Alabau, V.; Martínez Hinarejos, CD.; Romero Gómez, V.; Lagarda Arroyo, AL. (2014). An iterative multimodal framework for the transcription of handwritten historical documents. Pattern Recognition Letters. 35:195-203. doi:10.1016/j.patrec.2012.11.007.

An iterative multimodal framework for the transcription of handwritten historical documents

Vicent Alabau*, Carlos-D. Martínez-Hinarejos, Verónica Romero, Antonio-L. Lagarda

Institut Tecnológic d'Informatica, Universitat Politècnica de València, Camino de Vera s/n, 46022 València, Spain

Abstract

The transcription of historical documents is one of the most interesting tasks in which Handwritten Text Recognition can be applied, due to its interest in humanities research. One alternative for transcribing the ancient manuscripts is the use of speech dictation by using Automatic Speech Recognition techniques. In the two alternatives similar models (Hidden Markov Models and n-grams) and decoding processes (Viterbi decoding) are employed, which allows a possible combination of the two modalities with little difficulties. In this work, we explore the possibility of using recognition results of one modality to restrict the decoding process of the other modality, and apply this process iteratively. Results of these multimodal iterative alternatives are significantly better than the baseline uni-modal systems and better than the non-iterative alternatives.

Keywords: ancient text transcription, handwritten text recognition, speech dictation, multimodal systems, iterative systems, language modelling

1. Introduction

In the last years, many on-line archives and digital libraries are publishing large quantities of digitised legacy documents. These documents must be transcribed into an appropriate textual electronic format in order to allow text-based search of their contents and provide historians and other researchers new ways of indexing, consulting and querying their contents. However, the vast majority of these documents (hundreds of terabytes of digital image data) remain waiting to be transcribed into a textual electronic format. Therefore, manual transcription of these documents is an important task for making available the contents of digital libraries.

These transcriptions are usually carried out by experts in paleography, who are specialised in reading ancient scripts. These scripts are characterised by different handwritten/printed styles from diverse places and time periods. The time that takes for an expert to make a transcription of one of these documents depends on their skills and experience. Most paleographs agree that each page needs several hours to be transcribed.

In this context, Handwritten Text Recognition (HTR) (Toselli et al., 2004) has become an important research topic. HTR tries to obtain the word sequence contained in the image of a handwritten text line. This process needs a previous detection of lines of text in an image, as well as some preprocessing steps to make the handwritten text more regular. The final result is a sequence of words (transcription) of the text line, that may contain errors. When the rate of errors of the transcription is low enough, HTR can be a very useful tool to speed up the transcription of handwritten text documents.

However, when consulting paleographs on the most comfortable method to transcribe a handwritten text document, many of them claim that a dictation of the words is the best option. Consequently, Automatic Speech Recognition (ASR) systems are an important alternative to HTR systems. In addition, the current state-of-the-art ASR and HTR systems share many features: Hidden Markov Models (HMM) (Jelinek, 1998; Rabiner, 1989) are used to model the

*Corresponding author: Phone: (34) 96 387 70 69, Fax: (34) 96 387 72 39.

Email addresses: valabau@iti.upv.es (Vicent Alabau), cmartine@iti.upv.es (Carlos-D. Martínez-Hinarejos), vromero@iti.upv.es (Verónica Romero), alagarda@iti.upv.es (Antonio-L. Lagarda)

basic elements of the signal (sounds for speech, strokes for handwritten text) and n-grams language models (LM) are used to model word sequences (Jelinek, 1998). From this viewpoint, HTR systems fit in the Natural Language Processing paradigm. Therefore, many features that are usual to ASR systems (such as the use of training data for HMM and n-grams) are common to HTR systems as well.

The similarities between the two types of systems make possible to combine them easily into a multimodal system that may obtain a more reliable final hypothesis, since two different data sources (handwritten text and speech) can be used. In fact, previous attemps in combining handwritten input and speech input have been done (Liu and Soong, 2009), but most of them focus on the use of on-line handwritten text. In a previous work (Alabau et al., 2011) a first attempt of combining off-line HTR and ASR systems showed promising results. The method consisted basically of restricting the ASR decoding process based on the results of the HTR decoding. In this work we extend the process described in Alabau et al. (2011) towards two different directions: we explore the effect of using the different modalities (HTR and ASR) as starting modality, and we study the iterative use of the process.

The paper is organised as follows: Section 2 describes the fundamentals of HTR and ASR systems, Section 3 explains the use of the HTR decoding to improve the ASR recognition, Section 4 summarises the experimental set-up, Section 5 shows the results, and Section 6 provides the main conclusions and future work lines in this field.

2. Systems overview

Up-to-date systems for both HTR (Toselli et al., 2004) and ASR (Rabiner, 1989) share the same consolidated technology based on Hidden Markov Models (HMMs) (Jelinek, 1998). The most important differences lay in the type of input sequences of feature vectors: while in the case of off-line HTR are line-image features, the input sequences for ASR represent acoustic data. Figure 1 shows an example of how a HMM models two feature vector subsequences pertaining to the character "a" and the phoneme "a".

The problem of both handwriting and speech recognition can be formulated as the problem of finding the most likely word sequence, $\boldsymbol{w} = (w_1, w_2, \ldots, w_{|\boldsymbol{w}|})$, for a feature vector sequence $\boldsymbol{x} = (x_1, x_2, \ldots, x_{|\boldsymbol{x}|})$ describing a text image or speech signal along its corresponding horizontal or time axis i.e., $\boldsymbol{w} = \operatorname{argmax}_{\boldsymbol{w}} P(\boldsymbol{w}|\boldsymbol{x})$. Using the Bayes' rule we can decompose this probability into two probabilities, $P(\boldsymbol{x}|\boldsymbol{w})$ and $P(\boldsymbol{w})$, representing morphological or acoustical knowledge, and syntactic knowledge, respectively:

