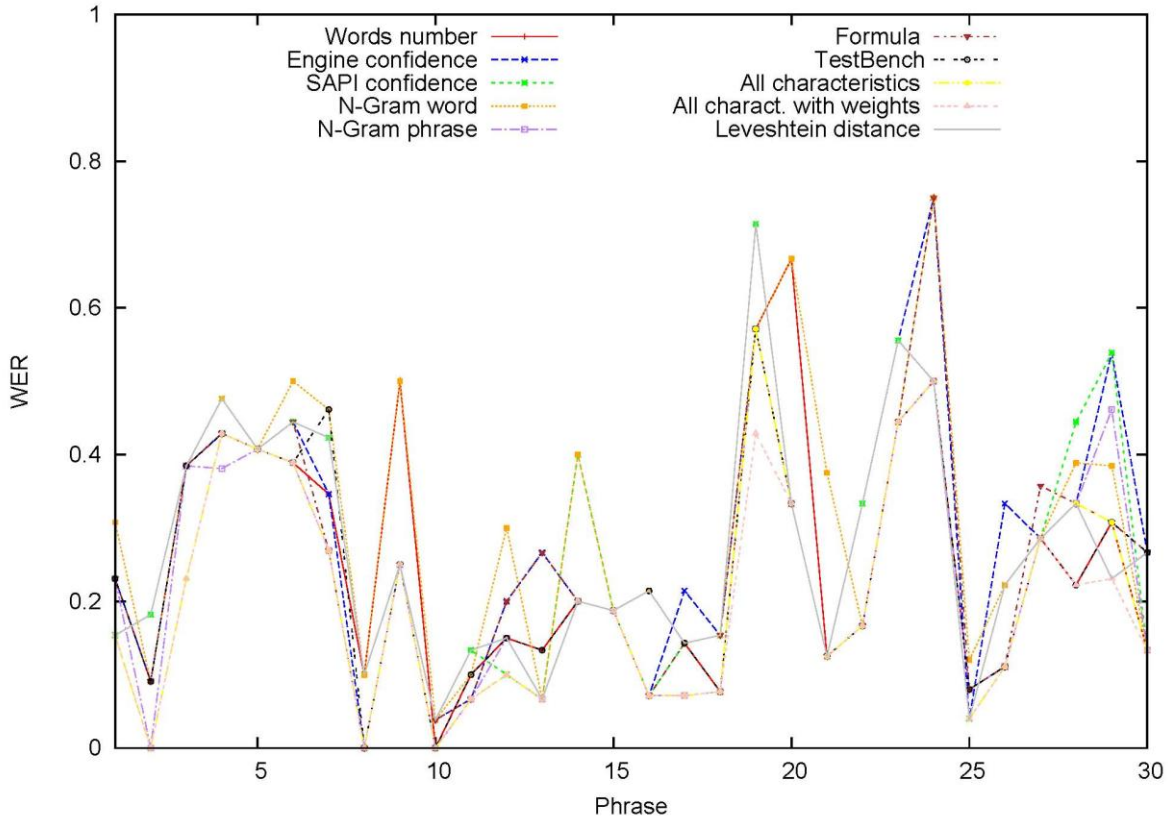

Figure 15. Instantaneous WER results obtained for different tests

In order to analyze in depth how affects the length of the phrase on the WER, we have measured the WER for the four largest and the four shortest phrases. The obtained results are shown in table VI.

We have observed that there is higher WER in short phrases than in long phrases. We can also see as the WER result improves significantly in long phrases respect the reference value. However, in short phrases, this behavior does not happen.

TABLE VI. INSTANTANEOUS AND AVERAGE WER FOR THE 4 LARGEST AND THE 4 SHORTEST PHRASES.

N of words / Characteristic	4	4	5	6	Average	26	26	27	30	Average
Refernce value	0.25	0.5	0.2	0.33	**0.32**	0.46	0	0.4	0.1	**0.24**
Num. words 0	0.5	0.5	0.2	0.66	**0.465**	0.34	0	0.4	0.1	**0.21**
SAPI confidence 2	0.5	0.25	0.4	0.33	**0.37**	0.03	0.42	0.4	0.13	**0.245**
Engine confidence 1	0.25	0.75	0.2	0.33	**0.382**	0.03	0.34	0.4	0.06	**0.207**
N-Gram word 4	0.5	0.75	0.4	0.66	**0.577**	0.46	0.03	0.4	0.1	**0.247**
N-Gram Phrase 5	0.25	0.5	0.2	0.33	**0.32**	0	0.26	0.4	0.06	**0.18**

