iATROS: A SPEECH AND HANDWRITING RECOGNITION SYSTEM

Míriam Luján-Mares, Vicent Tamarit, Vicent Alabau,
Carlos-D. Martínez-Hinarejos, Moisés Pastor, Alberto Sanchis, Alejandro Toselli

Instituto Tecnológico de Informática, Universidad Politécnica de Valencia,
Camino de Vera, s/n, 46022, Valencia, Spain

ABSTRACT

Speech technologies have developed in the last twenty years and now allow the implementation of real-world speech applications. Furthermore, handwriting recognition has gained attention in the last years due to their multiple applications and the opportunity of reusing the consolidated speech technology for that problem. In this work we present the implementation of the modules of a flexible recognition system, the iATROS system, which allows speech and handwriting input. The iATROS system is developed in a modular manner, with a core recognition engine and several utility functions that can be used in the construction of speech and handwriting-based applications, including multimodal and interactive applications. We show the capabilities and features of the modules and present a few schemes on how the modules can be used to build applications.

1. INTRODUCTION

In the last twenty years, the speech recognition systems have become widely available to research scientists and nowadays they are quite present in real world. Some free speech recognizers based on Hidden Markov Models (HMM), like Sphinx [1] or HTK [2], are available for the speech processing research community, which uses and modifies them to experiment with different techniques to enhance the speech recognition performance. These recognizers can be used on the construction of speech-based applications, but with some limitations due to the difficulty of integration with other software applications and possible license restrictions.

Parallel to the speech recognition development, text recognition has gained interest in the last years for its applications: automatic processing of forms [3], handwriting transcription [4], transcription of ancient books [5], etc. A few years ago, handwriting text recognition started to base on the same technology as speech recognizers (HMM-based). Therefore, many speech recognizers have been adapted by the handwriting text recognition researchers to cope with this new task.

In this work we present a new recognizer which allows the recognition of both speech and handwriting signals, the iATROS[1] recognizer. iATROS is composed of two preprocessing and feature extraction modules (for speech signal and handwriting images) and a core recognition module. The preprocessing and feature extraction modules provide feature vectors to the recognition module, that using HMM models and language models performs the search for the best recognition hypothesis. All the modules are implemented in C.

Since the iATROS system accepts both speech and handwriting signal, it is possible to build multimodal applications based on this system. The flexibility of the core recognition module allows the implementation of many applications based on this system.

The paper has the following content: Section 2 presents the speech preprocessing and feature extraction; Section 3 presents the handwriting images preprocessing and feature extraction; Section 4 describes the basic recognition process for the core recognizer; Section 5 describes a few examples on how to develop applications based on the iATROS system; Section 6 presents some concluding remarks and future plans to improve and use the iATROS system.

2. SPEECH PROCESSING

The iATROS sound system is based on the ALSA[2] sound modules. The software package includes record software for both online and offline recognition. The source code includes functions to load, save and play sounds. The software can read three different sound formats: raw data, AD files (defined by the PRHLT group) and WAV files without compression. The output for a recorded sound is only raw data, because we think is the most compatible format.

iATROS includes a feature extraction program based on the mel cepstral coeficients [6]. The architecture of the preprocess module is quite simple and reproduces the

WORK PARTIALLY SUPPORTED BY THE SPANISH RESEARCH PROGRAMME CONSOLIDER INGENIO 2010: MIPRCV (CSD2007-00018), BY SPANISH MEC AND FEDER UNDER PROJECT TIN2006-15694-C02-01, BY THE GENERALITAT VALENCIANA UNDER GRANT GVPRE/2008/331 RESEARCH PROJECT "UPENNSPANISH" AND BY VIDI-UPV UNDER GRANT FPI-PAID06 AND PROJECT 20070315.

[1] iATROS stands for improved Automatically Trainable Recognizer of Speech.

[2] Advanced Linux Sound Architecture

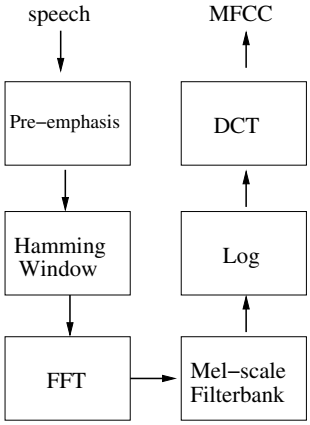

Figure 1. Process diagram for each frame of audio signal.

typical extraction process used in speech recognition. The audio signal is processed by moving a window over it; this portion of signal covered by the window is called *frame*. For each frame we compute its cepstrum coefficients using the modules shown in Figure 1. These modules are separate functions, with no dependencies between them except the input and output data. This modularization allows to easily modify the feature extraction process by changing the modules or adding new ones.

The functions use a structure which stores all the information needed in the process. These parameters are initially loaded from a configuration file. The values that affect the feature extraction and can be modified by the user are: size of the preprocess window, sample frequency, audio channels, number of coding bits, subsample frequency, length of the FFT, pre-emphasis factor, number of cepstrals, silence threshold, and duration of silence.