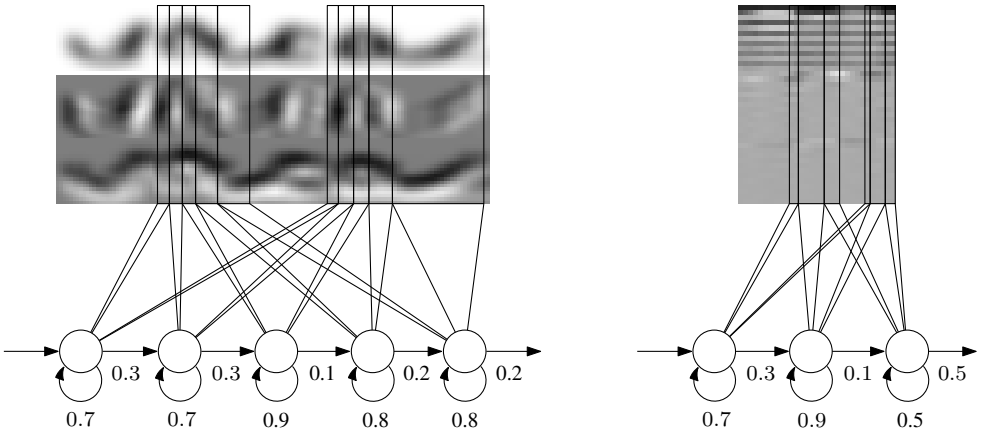
$$\hat{\boldsymbol{w}} = \operatorname*{argmax}_{\boldsymbol{w}} P(\boldsymbol{w}|\boldsymbol{x}) = \operatorname*{argmax}_{\boldsymbol{w}} P(\boldsymbol{x}|\boldsymbol{w}) P(\boldsymbol{w}) \tag{1}$$

$P(\boldsymbol{x}|\boldsymbol{w})$ is typically approximated by concatenated character/phoneme models, usually HMMs, and $P(\boldsymbol{w})$ is approximated by a word LM, usually n-grams (Jelinek, 1998).

Each character/phoneme is modelled by a continuous density left-to-right HMM with a Gaussian mixture per state. This mixture serves as a probabilistic law to the emission of feature vectors on each model state. The optimum number of HMM states and Gaussian densities per state are tuned empirically. Each lexical word is usually modelled by a stochastic finite-state automaton (SFSA), which represents all possible concatenations of individual character/phonemes to compose the word. By embedding the character/phoneme HMMs into the edges of this automaton, a lexical HMM is obtained. The model parameters can be easily trained from samples (handwritten text image or speech utterance) accompanied by the transcription of these samples into the corresponding sequence of characters/phonemes. This training process is carried out by using a well known instance of the EM algorithm called forward backward or Baum-Welch. On the other hand, text lines or sentences are modelled using smoothed word n-grams, estimated from the training transcriptions of the text images.

Once all the character/phoneme, word and language models are available, recognition of new test sentences can be performed. Thanks to the homogeneous finite-state nature of all these models, they can be easily integrated into a single global model on which a search process is performed for decoding the input feature vectors sequence into an output word graph. This search is efficiently carried out by using the Viterbi algorithm.

The two implemented systems (HTR and ASR) are presented in detail in Section 2.1 and Section 2.2. Section 2.3 defines the concept of word graph (or lattice) which is the basis for combining the two modalities.



(a) HMM-HTR (b) HMM-ASR

Figure 1: Example of 5-states HMM for HTR (left) and 3-states HMM for ASR (right) modelling (sequences of feature vectors) instances of the character "a" and the phoneme "a", respectively, within the Spanish word "saca". The states are shared among all instances of characters/phonemes of the same class.

2.1. Handwritten Text Recognition

The HTR system used here follows the classical architecture composed of three main modules: document image preprocessing, line image feature extraction and HMM training/decoding (Toselli et al., 2004).

The following steps take place in the preprocessing module. First, the skew of each page is corrected; we understand "skew" as the angle between the horizontal direction and the direction of the lines on which the writer aligned the words. Then, a conventional noise reduction method is applied on the whole document image, whose output is then fed to the text line extraction process which divides it into separate text lines images. Finally, slant correction and size normalisation are applied on each separated line. A more detailed description of the preprocess can be found in Toselli et al. (2004) and Romero et al. (2007).

As our HTR system is based on HMMs, each preprocessed line image is represented as a sequence of feature vectors. To do this, the feature extraction module applies a grid to divide the text line image into $N \times M$ squared cells. From each cell, three features are calculated: normalised gray level, horizontal gray level derivative and vertical gray level derivative. The way these three features are determined is described in Toselli et al. (2004). Columns of cells or *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 is obtained. In Figure 1 (left) an example of the sequence of feature vectors for the word "saca" is shown graphically.

2.2. Automatic Speech Recognition

The feature extraction of the ASR system is based on the Mel cepstral coefficients (Rabiner, 1989). Speech preprocessing reproduces the standard steps for speech recognition. The audio signal is captured from a microphone at 16kHz and digitalised. A sliding window with overlapping is passed over the signal. For each window the following procedure is carried out. First, in the pre-emphasis step, a high-pass filter is used to compensate the differences between high and low frequencies. Second, a Hamming window is applied to smooth out the borders of the window. Next, the signal is converted from the time domain to the frequency domain by means of a discrete Fourier transform. To mimic the mechanism of the human ear, the Mel scale is used to group the energy of frequencies that are indistinguishable to humans. Then, volume normalisation is carried out by applying a logarithmic transformation. Now, a discrete cosine transform is performed, resulting in the so-called cepstral coefficients. The frame energy is added as an extra element. This value is a global measure for the frame and it is computed as the first element of the discrete cosine transform. Finally, first and second derivatives are added to the final feature vector.

