UNIVERSITAT POLITÈCNICA DE VALÈNCIA

# Advances in Interactive Speech transcription

.

Master Thesis at the
Pattern Recognition and Language Technologies Group

by Isaías SÁNCHEZ CORTINA

Advisors:
Dr. Alfons Juan i Ciscar
Dr. J. Alberto Sanchis Navarro

.

~ Valencia, June, 2012 ~

# ACKNOWLEDGEMENTS

Work supported by the Ministerio de Ciencia e Innovación (MICINN) under iTrans2 (TIN2009-14511) project
and the the FPI Scholarship BES-2010-033005;
And by the the Generalitat Valenciana under the grant GV/2010/067.

# CONTENTS

# CHAPTER 1

# INTRODUCTION

Speech transcription is a crucial task in a broad range of important applications. Speech transcriptions produced by human transcribers can provide high quality results. However, the overall process is very slow and usually expensive. One way to deal with this important drawback is to produce automatically speech transcriptions based on automatic speech recognition (ASR) technology [8, 16]. However, this solution presents two main difficulties. First, the building of an ASR system implies usually an important human effort since statistical models have to be obtained. Acoustic and language models are not easily available for specific tasks and, thus, they have to be learned from manual annotated data. The second difficulty is that automatic transcriptions are still far from producing the desired quality out of some specific scenarios. Therefore, an important human effort to supervise the speech transcriptions is mandatory. Sometime, for very faulty transcriptions, the effort can even be higher than manually transcribing the whole.

To deal with the first difficulty, active and unsupervised learning techniques have been applied to rapidly prototype ASR systems reducing significantly user effort [4, 19]. Upon these approaches, a little manually annotated data is used to build rapidly an ASR system. Then, this initial ASR system is used to automatically transcribe a large amount of new speech data. These new annotated data is used to improve the underlying ASR models based on active and unsupervised learning techniques. To overcome the second difficulty, an interactive paradigm has been applied to reduce significantly the supervision effort [6, 9] of the speech transcriptions. Following this paradigm, the system produces automatically speech transcriptions and the user is assisted by the system to amend output errors as efficiently as possible.

In this work, we present a novel speech transcription system in which active and unsupervised learning techniques are applied along within an interactive paradigm in a tightly coupled manner. The main goal is to significantly reduce the human effort in the transcription of a speech task by allowing a maximum tolerance error in the resulting transcriptions. The supervision is performed while an estimation of the transcription errors is higher than the tolerance error. Word-level confidence measures [20] computed over the recognition word-graph are used to suggest which words should be supervised. During the transcription process, supervised and high-confidence parts of the transcriptions are used to improve incrementally the underlying statistical ASR models. This will likely improve the subsequent recognitions lowering the necessary number of supervisions.

This approach was successfully applied to handwriting transcription [13]; Here, we adopted it and improved the method for the estimation of the error . The empirical results confirm previous works on handwriting recognition showing that this strategy is effective to reduce the supervision effort by allowing a maximum tolerance error in the speech transcriptions. Moreover, results show that a tolerance error in the transcriptions does not affect critically on the incremental learning of acoustical models. Thus, this method can be used also for producing ASR models resulting in similar performance to those generated using fully manually transcribed corpora. In summary, this is useful approach implemented with a simple yet effective method to find an optimal balance between recognition error and supervision effort for the interactive transcription of the speech .

Let us summarize the layout of this work: In chapter 2, the foundations of the ASR foundation are explained briefly, as well as the interactive speech transcription (IST). Chapter 3 details our interactive approach and the method for balancing the error and user effort and the main contribution respect to related previous works in the literature. Next, in chapter 4, a general description objective magnitudes that might be useful to describe the performance of an IST system in terms of the user effort the features, and then, an evaluation resulting from applying the method for the task of transcribing the speech of two sets of the World Street Journal Speech database of 15 and 80 hours respectively. 5 describes an shows an interface developed for our IST method. Finally, conclusions on our method and future work are discussed on chapter 6 .