One of the most important parts of a feature extraction system is the Fast Fourier Transform. We used the FFTW3 library [7]. This library is free software and is one of the most efficient ways to compute the FFT. Another important piece in the preprocess pipe is the estimation of the Mel Filter Bank. Software like Sphinx [1] computes the filters in real time, but we decided to do that work offline because the automatical estimation is complicated (manual tuning is usually requiered). Moreover, it is not necessary to compute the filters for each run, since the filters only depend on the sample frequency.

The feature vector is formed by the cepstrum coefficients and an extra element, the frame energy. This value is a global measure for the frame and is computed as the first element of the Discrete Cosine Transform. The output of the feature extraction module is in plain text format with a header that indicates the number of cepstrum features and vectors, as well as other parameters.

3. HANDWRITING PROCESSING

The process starts from a PGM image. The following steps take place in the text preprocessing module. First, a conventional noise reduction method, and skew correction are applied on the whole document image. Its output is then fed to the text line extraction process, which divides it into separate text lines images. Finally, slope and slant correction, and size normalization are applied on each of these separate lines. More detailed description of this preprocessing can be found in [8, 9].

As our recognition system is based on HMM, each preprocessed line image is represented as a sequence of feature vectors. To do this, the feature extraction module applies a grid to divide line image into $N \times M$ squared cells ($N = 20$ is an usual value and M must satisfy the condition $M/N = \textit{original image aspect ratio}$). From each cell, three features are calculated: normalized gray level, horizontal gray level derivative and vertical gray level derivative. The image context is taken into account in this process. The way these three features are determined is described in [10]. Columns of cells (also called *frames*) are processed from left to right and a feature vector is constructed for each frame by stacking the three features computed in its constituent cells. Hence, at the end of this process, a sequence of M $(3N)$-dimensional feature vectors (N normalized gray-level components and N horizontal and N vertical derivatives components) is obtained. In Figure 2 is shown graphically an example of feature vectors sequence x.

4. DECODING PROCESS

The decoding process is performed by using the Viterbi algorithm. In this process, the most likely sentence is searched in a network that integrates the morphological (HMM), lexical and syntactic models. The network is composed of states. In each state, three types of transitions can be distinguished, according to the model that is involved.

The states pertaining to a recognition stage are stored in a heap, whose size is determined by a configuration parameter. A hash table is used to allow an efficient search for the states. Each state has the following essential information on the current:

- State of the language model
- History
- State of the morphological model
- State of the lexical model

The search uses two types of pruning:

- Histogram pruning: this pruning is provided by the size of the heap that stores the current stage; when the heap is full, the probability of the new generated state p_n is compared to the probability of the state in the heap with lowest probability p_l; if $p_n \leq p_l$, the new state is not introduced into the heap; if $p_l < p_n$, the state with p_l gets lost; therefore, an implicit pruning is performed.

Figure 2. Graphical representation of the feature extraction of the text image *"antiguos ciudadanos que en castilla se llamaban"*. The first corresponds with the normalized grey level features, whereas the second and third with the horizontal and vertical derivatives features respectively.

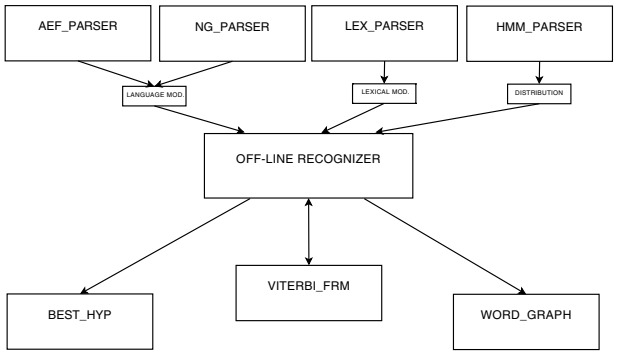

Figure 3. An example of organization of the iATROS modules.

- Beam-search: when a new state presents a probability that is lower than the probability of the current best state divided by the beam factor, the new state is not introduced into the next stage.

5. BUILDING APPLICATIONS: SOME EXAMPLES

The core of the recognizer is the main recognition function (viterbi_frame). This function receives as basic input the feature vector, the set of models (morphological, lexical and language) and the current recognition stage (heap of states), and produces as output the new stage using the search presented in Section 4. Apart from that function, iATROS provides one parser for each type of model, a function to calculate the word-graph and a function to calculate the best hypothesis (Figure 3).

The parsers have the current features:

- Language model parser: supports N-grams (in ARPA format) and Finite State Models (FSM); the language model is loaded in the same structure for both types of language models.
- Lexical model parser: supports FSM with alternative pronunciations.
- Morphological models: supports continuous density HMM in the HTK format, with gaussians as output distributions in the states; it is possible, with little effort, to modify the format, the parser and the recogniser to support HMM with other non-gaussian output distributions in the states.

The main advantage of this recognizer is that it is very easy to build new applications, since the main function of the recognizer has a stable and well defined interface, as well as the auxiliary functions. Therefore, new speech and handwritten text applications can be implemented by using the basic iATROS functions and implementing auxiliary functions that process the results provided by the iATROS functions.

Some examples of applications that can be build based on iATROS are presented in the following subsections.