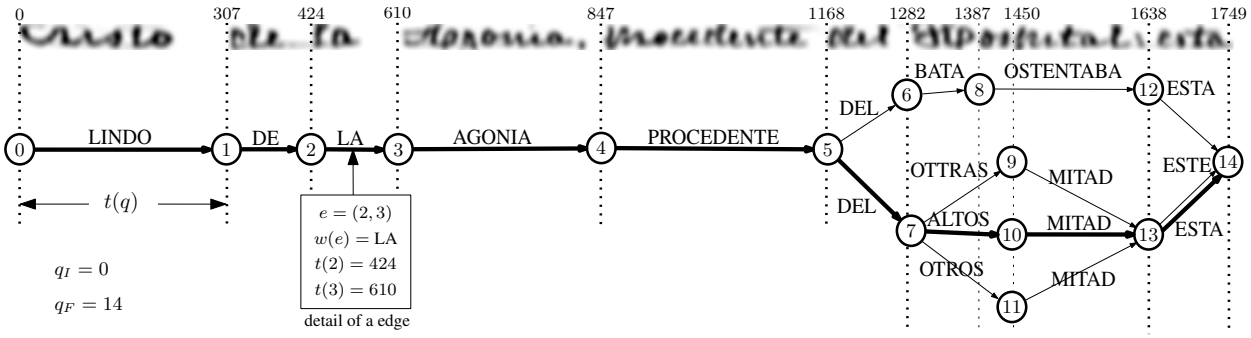


Figure 2: Graphical representation of a word graph. In the upper side, the preprocessed image for the text 'CRISTO DE LA AGONIA PROCEDENTE DEL HOSPITAL ESTA' is shown with the numbers of the relevant frame indices. Vertical dotted lines align each of the nodes with its corresponding index in the image.

2.3. Word Graph

A word graph (WG) is a data structure that represents very efficiently a large amount of sequences of words. WGs are useful for characterising a subset of the most likely solutions from the hypothesis space. First, they can encode hypotheses in a much more compact way than traditional n-grams. Second, there exists a reasonable collection of well-defined and well-known efficient algorithms for them (Vidal et al., 2005a,b). The hypotheses encoded in the WG are those whose posterior probability is large enough, according to the morphological/acoustic likelihood and language models used to decode the input signal, represented as a sequence of feature vectors (Toselli et al., 2011; Liu and Soong, 2006; Ortmanns et al., 1997a). In the case of HMM modelling, a WG can be seen just as a pruned version of the Viterbi search trellis that is obtained when transcribing a whole image line or speech signal.

Formally, a word graph can be represented as a weighted directed acyclic graph (WDAG). A WDAG is defined by the eight-tuple $G = (Q, t, q_I, q_F, E, V, P, \omega)$. The set of nodes ($Q = \{0, 1, 2, ..., |Q| - 1\}$) is ordered following a topological order and each node is labelled with its corresponding index in this order. In addition, each node, q, is associated with an index of the input vector sequence, $t(q)$. The WG has a start node $q_I \in Q$ and an end node $q_F \in Q$. On the other hand, each edge $e \in E$ is denoted by its start and end nodes $e = (u, v) : u \in Q \wedge v \in Q \wedge u < v$. The edges are related with a word, $\omega(e)$, from the task vocabulary V and with a probability function $P(e|x_{t(u)+1}^{t(v)})$ that represents the probability of the hypothesis $\omega(e)$ appearing between $t(u) + 1$ and $t(v)$. An example of WG for HTR is shown in Figure 2.

A path, $\phi = \{e_1 = (q_I, q_1), e_2 = (q_1, q_2), \ldots, e_l = (q_l, q_F)\}$, is a sequence of connected edges from q_I to q_F whose words, $\boldsymbol{w} = \{\omega(e_1), \omega(e_2), \ldots, \omega(e_l)\}$, form a complete hypothesis. The probability of a path can be decomposed using a naive Bayes approach. Thus, it can be computed as the product of the probabilities of the edges along the path:

$$P(\phi|\boldsymbol{x}) = \prod_{k=1}^{l} P(e_k|x_{t(u)+1}^{t(v)}) \tag{2}$$

Given that WGs are ambiguous (for each node and word pair there may be several possible next nodes), in general there is more than one path associated with the sequence $\boldsymbol{w}$. Let $d(\boldsymbol{w})$ be the set of all the paths associated with $\boldsymbol{w}$ and $\phi_{\boldsymbol{w}}$ one of these paths. The probability of the word sequence $\boldsymbol{w}$ is computed as:

$$P(\boldsymbol{w}|\boldsymbol{x}) = \sum_{\phi_{\boldsymbol{w}} \in d(\boldsymbol{w})} P(\phi_{\boldsymbol{w}}|\boldsymbol{x}) \tag{3}$$

Then, the word sequence with greatest probability can be written as:

$$\hat{\boldsymbol{w}} = \underset{\boldsymbol{w}}{\text{argmax}} \sum_{\phi_{\boldsymbol{w}} \in d(\boldsymbol{w})} P(\phi_{\boldsymbol{w}}|\boldsymbol{x}) \tag{4}$$

However, this maximization problem is NP-hard (Casacuberta and Higuera, 2000). Therefore, an adequate solutions is to approximate the sum by its dominant addend:

$$\hat{\boldsymbol{w}} \approx \underset{\boldsymbol{w}}{\operatorname{argmax}} \max_{\phi_{\boldsymbol{w}} \in d(\boldsymbol{w})} P(\phi_{\boldsymbol{w}}|\boldsymbol{x}) \tag{5}$$

which can be solved by means of efficient search algorithms (Jelinek, 1998).

In handwritten or speech recognition, the probability of an edge, $P(e|x_{t(u)+1}^{t(v)})$, where $e = (u, v)$, is the product of the morphological/acoustic probability of the lexical elements between its start and end node positions, $x_{t(u)+1}^{t(v)}$, times the language model probability of the given element (word) at the edge, $\omega(e)$. That is,

$$P(e|x_{t(u)+1}^{t(v)}) = P(\omega(e))^{\beta} \cdot P(x_{t(u)+1}^{t(v)} \mid \omega(e)) \tag{6}$$

where β is used to scale the language model probability. Since there is a difference in the range of both probabilities, the language model probability needs to be scaled to match the dynamic range of the morphological/acoustic probability.