# CHAPTER 2

# INTERACTIVE APPROACH TO SPEECH TRANSCRIPTION

## 1. Foundations of the Automatic Speech Recognition

Speech Recognition is founded on the assumption that the natural speech can be described by means of probabilistic models in a satisfactory way. Under this framework, the facto standard is that in which the transcription is obtained as the maximum a posteriori probability (MAP) of a sequence of words ($\vec{w}$) given the sequence of acoustic observations ($\vec{X}$ , which are derived from the audio of the speech):

$$\hat{\vec{w}} = \text{argmax}_{\vec{w} \in W} P(\vec{w}|\vec{X}) = \text{argmax}_{\vec{w} \in W} P(\vec{X}|\vec{w}) \cdot P(\vec{w}). \tag{1}$$

The right hand term has been obtained by applying the Bayes rule. This has been the standard way to proceed over the last decades since it presents several advantages: It allows the search for the most probable transcription (string decoding) to be effectively pruned thanks to term $P(\vec{w})$ , which is called the language model (LM). LMs can incorporate both syntactic and semantic constraints of the language and the recognition task. Also, they can be trained using the standard methods for the estimation of the n-grams probabilities from text resources independent to the speech task.

On the other hand, the generative model $P(\vec{X}|\vec{w})$, called the acoustic model (AM), can be estimated in a easier and more precisely way than the discriminative model $(\vec{w}|\vec{X})$. However, for large vocabulary speech recognition systems, it is necessary to build statistical models for sub word speech units, build up word models from these sub word speech unit models (using a lexicon to describe the composition of words), and then postulate word sequences and evaluate the acoustic model probabilities via standard concatenation methods.

The commonly used sub word units are the phonemes, or tri-phonemes. And the most accepted probabilistic for the representation of these units are the hidden Markov models (HMMs) . HMMs representations are good in matching patterns of high variability, like the realization of the phonemes. The main parameters of the HMMs are its structure and the emission distribution of the states. The structure is defined by number of internal states and the allowed connections amongst them. A state is a hidden discrete variable thanks to which the model achieves some amount of histeresis depending on the already seen observations. The emission distributions are the probability of a phoneme given the the observations and current value of of the HMMs . The standard emission distributions are the Gaussian mixtures . Additionally, the transition probabilities from state to state also play a key role in the structure. But it has been showed that its values have little impact on the recognition performance.

The training procedure of the HMMs an LMs , as well as the recognition process is beyond the scope of this introduction. But is should be noted that decoding with the Viterbi algorithm using HMMs representations can be efficiently computed in linear time with the number of states and the number of acoustic vectors (proportional to the duration of the audio signal). Furthermore, the LM will reduce the number of possible paths, and even more if some state-of-the-art LM look-ahead techniques are applied along the viterbi search. In summary, HMMs have been so successful for speech recognition because of their computational efficiency and their ability to match variable signals.

## 2. Unsupervised Learning

The training of accurate HMMs usually requires a high amount of transcribed speech. And, usually, the training data (speech) should be similar enough to the task to automatically transcribed (in terms of the speech style, the environment, the speakers, etc.). But, as commented in the introduction, the manual process is slow and expensive, usually requiring of professional transcribers. Fortunately, HMMs can be improved even when no transcription is available using the own output of an ASR. In the context of training, this way of learning (or adapting the initial model) is known as unsupervised learning. Thus, better ASR model might be obtained with no extra human supervision than the that necessary to just bootstrap the initial models.

The simplest and most extended way for the unsupervised learning confidence values is as in [19]. The confidence values can be derived using features computed over the recognition word-graphs. The word-graphs are structures containing not only the most probable decoded transcription but also other less probable paths. Also, the structure include recognition scores, time alignments, etc. The confidence values of the recognized transcription (at sentence , word, or phoneme level) serve to select pieces of the recognized samples which are likely to have been properly recognized. These selected pieces of samples and their corresponding transcriptions can added to the existing training data so the ASR model can be then re-estimated. Precisely, this is the method for unsupervised learning which has been used in this work.