Levenshtein distance 3	0.25	0.5	0.2	0.33	**0.32**	0.42	0.03	0.4	0.13	**0.245**
Formula 6	0.25	0.75	0.2	0.33	**0.382**	0.03	0.26	0.4	0.06	**0.187**
All characteristics	0.25	0.5	0.2	0.33	**0.32**	0	0.26	0.4	0.06	**0.18**
All characteristics with weights	0.25	0.5	0.2	0.33	**0.32**	0	0.26	0.4	0.06	**0.18**

Observing the results shown in table IV, we may think that short sentences affect more to the WER than long sentences because there are fewer words. But the point is that although the sentences are longer, the number of wrongly recognized words in the longer case is lower, so the system fails less with long sentences.

9 Conclusions and Future Work

A real-time statistical speech translation system voice recognition optimization using multimodal sources of knowledge and characteristics vectors has been presented. We have tested it in pedagogical environments. The main innovation and contribution of this work is the way in which additional sources of knowledge are used to improve the accuracy of the system, thus having a notable improvement compared to existing systems. Moreover, any new proposal has been tackled from an analytical perspective, while remaining a practical work.

Training the system with other sources of information, which are also related to the class topic, also helps the system considerably. They do not need to be exactly the notes of the slides of the lecturer; they can even be texts about the same concepts developed in the classroom, such as books referenced in the class. Generally, these texts and books are often available in different languages, and training the system with such pre-existing material also improves the system.

The way we have used to improve the recognition is by means of characteristics vectors. Our results confirm that the SAPI-based speech recognizer using characteristics vectors improves the recognition results. There are two characteristics that improve the baseline considerably when they are used individually: the one that evaluates the number of words of each hypothesis respect to the rest hypothesis, and the probability at the phrase level given a language model of each hypothesis. We also think we can improve these results a little bit more by eliminating some characteristics vectors that did not give good results.

Taking into account Figure 4, we conclude that, in general, the recognition results are good and very similar in long phrases. Thus, we will continue working on improving the speech recognition engine using short phrases, but we must take into account that the worst WER results are almost always obtained in short phrases.

Another interesting contribution of our work has been the developed tool. It allows a collaborative relationship with the user. The application feeds a priori the recognition engine by using the teacher's slides and notes. Thus, the user takes an important role improving the recognition. The more implication from the user, the better is the recognition result. It allows working with a highly efficient tool for improving the recognition system.

The evaluated characteristics have been the Levenshtein distance, the probability at phrase level based on a given n-gram language model, confidence values given by recognition engine and the Microsoft Windows speech application programming interface (SAPI), etc. The optimal value obtained of the sum of these characteristics will determine a posteriori the speech recognition engine output.

The experiments demonstrate how this usage of additional sources of information really improves significantly the overall results by a 12%. Especially, when we make use of lecture notes in both languages (previous to the real-time operation), the accuracy rate increases by a 35%.

We are going to extend this work in future by performing next studies:

- We are going to analyze how varying β of the joint probability affects to the recognition result.
- We are going to find the optimal values of the weights that affect to the characteristics, as well as to select those that positively affect the result and delete the others.
- We are going to check the number of N-bestlist hypothesis that optimizes recognition results.
- We are going to use new corpuses related with the topic of the subject (apart of the used corpus: Europarl) in order to improve the recognition result.

Moreover, in a future work we will improve the system by using fuzzy network models as interfaces between the automatic speech recognition and machine translation modules. New heuristic elements will also be added to the characteristics vectors of the translator in order to study more enhancements.

Finally, we believe that our proposal will be quite more commercial if the output of the system is voice, thus we will have a system where the speaker can talk in one language and the system will send the output translated to other language through the speakers.

ACKNOWLEDGMENT

This work has been partially supported by the Generalitat Valenciana and the Universidad Politécnica de Valencia.

REFERENCES

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

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

[3] J.C. Amengual, A. Castaño, A. Castellanos, V.M. Jiménez, D. Llorens, A. Marzal, F. Prat, J.M. Vilar, J.M. Benedi, F. Casacuberta, M. Pastor, E. Vidal, The EuTrans Spoken Language Translation System, Machine Translation. 15 (2000) 75-103.

[4] A. Lavie, C. Langley, A. Waibel, F. Pianesi, G. Lazzari, P. Coletti, L. Taddei, F. Calducci, Architecture and Design Considerations in NESPOLE!: a Speech Translation System for E-Commerce Applications, In Proc. of the HLT2001, San Diego, CA. ACM, 2001.