3. Iterative combination of HTR and ASR

3.1. Combining HTR and ASR

Basically, the problem consists in obtaining a sequence of words $\boldsymbol{w}$ which is, at the same time, a transcription of the handwritten text $\boldsymbol{x}$ (from the HTR problem) and a speech utterance $\boldsymbol{s} = (s_1, s_2, \ldots, s_{|\boldsymbol{s}|})$. Statistically, this problem can be formulated as

$$\hat{\boldsymbol{w}} = \underset{\boldsymbol{w}}{\operatorname{argmax}}\, P(\boldsymbol{w}|\boldsymbol{s}, \boldsymbol{x}) = \underset{\boldsymbol{w}}{\operatorname{argmax}}\, P(\boldsymbol{s}|\boldsymbol{x}, \boldsymbol{w}) P(\boldsymbol{w}|\boldsymbol{x}) \tag{7}$$

Making the safe assumption that $P(\boldsymbol{s}|\boldsymbol{x}, \boldsymbol{w})$ is independent of $\boldsymbol{x}$, Eq. (7) can be rewritten as

$$\hat{\boldsymbol{w}} \approx \underset{\boldsymbol{w}}{\operatorname{argmax}}\, P(\boldsymbol{s}|\boldsymbol{w}) P(\boldsymbol{w}|\boldsymbol{x}) \tag{8}$$

where $P(\boldsymbol{s}|\boldsymbol{w})$ is a conventional acoustic HMM for ASR, and $P(\boldsymbol{w}|\boldsymbol{x})$ is a LM conditioned on the handwritten text image, $\boldsymbol{x}$. Note that if $\boldsymbol{x}$ is dropped, the LM can be approximated by a standard n-gram LM, $P_N(\boldsymbol{w})$. In that case, Eq. (8) can be decoded with a state-of-the-art ASR system. However, a more interesting approach would be to take advantage of the information given by $\boldsymbol{x}$.

Although in principle an integrated decoding could be possible (Bengio, 2004), it would require a specific training and decoding. This is especially complicated since both input signals have different lengths and they are not synchronised. A possible alternative is based on a semi-coupled approximation, in which Eq. (1) can be transformed, after a HTR decoding, into a statistical LM that can be used with current ASR systems.

In Alabau et al. (2011) we presented a procedure to perform such transformation. However, only preliminary experiments were carried out. The procedure consists on estimating an n-gram based language model from the posterior probabilities of an HTR WG. First, the WG can be obtained from the HTR system as explained in (Ortmanns et al., 1997b).

Then, the posterior probabilities for each node and edge must be computed. These probabilities are based on the forward and backward probabilities of the nodes. The forward probability $\alpha(u)$ can be defined as the sum of the probability of all the prefix paths in the WG reaching the node u from q_I. Correspondingly, the backward probability $\beta(u)$ can be defined as probability of all the suffix paths in the WG reaching q_F from u. These probabilities can be efficiently computed with the well-known *forward-backward* algorithm (Wessel et al., 2001).

The posterior probability for a specific edge e can be computed by summing up the posterior probabilities of all hypotheses of the WG containing it:

$$P(e \mid \boldsymbol{x}) = \frac{\alpha(u) \cdot P(e|x_{t(u)+1}^{t(v)}) \cdot \beta(v)}{\alpha(q_F)} \tag{9}$$

Similarly, the posterior probability for a specific node i is

$$P(u \mid \boldsymbol{x}) = \frac{\alpha(u) \cdot \beta(u)}{\alpha(q_F)} \tag{10}$$

Then, the expected count for a word sequence $w_{i-n+1}^{i} = (w_{i-n+1}, \ldots, w_i)$ can be estimated efficiently as in Campbell and Richardson (2008):

$$C^*(w_{i-n+1}^{i} \mid \boldsymbol{x}) = \sum_{l_1^n \in \mathcal{N}(w_{i-n+1}^{i})} \frac{\prod_k P(l_k|\boldsymbol{x})}{\prod_k P(u_k|\boldsymbol{x})} \tag{11}$$

where $\mathcal{N}(w_{i-n+1}^{i})$ are all the sequences of concatenated edges in the WG generating w_{i-n+1}^{i}.

Now, n-gram posterior probabilities can be calculated after a proper normalisation:

$$P_L(w_i|w_{i-n+1}^{i-1}, \boldsymbol{x}) = \frac{C^*(w_{i-n+1}^{i} \mid \boldsymbol{x})}{C^*(w_{i-n+1}^{i-1} \mid \boldsymbol{x})} \tag{12}$$

This simple estimation presents two problems: no back-off is included, and only words present in the WG are included into the model (which implies a high number of out-of-vocabulary words, since WG only contain the words of the most likely hypotheses). The estimation of the back-off probabilities for the n-gram is obtained by applying a suitable discount method before normalisation. The out-of-vocabulary (OOV) problem is solved by equally distributing among the OOV words the discounted probability mass of the 1-gram.

The resulting LM can be defined as

$$P_L(\boldsymbol{w}|\boldsymbol{x}) = \prod_i P_L(w_i|w_{i-n+1}^{i-1}, \boldsymbol{x}) \tag{13}$$

Finally, in order to avoid poor estimations of $P_L(\boldsymbol{w}|\boldsymbol{x})$, a linear interpolation with the original LM probability $P_N(\boldsymbol{w})$ can be used as well

$$P_\gamma(\boldsymbol{w}|\boldsymbol{x}) = \gamma P_L(\boldsymbol{w}|\boldsymbol{x}) + (1-\gamma) P_N(\boldsymbol{w}) \tag{14}$$

Note that Eq. (7) could have been decomposed, as well, in the following way,

$$\hat{\boldsymbol{w}} = \operatorname*{argmax}_{\boldsymbol{w}} P(\boldsymbol{w}|\boldsymbol{x}, \boldsymbol{s}) = \operatorname*{argmax}_{\boldsymbol{w}} P(\boldsymbol{x}|\boldsymbol{s}, \boldsymbol{w}) P(\boldsymbol{w}|\boldsymbol{s}) \tag{15}$$

Obviously, the estimation of $P(\boldsymbol{w}|\boldsymbol{s})$ can be done by means of posterior n-grams from the ASR word graph. Initially, we can find which alternative to use empirically. However, the intuition says that the system with lower error rate would constitute a better prior for the system with higher error rate.

3.2. Iterative framework

The main contribution of our proposal with respect to our initial approach (Alabau et al., 2011) is the iterative procedure. Here, HTR and ASR are used alternatively in order to improve the final result of the system.