Further refinement can be achieved as in [4]. In that work, the pieces of the recognized transcriptions selected for being new training data, will have an reduced impact in the LM estimation. This is achieved by modulating the counts of the recognized words by their confidence values. However, it is patent that the main difficulty for the unsupervised learning in getting improved models is due to the performance of the confidence measure in predicting whether words are correctly or incorrectly recognized. Tasks in which the confidence measure yields a high classification error ratio (CER), the unsupervised learning might even worsen the initial models.

## 3. The Interactive Speech Recognition

Interactive Speech Transcription (IST), from a general point of view, is the process of obtaining the transcription of a speech in the core of an automatic system with the help of a user. Thereby combining human accuracy with ASR efficiency.

The contribution of the user can be as much as typing the full manual transcription, or as little as a few clicks or keystrokes, or any other modal interaction. On the other hand the system may provide as little as just a convenient interface; automatic segmentation; predicting the text being typed (ex. [10]); or as much as performing the whole transcription automatically and then asking the user for supervision. Thus, depending on whether the system integrates ASR engine or not , the paradigm will be focused on transcription or supervision.

The method to be presented in this work assumes the interactive speech transcription paradigm in which the human operator supervises the ASR output within an interactive editing environment in order to generate the final transcription. The main goal in this paradigm is to to reduce user effort: In the framework of this paradigm, different approaches have been proposed for the purpose of obtaining completely accurate transcripts of speech [6, 9]. Here, however, we will focus on reaching a balance between the user effort on supervision and the final residual transcription error. Few research on this approach can be found on the literature (ex. [4] and previous works from the same authors).

# CHAPTER 3

# BALANCING THE ERROR AND SUPERVISION EFFORT

In the following, an Interactive speech transcription procedure is presented for the purpose of obtaining non-perfect transcriptions but improved respect to the automatic output. Section 2 explains the method used to predict transcription error. Next, a clarification on the use of the confidence measures. In the last section, the main difference between this and the other similar published works are highlited.

## 1. Interactive Speech Transcription Procedure

(1) **Initialization**:
   (a) The task should be split into smaller audio blocks.
   Preferably, each one containing several utterances (samples) .
   (b) The user must establish the acceptable minimum quality on the final transcriptions by means of a tolerance error ($W^*$).

The semi-supervision takes then place iteratively one block after another:

(2) **Recognition**: A block is transcribed using the last trained ASR models
(3) **Interaction**: For each new automatically transcribed utterance:
   (a) $\hat{W}^-$ is estimated including the whole utterance under study .
   $\hat{W}^-$ is the predicted error on the unsupervised transcribed parts .
   (b) While $\hat{W}^- > W^*$:
      (i) The user is asked to supervise the next lowest-confidence word in the utterance.
      (ii) $\hat{W}^-$ is updated accordingly.
(4) **Learning**:
   (a) New samples are extracted from the supervised and high-confidence parts, and added to the training set.
   (b) A new ASR system is trained using the available training data.

Further explanation of each one of these steps is to follow on the next sections.

**1.1. Initialization.** First, the number of the blocks and their duration have indeed an impact in the overall process. In particular, the more number and the smaller size of the blocks, the faster the ASR system will get adapted; yet much additional computation time will be required. Thus the optimal number depends on the scenario and the urge for adaptation of the ASR models to the task. For instance, in case of users that are expert transcribers, it seems wise to split the task into blocks of a duration such that each block could be semi-supervised in a day of their daily-work routine. This way, the hard computational part would be carried on during the nights.

Moreover, apart from the segmentation of the data into blocks, each block should also be segmented into utterances, as assumed in section 1. Intentionally, it has been omitted how long should the utterances be (one word, one sentence, one paragraph or the whole block long). This is because the method will operate properly regardless of the segmentation. Nonetheless, the segmentation level will indeed vary the overall performance of the method, as commented later.