[5] W. Wolfgang, Verbmobil: Foundations of speech-to-speech translations, Springer Verlag, Berlin, 2000.

[6] S. Nakamura, A. Lavie, The ATR multi-lingual speech-tospeech translation system, IEEE Transactions on Speech and Audio Processing, Vol. 14, No. 2, 2006, pp. 365–376.

[7] L. E. Baum, T. Petrie, Statistical inference for probabilistic functions of finite state Markov chains, Ann.Math.Stat. 37 (1966) 1554-1563.

[8] S.M. Katz, Estimation of probabilities from sparse data for the language model component of a speech recognizer, IEEE Trans. on Acoustics, Speech and Signal Processing. 35 (1987) 400-401.

[9] P.F. Brown, S.A. Della Pietra, V.J. Della Pietra, R.L. Mercer, The mathematics of statistical machine translation: Parameter estimation, Computational Linguistics. 19(2) (1993)263–310.

[10] F.J. Och, H. Ney, The alignment template approach to statistical machine translation. Computational Linguistics, 30(4) (2004) 417–450.

[11] J. Tomas, F. Casacuberta, Monotone statistical translation using word groups. In Proceedings of the Machine Translation Summit VIII, Santiago de Compostela, 2001, pp. 357–361.

[12] R. Zens, F.J. Och, H. Ney. Phrase-based statistical machine translation. Advances in artificial intelligence, 25 (2002) 18-32.

[13] F. Koehn, F.J. Och, D. Marcu, Statistical phrase-based translation, in Proceedings of the HLT-NAACL-03, Edmonton, Alberta, , 2003, pp. 127-133.

[14] F. Casacuberta, D. Llorens, C. Martınez, S. Molau, F. Nevado, H. Ney, M. Pastor, D. Picó, A. Sanchis, E. Vidal, J.M. Vilar, Speech-to-speech Translation Based on Finite-State Transducers, Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing, Salt Lake City, UH, 2001, pp. 613–616.

[15] M. Dymetman, J. Brousseau, G. Foster, P. Isabelle, Y. Normandin, P. Plamondon, Towards an Automatic Dictation System for Translators: the TransTalk Project, Proc. of ICSLP-94, Yokohama, Japan, 1994, pp. 193-196,

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

[17] J. Tomas, J. Vilar, F. Casacuberta, The ITI statistical machine translation system, in: Proceedings of the TC-Star Speech to Speech Translation Workshop, Barcelona, Spain, 2006, pp. 49–55.

[18] F.J. Och, H. Ney, Statistical Multi-Source Translation. Proc. of Machine Translation Summit VIII, Santiago de Compostela. Spain, 2001, pp. 253-258.

[19] F.J. Och, Minimum Error Rate Training for Statistical Machine Translation. Proc. of the ACL-03, Japan, Sapporo, 2003, pp. 160-167.

[20] R. Zhang, G. Kikui, H. Yamamoto, T. Watanabe, F. Soong, W.K. Lo, A Unied Approach in Speech-to-Speech Translation: Integrating Features of Speech Recognition and Machine Translation, in Proceedings of COLING, Geneve, Switzerland, 2004.

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

[22] N. Ueffing, H. Ney, Word-level confidence estimation for machine translation, Computational Linguistics. 33 (2007) 9-40.

[23] J. Blatz, E. Fitzgerald, G. Foster, S. Gandrabur, C. Goutte, A. Kulesza, A. Sanchis, N. Ueffing, Confidence estimation for machine translation, in: Proceedings of the 20th International Conference on Computational Linguistics, Geneva, 2004, pp. 315-321.

[24] A. Sanchis, A. Juan, E. Vidal, Estimation of confidence measures for machine translation, in: Proceedings of the Machine Translation Summit XI, 2007, pp. 407-412.

[25] F. Casacuberta, E. Vidal, Reconocimiento Automático del Habla, Marcombo, Barcelona, 1987.

[26] O. Duda, E. Hart, G. Stork, Pattern Classification, 2nd ed., Wiley Interscience, 2000.

[27] J. Amengual, J. Benedı, F. Casacuberta, M. Castao, A. Castellanos, V. Jimenez, D. Llorens, A. Marzal, M. Pastor, F. Prat, E. Vidal, J. Vilar, The EuTrans-I speech translation system, Machine Translation 1, 2000.

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

[29] J. Tomás, J. Lloret, F. Casacuberta, 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, 605–613M.