In general, we can state the problem as: given two input signals S_a and S_b in two different modalities $\mathcal{M}_a$ and $\mathcal{M}_b$ that represent the same object, follow the process:

1. Obtain the initial set of models m_a^0 of modality $\mathcal{M}_a$
2. Perform the classification process with S_a with the set of models m_a^0
3. Derive from the classification of S_a a set of models m_b^0 of modality $\mathcal{M}_b$
4. Perform the classification process with S_b with the set of models m_b^0
5. Let $i = 1$
6. Derive from the classification of S_b with models m_b^{i-1} a set of models m_a^i of modality $\mathcal{M}_a$
7. Perform the classification process with S_a with the set of models m_a^i of modality $\mathcal{M}_a$
8. Derive from the classification of S_a a set of models m_b^i of modality $\mathcal{M}_b$

Input: $\boldsymbol{x}, \boldsymbol{s}, I$
Output: $\boldsymbol{w}$
$i \leftarrow 0$;
$\hat{\boldsymbol{w}_s}^{(0)} \leftarrow \text{argmax}_{\boldsymbol{w}}\, P(\boldsymbol{s}|\boldsymbol{w})P(\boldsymbol{w}|\boldsymbol{x})$;
repeat
  $i \leftarrow i + 1$;
  $P^{(i)}(\boldsymbol{w}|\boldsymbol{s}) \leftarrow n$-gram posteriors from WG($\boldsymbol{s}$) in iteration $i - 1$;
  $\hat{\boldsymbol{w}_x}^{(i)} \leftarrow \text{argmax}_{\boldsymbol{w}}\, P(\boldsymbol{x}|\boldsymbol{s}, \boldsymbol{w})P^{(i)}(\boldsymbol{w}|\boldsymbol{s})$;
  $P^{(i)}(\boldsymbol{w}|\boldsymbol{x}) \leftarrow n$-gram posteriors from WG($\boldsymbol{x}$) in iteration i;
  $\hat{\boldsymbol{w}_s}^{(i)} \leftarrow \text{argmax}_{\boldsymbol{w}}\, P(\boldsymbol{s}|\boldsymbol{x}, \boldsymbol{w})P^{(i)}(\boldsymbol{w}|\boldsymbol{x})$;
until $\hat{\boldsymbol{w}_s}^{(i)}$=$\hat{\boldsymbol{w}_s}^{(i-1)}$ **or** i=I;
return $\hat{\boldsymbol{w}_s}^{(i)}$

Algorithm 1: Iterative decoding algorithm.

9. Perform the classification process with S_b with the set of models m_b^i
10. If results with sets of models m_b^i and m_b^{i-1} are different, increment i and go to 6

In our case, the modalities are images of handwritten text and speech signals that represent the same sentence. The initial set of models are, in the case of handwritten text, the HMM that represent the different characters and, in the case of speech signals, the HMM that represent the different sounds (phonemes). In the two modalities, a language model and a lexical model are used as well. Lexical models indicate the composition of each word, and they are different for each modality (since there is no a one-to-one association between characters and sounds).

However, the language model, since it models the composition of the final object (sequences of words) is shared by the two modalities. Actually, this capability of using the same language model for the two modalities allows us to perform the derivation of the models for the other modality. This derivation follows the framework presented in Section 3.1.

A specialisation of this procedure for our particular case, supposing that $\mathcal{M}_a$ is handwritten text and $\mathcal{M}_b$ is speech, is represented in Algorithm 1. Note that we have added a new input to the procedure, I, that represents the maximum number of iterations. This is necessary since there is no proof for convergence for this algorithm and, in practice, it does not converge in some specific cases.

4. Experimental framework

4.1. Corpora

The experiments have been carried out using a Spanish corpus compiled from a legacy handwritten document identified as "Cristo-Salvador", from the XIX century, which was kindly provided by the *Biblioteca Valenciana Digital* (BIVALDI)[1].

The document was written by only one writer and scanned at 300dpi. As a legacy document, it suffers from the typical degradation problems of this kind of documents (Drida, 2006). Among these are the presence of smear, significant background variations and uneven illumination, humidity spots, and marks resulting from the ink that goes through the paper (generally called bleed-through). In addition, other kind of difficulties appear in these pages as different sizes in the words, underlined words, etc. The combination of these problems makes the recognition of this document a difficult process. This is a rather small document composed of 53 colour images of text pages. In Figure 3 there is one of these pages.

The page images were preprocessed and automatically divided into lines (Romero et al., 2007). The results were visually inspected and the few line-separation errors (around 4%) were manually corrected, resulting in a data-set of

[1] http://bv2.gva.es

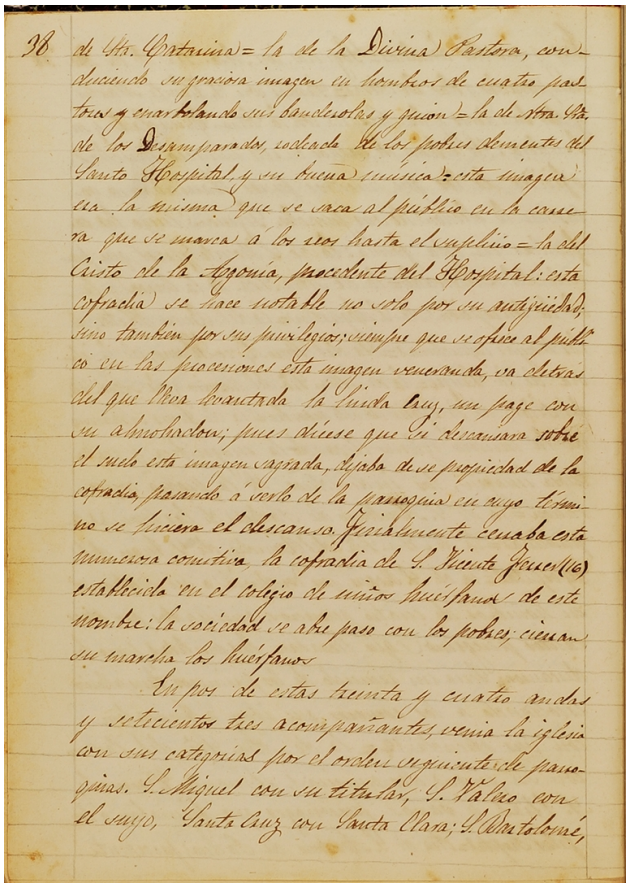


Figure 3: Detailed section of a page from the corpus “Cristo-Salvador”.