It should be noted that published results using the same approach for handwriting transcription (HWR) [13], were word-level based, although the samples were segmented at line-level since this is quite straight for HWR . However, in this work we will focus on the sentence-level procedure . In chapter 4 it will be showed that the conducted experiments outperform for the sentence-level based procedure. So, in case of unsegmented audio, automatic segmentation should be performed. Fortunately, the automatic segmentation has been a deeply researched topic on the literature (for instance, see the recent work [1]) .

Of course, badly segmented samples can degrade the performance on the recognition and supervision at the extremes; but the impact is usually insignificant, especially if some degree of errors is to be tolerated. Anyhow, manual segmentation is discouraged because it would require too much user effort (i.e. listening once the whole speech at least ).

Finally, on the tolerance error $W^*$, it is mandatory to establish its value at the very beginning. Nevertheless, users can change the threshold later on in order to reduce or increase the amount effort they are willing to do. The interface presented in chapter 5 allows for this online modification.

**1.2. Recognition.** The ASR system generates the most probable transcription for each of the utterances, as well as word-nets which include other less probable paths and the recognition scores. These word-nets will be used to calculate confidence measures and time alignments for each of the recognized words. These two features play a key role in the interactive procedure, as explained later.

It should be recalled that an ASR system needs an Acoustic Model (AM) and Language model (LM) in order to operate. Both of them are to be updated during the re-training phase. However, it is mandatory to bootstrap the system with initial versions of the AM and the LM. These models might be obtained from external resources similar enough to the task. But, while there are more chances of building a suitable external LM because of the high amounts of text data available, it is not usually feasible to build up a proper AM for every task. Fortunately, the initial models can be built up using a small fully transcribed part of the task to fulfill; so the user will just have to manually transcribe a small part of the speech when no external ASR models are available.

It is worthy to note that in order to achieve reasonable performance, an ASR system needs some of its operational parameters to have values adapted to the current task . For instance, the Grammar Scale Factor (GSF) and the Word Insertion Penalty (WIP) ; which can be safely mentioned as the most influent parameters given certain pruning parameter values . The optimal values should be updated every some number of blocks or even every time a new ASR system is trained. The search for the optima is performed over a development set. This set can have been obtained from a external resource, or by selecting some of the couples of the utterances and their transcriptions from the part of the task that would have had to be fully transcribed. Even, better adaptation can be achieved if the development set is modified after each step by including some data from the lasts semi-supervised blocks, and possible removing some old data. Unfortunately, the search process is computationally very expensive. But, if the audio blocks to recognize are similar enough to the data used to train the current ASR models, this optimization might be carried just once at the very beginning .

**1.3. Interaction.** First, an estimation of the error in the transcription is performed, then if necessary the user is asked to supervise a recognized word . And finally, the learning (updating ASR models) phase.

- Error estimation As suggested above, the main driver for the balancing of the error and user effort is the estimation of the accumulated WER of the semi-supervised parts of the transcription. In section 2 below, the actual method for this estimation is detailed. However, the presented procedure is not limited to this estimation. So, more complex estimators might be proposed for this same procedure. For instance, a cost function including an estimation of the user effort in listening and typing might be used.
- User interaction Once a recognized word is selected for supervision, only the associated audio segment should be played. This is why the time alignment at word level is necessary. Then, the corrections introduced by the user will be assumed to have the highest possible value of the confidence measure.

It should be noted that the time necessary to fulfill a manual transcription is several times greater than the duration of the speech. The exact factor depends also on the speech style, speakers, and mainly on the particular user who does the transcription. Supervising and correcting usually takes less than a full manual transcription and always more than total listened time; unless the transcription to be supervised has a very high error rate. Thus reducing the duration of the audio to be listened by the user should decrease the global supervision time in a effective way.