5.1. Off-line recognition

To carry out off-line recognition only some steps are necessary:

- Read the configuration file.
- Load the models: morphological, lexical and syntactic models.
- For each sentence to be recognized:
  - Analyze frame to frame.
  - Optionally: obtain the word-graph.
  - Return the best hypothesis.
- Free memory and end processes.

This application is actually implemented in a small piece of code.

5.2. On-line speech recognition

For the on-line speech recognition task, the audio system must be initialized and used to feed the recognizer with frames. The on-line recognizer follows this scheme:

- Read the configuration file.
- Load the models: morphological, lexical and syntactic models.
- Init audio system.
- While user does not finish the process:
  - Wait for audio input.
  - While input cepstra are present, analyze frame to frame.
  - Return the best hypothesis.
- Free memory and end processes.

5.3. Combining handwritten text and speech recognition

This application is actually not implemented, but it is shown as an easy example of construction of a multimodal application based on iATROS. In this case, an image representing a text is presented to the user, who utters the corresponding words to the recognizer. The recognizer uses both types of inputs (multimodal input) to enhance the recognition. The scheme for this kind of application is the following:

- Read the configuration.
- Load the models: morphological (only text), acoustic (only speech), lexical and syntactic models (common).
- For each sentence to be recognised:
  - Analyze text input frame to frame.
  - Obtain word-graph for text recognition.
  - Init audio system.
  - Wait for audio input.
  - Analyze speech input frame to frame.
  - Obtain word-graph for speech recognition.
  - Process text and speech word-graphs to return the best common hypothesis.
- Free memory and end processes

6. CONCLUDING REMARKS

In this article we have introduced the basic architecture and modules than form the iATROS system. We presented the steps that are used in speech and handwritten text process to obtain the feature vectors that can be processed by the decoder. We showed the basic features of the decoder, as well as the different models and formats that can be used.

We presented some applications based on the iATROS system: two basic recognizers (on-line and off-line), whose implementation is quite simple (only short programs are requiered to manage the iATROS functions) and a multimodal recognizer that allows both handwritten text and speech. This last recognizer should be implemented as future work, but the presented scheme shows that its construction is not difficult at all.

Future work is directed to the implementation of new applications based on iATROS, as well as its use in new tasks and experiments. Some improvements on temporal complexity and the addition of new features (e.g., to allow not-gaussian output distributions) will be done at the internal level. The implementation of new applications could be used to improve the core system with new utility functions (e.g., recognition with confidence measures, speaker adaptation, etc.).

7. REFERENCES

[1] Xuedong Huang, Fileno Alleva, Hsiao-Wuen Hon, Mei-Yuh Hwang, and Ronald Rosenfeld, "The SPHINX-II speech recognition system: an overview," *Computer Speech and Language*, vol. 7, no. 2, pp. 137–148, 1993.

[2] S. Young, G. Evermann, T. Hain, D. Kershaw, G. Moore J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. Woodland, *The HTK Book for HTK V3.2*, Cambridge University Press, Cambridge, UK, 2004.

[3] A.C. Downton, A. Amiri, L. Du, and S.M. Lucas, "A configurable toolkit approach to handwritten forms recognition," in *IEE Coll. on Doc. Image Processing and Multimedia Environments*, Nov 1995.

[4] Alessandro Vinciarelli, Samy Bengio, and Horst Bunke, "Offline recognition of unconstrained handwritten texts using hmms and statistical language models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 6, pp. 709–720, 2004.

[5] V. Romero, A. H. Toselli, L. Rodríguez, and E. VidalA, "Computer Assisted Transcription for Ancient Text Images," in *International Conference on Image Analysis and Recognition (ICIAR 2007)*, vol. 4633 of *LNCS*, pp. 1182–1193. Springer-Verlag, Montreal (Canada), August 2007.

[6] L. Rabiner and B. Juang, *Fundamentals of Speech Recognition*, Prentice Hall PTR, 1993.

[7] Matteo Frigo and Steven G. Johnson, "The design and implementation of FFTW3," *Proceedings of the IEEE*, vol. 93, no. 2, pp. 216–231, 2005, special issue on "Program Generation, Optimization, and Platform Adaptation".

[8] Moisés Pastor, Alejandro Toselli, and Enrique Vidal, "Projection profile based algorithm for slant removal," in *International Conference on Image Analysis and Recognition (ICIAR'04)*, Porto, Portugal, Sept. 2004, Lecture Notes in Computer Science, pp. 183–190, Springer-Verlag.

[9] V. Romero, M. Pastor, A. H. Toselli, and E. Vidal, "Criteria for handwritten off-line text size normalization," in *Procc. of The Sixth IASTED international Conference on Visualization, Imaging, and Image Processing (VIIP 06)*, Palma de Mallorca, Spain, August 2006.

[10] A. H. Toselli, A. Juan, D. Keysers, J. González, I. Salvador, H. Ney, E. Vidal, and F. Casacuberta, "Integrated Handwriting Recognition and Interpretation using Finite-State Models," *Int. Journal of Pattern Recognition and Artificial Intelligence*, vol. 18, no. 4, pp. 519–539, June 2004.