Table 1: Basic statistics of the partition *book* of the database Cristo-Salvador.

Number of:	Training	Test	Total Lexicon
Pages	33	1	–
Text lines	675	24	–
Words	6,227	222	3,257
Characters	35,863	1,239	78

1,172 text line images. The transcriptions corresponding to line images are also available, containing 10,918 running words with a vocabulary of 3,287 different words.

In this work, we have used the training set of the so called *book* (or *hard*) partition of the *Cristo Salvador* corpus. This partition is the one that best approaches a realistic transcription process. To assess our system, we have chosen one of the original test pages, number 41, which is the page with the most similar error distribution with respect to the global error. Table 1 summarizes the information of this partition.

It is important to remark that this corpus has quite a small training ratio (around 2.8 training running words per lexicon-entry). This is expected to result in undertrained (n-gram) language models, which will clearly increase the difficulty of the recognition task.

In order to train the acoustic HMMs needed in the speech recognition system, we have used the *Albayzin geographic corpus*, which consists of oral queries to a geographic database. It contains phonetically balanced spoken sentences in Spanish, which were acquired in a project unrelated to the present work (Moreno et al., 1993). Some

Table 2: Basic statistics of the Spanish speech acoustic training corpus Albayzin.

Speakers	164
Running words	42,000
Length (hours)	4

Table 3: Basic statistics of the speech dictation test.

Speakers	5
Running words	222
Running characters	1,239
Length (seconds)	454

details can be found in Table 2.

In addition, in order to assess the speech dictation systems five different users dictated the selected page from the *Cristo Salvador* corpus line by line. It resulted in a test data-set composed by 120 dictated lines. Some basic details are shown in Table 3.

4.2. Models

As was mentioned above, the recognition process is based on HMMs. For the HTR system, the characters (78 different symbols including capital and lowercase letters, digits, and punctuation marks) are modelled by continuous density left-to-right HMMs with 12 states and 32 Gaussian mixture components per state. The optimal number of HMM states and Gaussian densities per state were tuned empirically. Speech models are HMM with three states, with left-to-right with loops topology and 64 Gaussians per state (a total number of 4.6K Gaussians). Each speech model represents a phonetically context-independent unit (monophone). These models were estimated using the data of the Albayzin Spanish speech corpus Moreno et al. (1993), of about 4 hours of speech signal. All HMM were trained with the HTK tool (Young et al., 2006). In the two systems, each lexical word is modelled by a stochastic finite-state automaton (SFSA), which represents all possible concatenations of individual characters or phonemes to compose the word. On the other hand, the baseline language model for text lines is a 2-grams with Kneser-Ney back-off smoothing Kneser and Ney (1995) directly estimated from the training transcriptions of the text line images.

4.3. Evaluation metrics

Our system was assessed by means of *word error rate* (WER), which obtains the ratio between the number of editions of the Levenshtein distance and the number of words in the reference. In addition, in order to evaluate the quality of the chosen LM, we have employed perplexity Rosenfeld (2000). Perplexity, measured for a text with respect to a LM, is a function of the likelihood of that text being produced by repeated application of the model. Significance of our results has been assessed by the *paired bootstrap resampling* method, described in Bisani and Ney (2004). This technique compares two systems and finds out whether one of them significantly outperforms the other one.

5. Results

This section is devoted to analyse the experimental results of the methods proposed in Section 3 and the iterative process described in Section 3.2. The results will be compared against a baseline result which uses only one modality and in a non-iterative fashion.

5.1. Baseline results

The baseline results are those obtained with the initial models presented in Section 4.2 for the two modalities. All the recognition processes were performed by using the iATROS system [2]. Recognition parameters were tuned on the test set to obtain the optimal results.

[2] http://prhlt.iti.upv.es/page/projects/multimodal/idoc/iatros

Table 4: Baseline results for handwritten text and speech modalities, along with average decoding times for each sample (in seconds) and each feature (in miliseconds)

Modality	WER	Time per sample	Time per feature
HTR	29.7 ± 6.7	224 sec	1.72 msec
ASR	43.2 ± 3.3	25 sec	1.56 msec

Some details must be given to interpret the results. In the recognition system, the quality of the language model and the morphological/acoustic models are responsible for the recognition accuracy. Morphological models for HTR are in this case trained with a subset of the corpus; since all the book was written by a single writer, the morphological models are well estimated for that writer. However, acoustic models for ASR are speaker independent, and consequently not so adapted to the test speakers. Moreover, language model is estimated from a small corpus ($6.4k$ running words), which causes a poor estimation of the language model probabilities (actually, for a 2-gram the perplexity is of 552, and it does not improve when using higher degree n-gram models). Since the corpus addresses a very specific topic, it is difficult to obtain a more representative language model.

With this conditions, we can expect a better baseline result for HTR than for ASR, since HMM for HTR fit better to test conditions that HMM for ASR, while the language model is the same (with the same perplexity). All the results were obtained in the same conditions: punctuation marks were not considered; initial, final, and silence/blank symbols were eliminated from the decoder output; all words were transcribed to capital letters. The results obtained by the iATROS system (including 90% confidence intervals), along with approximated decoding times per sample and feature (considering as feature each element of the feature vectors), are presented in Table 4. These results are similar to those presented in Alabau et al. (2011), although HTR results are now obtained with the iATROS system.

As was expected, HTR results are significantly better than speech results. The different magnitude of the confidence intervals is caused by the size of the test set (24 lines for HTR and 120 sentences for ASR). The results are coherent with our previous hypothesis. Moreover, decoding times are an order of magnitude lower for ASR than for HTR, which allows us to predict that in the iterative recognition process ASR will not add a significant amount of extra time to the process and makes feasible its use in real transcription applications.