Nevertheless, understanding utterances of isolated words poses several issues: The linguistic context helps in the speech understanding. But listening to isolated words makes it difficult to have a good grasp of the topic or , even worse, to realize when the context has changed. Furthermore, the time alignments of the recognized words usually differ from the manual alignments. And this is fatal for speech understanding, since listening to a hundred of mili-seconds more or less can make the user not understand the word, or figuring another different word. Besides, unlike in writing, in natural speech there is little or no separation between the vast majority of the words. Especially, for spontaneous speech , usually the speakers make no pauses when a long pause is expected, or they do in the middle of the words. Some other linguistic aspects in the realization of the speech, such as the cross word

One possible workaround is to enlarge the piece of audio to be played. A simple fixed-length increase of the margins might prevent the chopped spelling of some words; although most of the issues would still hold. Possibly, a dynamic variation in the length using the phoneme-level alignments would improve the chances of playing whole words. But this is left as an open issue out of the scope of our work.

Fortunately, the text context helps in understanding. Even with no speech, it has been proven that users can correct some kind of errors just by reading the transcribed sentences.

In any case, in order to overcome these difficulties, user should adopt some conventions. For instance, chopped words which the user can positively guess should be only transcribed (this might be a substitution, insertion or equal operation) if more most of the spelling is listened; or they should be deleted when not. In case of no recognized words, the full sample audio should be played.

**1.4. Learning .** ASR models will be re-estimated each time after a new recognized audio block is semi-supervised. Models are expected to progressively adapt and perform better, because the training corpus is iteratively augmented with new samples obtained from the supervised and high-confidence parts of the last semi-supervised block. High-confidence parts are those words with a confidence measure greater or equal than an estimated confidence threshold ($C_\tau$). The word-level confidence measure is based on the word posterior probabilities [20] computed over the recognition word graph.

It should be noted that the new built samples recognized samples can contain chopped words. Thus a new force alignment should be done; or, better, a constrained recognition using the user corrections as fixed constrains. Also, it must be considered that new words will be learned as the user enter corrections; So a huge phonetic dictionary and/or rules to transliterate words into phonemes must be available from the very beginning.

HMMs training is an expensive computational task. This is why re-training is delayed until enough utterances (a block) have been semi-supervised. Incremental training might help in reducing the computation time, so models could be updated more frequently, but recognition performance would be degraded.

LM training computation time is much faster. Thus, it may be updated more frequently than the HMMs, so unknown words would be learned sooner.

Our method must start with models built from a fully supervised corpus. Fortunately, the size of fully supervised corpus can be very small compared to the audio to semi-supervise: A small starting corpus will probably yield low recognition performance at the first steps, but it will improve faster, since more words will be asked for correction.

## 2. Estimation of the Transcription Error

The transcription error can be measured in terms of the well-known Word Error Rate (WER) [18]. WER accounts for the average number of elementary editing operations needed to transform a faulty text into the correct reference text. The WER ($W$) of a transcription is be defined as:

$$W = \frac{E}{N} \tag{2}$$

where $N$ is the total number of reference words. And $E$ is the sum of the edit cost of each utterance, calculated here as the Levenshtein distance with unitary costs compared to the corresponding reference.

On the formulae notation it should be noted that, all the variables presented in this section are assumed to be calculated from the very first transcribed utterance up to the one about to be supervised . Furthermore, those concerning to the unsupervised words, so their value is unknown, are distinguished with a minus superscript (-). And, analogously, those involving only the supervised parts are denoted with a plus superscript ($+$).

Assuming user corrections are flawless, then WER is only be due to the errors on the the unsupervised parts:

$$W^- = \frac{E^-}{N^+ + N^-} \tag{3}$$

where $N$ and $E$ has been decomposed into the contributions of the supervised ($+$) and unsupervised parts ($-$). Thus, $E^+$ has been assumed to be zero.