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

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

[32] H. Ney, S. Nießen, F. Och, H. Sawaf, C. Tillmann, S. Vogel, Algorithms for statistical translation of spoken language, IEEE Transactions on Speech and Audio Processing. 8(1) (2000) 24–36.

[33] F.J. Och, C. Tillmann, H. Ney, Improved alignment models for statistical machine translation, in Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP99), University of Maryland, College Park, MD, USA, 1999, pp. 20–28.

[34] Microsoft Speech API 5.3 (ISpeechPhraseRule) Website. At: http://msdn.microsoft.com/en-us/library/ms721679%28v=VS.85%29.aspx

[35] Microsoft Speech API 5.3 (SR Engine Vendor Porting Guide) Website. At: http://msdn.microsoft.com/en-us/library/ms717034%28v=VS.85%29.aspx#_Toc503606917 (Last Access October 30, 2011)

[36] R. San-Segundo, J. Macías-Guarasa, J.M. Montero, J. Ferreiros, R. Córdoba, J.M. Pardo, Medidas de confianza en sistemas de diálogo. Procesamiento del Lenguaje Natural, Nº 33, September 2004. At: http://www.sepln.org/revistaSEPLN/revista/33/33-Pag95.pdf

[37] R. A. Wagner, M. J. Fisher, The string to string correction problem, J. Assoc. Comput. 21 (1974) 168-173.

[38] J. Tomas, J. Lloret, F. Casacuberta, Phrasebased alignment models for statistical machine translation, in Iberian Conference on Pattern Recognition and Image Analysis, volume 3523 of Lecture Notes in Computer Science, Springer-Verlag, Estoril (Portugal), 2005, pages 605–613.

[39] D. Marcu, W. Wong, Joint probability model for statistical machine translation, In Proceedings of the Conference on Empirical Methods in Natural Language Processing and Very Large Corpora EMNLP-02, 2002.

[40] R. Zens, F.J. Och, H. Ney, Phrase-based statistical machine translation, in G. Lakemeyer M. Jarke, J. Koehler, editor, Advances in artificial intelligence. 25. Annual German Conference on AI, KI 2002, volume 2479 of *LNAI*, Springer Verlag, September, 2002, pp. 18–32.

[41] Philipp Koehn, Franz Josef Och, Daniel Marcu, Statistical phrase-based translation, in Proceedings of the 2003 Meeting of the North American chapter of the Association for Computational Linguistics (NAACL-03), Edmonton, Alberta, 2003.

[42] F.J. Och, Statistical Machine Translation: From Single-Word Models to Alignment Templates, Ph.D. thesis, RWTH Aachen, Aachen, Germany, October 2002.

[43] F.J. Och, H. Ney, A systematic comparison of various statistical alignment models, Computational Linguistics. 29(1) (2003) 19–51.

[44] F.J. Och, Minimum error rate training in statistical machine translation, in Proc. of the Association for Computational Linguistics, Sapporo, Japan, July 6-7 2003.

[45] J.A. Nedler, R. Mead, A simplex method forfunction minimization, Computer Journal. 7 (1965) 308–313.

[46] A.L. Berger, P.F. Brown, S.A. Della Pietra, V.J. Della Pietra, J.R. Gillett, A.S. Kehler, R.L. Mercer, Language translation apparatus and method of using context-based translation models, United States Patent, No. 5510981, Apr. 1996a.

[47] J. Tomas, F. Casacuberta, Statistical machine translation decoding using target word reordering, in Structural, Syntactic, and Statistical Pattern Recongnition, volume 3138 of Lecture Notes in Computer Science, Springer-Verlag, 2004, pp. 734–743.

[48] S. Vogel, Y. Zhang, F. Huang, A. Venugopal, B. Zhao, A. Tribble, M. Eck, A. Waibel, The CMU statistical machine translation system, in Proceedings of the Machine Translation Summit IX, September 2003, pp 110–117.

[49] P. Koehn, Pharaoh: a beam search decoder for phrase-based statistical machine translation models, in Proceedings of the The 6th Conference of the Association for Machine Translation in the Americas (AMTA04), volume 3265 of Lecture Notes in Artificial Intelligence, Springer, Georgetown University, Washington DC, USA, September-October 2004, pages 115–124.

[50] Y.-Y. Wang, A. Waibel, Decoding algorithm in statistical machine translation, in Proceedings of the 35th. Annual Meeting of the Association on Computational Linguistics, Madrid, Spain, 1997.

[51] P. Koehn, Europarl: A parallel corpus for statistical machine translation, in: MT Summit 2005, Phuket, Thailand, 2005.