5.2. Iterative results

In the iterative framework, the initial decoding is performed in one of the modalities (HTR or ASR) and the next step is obtaining the 3-gram model from the recognition word graph (according to Equation (13)) and perform the interpolation with the original 2-gram (Equation (14)). Both operations are done by using the tools implemented in the SRILM toolkit (Stolcke, 2002) (`lattice-tool`, `ngram-count`, and `ngram`). The posterior scale and interpolation parameters were optimised in previous experiments and were kept with the same value along all the process. The decoding parameters for the iATROS system were kept to those used in the baseline experiments, although in the ASR process were slightly modified according to a study of the HTR word graphs obtained in Alabau et al. (2011).

Two different approaches were used: start with HTR modality and start with ASR modality. The process stops when the hypothesis of the non-starting modality does not change with respect to the hypothesis in the previous iteration. In any case, the process was limited to 10 iterations.

Results obtained for this process are presented in the Tables 5 and 6, each one for HTR start and ASR start, respectively.

5.3. Comparison and error analysis

5.3.1. Iterative from initial HTR

Starting from HTR recognition, a large reduction in WER is obtained by only applying the ASR step on the new language model. The obtained result differs from that obtained in Alabau et al. (2011), but this is justified by two differences in the process:

1. Decoding parameters for ASR where obtained by global optimisation on HTR word graph results, instead of using a leaving-one-out individual optimisation.

Table 5: WER and time results for the iterative process starting from the HTR process. In italics, the baseline result. Convergence is assumed when hypothesis of ASR in one iteration does not change from the hypothesis of the previous iteration.

Iteration	Modality		Time per sample
	HTR	ASR	
0	*29.7 ± 6.7*	20.6 ± 2.4	256 sec.
1	25.8 ± 2.7	20.1 ± 2.4	470 sec.
Convergence	25.4 ± 2.7	19.9 ± 2.4	585 sec.

Table 6: WER and time results for the iterative process starting from the ASR process. In italics, the baseline result. Convergence is assumed when hypothesis of HTR in one iteration does not change from the hypothesis of the previous iteration.

Iteration	Modality		Time per sample
	ASR	HTR	
0	*43.2 ± 3.3*	25.5 ± 2.5	232 sec.
1	18.4 ± 2.6	24.3 ± 2.5	462 sec.
Convergence	18.3 ± 2.4	24.1 ± 2.5	1244 sec.

2. Initial HTR word graphs are those provided by iATROS, which are less dense that HTK word graphs; in contrast, decoding times for iATROS are two orders of magnitude lower than HTK decoding times.

In any case, differences between baseline HTR and ASR are statistically significant, since confidence intervals do not overlap.

In the iterative process which starts with HTR, the distance between confidence intervals of ASR and the baseline HTR gets higher in each iteration till convergence, although differences are not significant for ASR results from initial iteration to convergence. This similarity between these results is caused by the small number of iterations required to obtain convergence: most of the samples (80%) converge with an only iteration (10% require 2 iterations and only 2.5% do not converge within a limit of 10 iterations). Consequently, we can assume that using only one iteration is enough to obtain the better results. This reduces the time needed for obtaining the final hypothesis.

However, in this iterative process starting with HTR, the iterative HTR results do not present a statistically significant improvement with respect to baseline. Moreover, HTR results are always worse than ASR results. In this case, the cause can be the decoding parameters, that were kept to the same value than in the baseline HTR experiment without performing an optimisation on the ASR word graphs that provided the language model for the iterative HTR decoding.

Error analysis was centered in some special cases that are particular to this HTR task:

- Hypenathed words: including the first part (with an hypenation symbol - at the end) and the second part (that starts in the following line).
- Abbreviations: in this case, =, *NTRA*, *S*, *STA*, and *SRA*; they are pronounced as whole words in ASR but kept as abbreviations in HTR.
- Numbers: in this case, *(16)*, *38*, and *TREINTA Y CUATRO*; in ASR, the same lexical model represents different words (e.g., *(16)* and *16*).

The comparison between the results of convergence HTR and ASR showed small differences, that concentrated mainly on abbreviations = and *STA*, and on number *(16)*. ASR presents a quite lower error rate in these cases with respect to those obtained by HTR. This can be caused by the scarce presence of the corresponding symbols (=, numbers) in the training test for HTR, whereas in ASR the corresponding phones are as usual as those of other words.

This causes a poor estimation of the HMM associated to the symbols, which explains the differences in this case and the better results of the ASR variant.

5.3.2. Iterative from initial ASR

Starting from ASR recognition, the only application of HTR using the language model derived from the output word graph obtains a dramatical decrease of WER with respect to baseline ASR results. However, when comparing with the HTR baseline results, differences are not significant.

When starting with the iterative process, the ASR results become significantly better than the baseline results for any of the modalities, although HTR results present a worse WER. This behaviour is similar to that presented by the iterative process starting from HTR, where in each iteration HTR results are far worse than ASR results. As happened in that previous case, the configuration of the decoding parameters seem the cause of this result.

Anyway, ASR results in this iterative process are the best results that are obtained with the test set (18.3% of WER with respect to the original baseline of 29.7% in the HTR baseline). Moreover, although the number of iterations for convergence is higher that in the iterative process starting from HTR (only a 44% of the samples converge in one iteration and more than a 28% do not present convergence), results in first iteration for ASR are very similar to convergence results. Thus, only one iteration is enough for having an accurate result, which implies a faster process.

Error analysis showed similar conclusions that those of the iterative process starting from HTR. The lack of convergence in the HTR hypothesis during this process was caused mainly by the alternative appearance of two very similar (but different) hypothesis in each step (i.e., hypothesis *A* appeared in odd iterations and hypothesis *B* appeared in even iterations).

6. Conclusions

In this article we have tackled the transcription of handwritten historical documents from a multimodal (HTR and ASR) and iterative perspective. The multimodal system can be compared against two uni-modal baseline systems and the iterative approach can be compared against the non-iterative multimodal system.

The multimodal approach shows a significant improvement in results with respect to the baseline uni-modal system. The adoption of the iterative approach shows a greater improvement in the results with respect to the baseline uni-modal systems, although not always significantly better than multimodal non-iterative results.

The influence of the initial modality in behaviour of the multimodal iterative approach was studied for the two modalities. Results show that starting from ASR allows for a better final performance. In all cases of the iterative process, HTR results present a poorer result that their ASR counterparts, but their decoding parameters seem not as optimal as the decoding parameters of the ASR systems.

Although errors are significantly less than those of baseline systems and time response is appropriate for the transcription task, more work can be done to improve these two aspects and obtain a better performance in a real system. On the one hand, HTR decoding needs a better tuning for the recognition parameters, that can be obtained in a similar way to those obtained for the ASR decoding; the improvement in the HTR decoding could cause a positive impact in the final performance of the iterative systems, both in terms of WER and convergence time. On the other hand, an integrated decoding of HTR and ASR is a suggestive alternative to the iterative paradigm that can reduce dramatically the final time and obtain better quality in the automatic transcriptions.

Acknowledgements

Work supported by the EC (FEDER/FSE) and the Spanish MEC/MICINN under the MIPRCV "Consolider Ingenio 2010" program (CSD2007-00018), iTrans2 (TIN2009-14511) and MITTRAL (TIN2009-14633-C03-01) projects. Also supported by the Spanish MITyC under the erudito.com (TSI-020110-2009-439) project and by the Generalitat Valenciana under grant GV/2010/067, and by the UPV under grant UPV/2009/2851.

References

Alabau, V., Romero, V., Lagarda, A. L., Martínez-Hinarejos, C. D., 2011. A multimodal approach to dictation of handwritten historical documents. In: Proceedings of the Interspeech 2011. pp. 2245–2248.

Bengio, S., 2004. Multimodal speech processing using asynchronous hidden markov models. Information Fusion 5 (2), 81 – 89.

Bisani, M., Ney, H., May 2004. Bootstrap estimates for confidence intervals in asr performance evaluation. In: Proceedings of ICASSP2004. Vol. 1. Montréal, Canada, pp. 409–412.

Campbell, W., Richardson, F., 2008. Discriminative keyword selection using support vector machines. In: Platt, J., Koller, D., Singer, Y., Roweis, S. (Eds.), Advances in Neural Information Processing Systems 20. MIT Press, Cambridge, MA, pp. 209–216.

Casacuberta, F., Higuera, C. D. L., 2000. Computational complexity of problems on probabilistic grammars and transducers. In: ICGI '00: Proceedings of the 5th International Colloquium on Grammatical Inference. Springer-Verlag, London, UK, pp. 15–24.

Drida, F., 2006. Towards restoring historic documents degraded over time. In: Proceedings of the DIAL'06. IEEE Computer Society. Washington, DC, USA, pp. 350–357.

Jelinek, F., 1998. Statistical Methods for Speech Recognition. MIT Press.

Kneser, R., Ney, H., 1995. Improved backing-off for m-gram language modeling. In: Proceedings of ICASSP 95. Vol. 1. IEEE Computer Society, Los Alamitos, CA, USA, pp. 181–184.

Liu, P., Soong, F., march 2009. Graph-based partial hypothesis fusion for pen-aided speech input. Audio, Speech, and Language Processing, IEEE Transactions on 17 (3), 478 –485.

Liu, P., Soong, F. K., 2006. Word graph based speech recognition error correction by handwriting input. In: ICMI '06: Proceedings of the 8th international conference on Multimodal interfaces. ACM, New York, NY, USA, pp. 339–346.

Moreno, A., Poch, D., Bonafonte, A., Lleida, E., Llisterri, J., Mariño, J. B., Nadeu, C., sep 1993. Albayzin speech database: design of the phonetic corpus. In: Proceedings of EuroSpeech'93. Berlin, Germany, pp. 175–178.

Ortmanns, S., Ney, H., Aubert, X., 1997a. A word graph algorithm for large vocabulary continuous speech recognition. Computer Speech and Language 11 (1), 43 – 72.
URL http://www.sciencedirect.com/science/article/B6WCW-45N4RJB-N/2/0f22c5b07ee9378da100a928907910e7

Ortmanns, S., Ney, H., Aubert, X., Jan. 1997b. A word graph algorithm for large vocabulary continuous speech recognition. Computer Speech and Language 11 (1), 43–72.

Rabiner, L., 1989. A Tutorial of Hidden Markov Models and Selected Application in Speech Recognition. Proceedings IEEE 77, 257–286.

Romero, V., Toselli, A. H., Rodríguez, L., Vidal, E., August 2007. Computer Assisted Transcription for Ancient Text Images. In: ICIAR 2007. Vol. 4633 of LNCS. Springer-Verlag, Montreal (Canada), pp. 1182–1193.

Rosenfeld, R., 2000. Two decades of statistical language modeling: Where do we go from here? In: Proceedings of the IEEE. Vol. 88. pp. 1270–1278.

Stolcke, A., 2002. SRILM – an extensible language modeling toolkit. In: Proceedings of ICSLP. Vol. 2. Denver, USA, pp. 901–904.

Toselli, A. H., Juan, A., González, J., Salvador, I., Vidal, E., Casacuberta, F., Keysers, D., Ney, H., 2004. Integrated Handwriting Recognition and Interpretation using Finite-State Models. IJPRAI 18 (4), 519–539.

Toselli, A. H., Vidal, E., Casacuberta, F. (Eds.), Jun 2011. Multimodal Interactive Pattern Recognition and Applications, 1st Edition. Springer, http://www.springer.com/computer/hci/book/978-0-85729-478-4.

Vidal, E., Thollard, F., Higuera, C., Casacuberta, F., Carrasco, R. C., 2005a. Probabilistic finite-state machines - part i. IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (7), 1013–1025.

Vidal, E., Thollard, F., Higuera, C., Casacuberta, F., Carrasco, R. C., 2005b. Probabilistic finite-state machines - part ii. IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (7), 1025–1039.

Wessel, F., Schluter, R., Macherey, K., Ney, H., Mar 2001. Confidence measures for large vocabulary continuous speech recognition. IEEE Trans. Speech and Audio Proc. 9 (3).

Young, S. J., Evermann, G., Gales, M. J. F., Hain, T., Kershaw, D., Moore, G., Odell, J., Ollason, D., Povey, D., Valtchev, V., Woodland, P. C., 2006. The HTK Book, version 3.4. Cambridge University Engineering Department, Cambridge, UK.