The exact values of $E^-$ and $N^-$ are indeed unknown during the process. A simple estimation can be done assuming an uniform distribution as on supervised counterparts:

$$\hat{N}^- = N^+ \frac{R^-}{R^+} \tag{4}$$

$$\hat{E}^{-} = E^{+}\frac{\hat{N}^{-}}{N^{+}} \tag{5}$$

where $R^{+}$ and $R^{-}$ are the number of recognized words which have been supervised and non-supervised, respectively, up to current utterance. And $E^{+}$ is the accumulated edition cost of the supervised parts (up to current utterance) before corrections are made.

However, a better estimation of $W^{-}$ can be achieved by ranking the recognized words into one of the $C$ groups depending on its confidence measure [13]. Let the groups from 1 to $C-1$ refer to the word with lowest confidence, second lowest, and so on respectively, in a utterance. And let Group $C$ refer to the, high-confidence, rest of words not in the previous groups. With this modification (4) and (5) are expressed as follows:

$$\hat{N}^{c-} = N^{c+}\frac{R^{c-}}{R^{c+}} \tag{6}$$

$$\hat{E}^{-} = \sum_{c}^{C} E^{c+}\frac{\hat{N}^{c-}}{N^{c+}} \tag{7}$$

Finally, WER of the unsupervised parts ($W^{-}$) can be obtained using the estimations (6) and (7) for the unknown WER of the unsupervised parts (3) :

$$\hat{W}^{-} = \frac{\sum_{c}^{C} E^{c+}\frac{R^{c-}}{R^{c+}}}{\sum_{c}^{C} N^{c+}\left(1+\frac{R^{c-}}{R^{c+}}\right)} \tag{8}$$

## 3. Confidence Measures ranking and classifier

As described before, the method needs to rank the recognized words depending on a confidence measure value ; and also for building new train samples using the confidence measure as the input of a binary classifier.

The classifier may be as sophisticated as desired. However, the recognition posterior probability of the words (post-max) alone is the most important feature [14]. A simple threshold classifier can then be built using the post-max. The optimal values for combination factor and the threshold are found , for instance, simply by maximizing the area under the ROC curve ([12]). Although recent publication achieve further performance ([11]).

## 4. Main contributions of our prodecure

In this section a brief remarks highlighting the difference with similar work published by D. Hakkani-Tur , and the work N. Serrano for handwriting transcription (HWR) in [13].

• Compared to "An Active Approach to Spoken Language Processing" by D. Hakkani-Tur in [4]:

- Their approach centers on manual supervision of whole utterances, thus the effort cannot be reduced so much.
- It cannot be properly implemented if the speech is not sentence-level segmented . While here the segmentation level has some influence, but this is optional.
- Their unsupervised learning for the LM is cleverer, but it has a slight impact since the greatest impact is related to the performance of the confidence measures .

- Their purpose is to ask for supervision the most informative utterances. But, in the end, their cost functions behaves much like the estimation of the WER presented here, but based on the single global confidence measure of the utterance. Thus, much less reliable.

• Compared to "Balancing error and supervision effort in interactive-predictive handwriting recognition" by N. Serrano in [13]:

- Our procedure can be considered a port of N. Serrano work for HWR. However, they implicitly worked like if the utterances were segmented at word-level . While, here, despite being also working in a word by word basis, calculations are made over the whole utterances under study. As a very convenient consequence, the system can jump over the high confidence rest of the utterance or even the whole utterance . See "Considering a Utterance as a word or as a sentence" in chap. 4 sec. 3
- The estimation of the error $\hat{W}^-$ (eq. 8) in their work is normalized using the estimation of the corresponding reference words of the unsupervised parts ($\hat{N}^-$ not depending on the confidence measure rank as used in the numerator. Previous experiments (not shown in this work) proved the current estimation of $\hat{W}^-$ is 0.5% lower in average than the older. In despite of this having little impact on the overall performance for low tolerated errors, is still a convenient improvement.
- Speech transcription poses different difficulties compared to HWR.
  The main issue was the word-level segmentation is more error prone: Thus during the interaction it is more likely the user would not understand the uttered words in the selection of the corresponding audio. This encouraged us to build a functional prototype in order to test the system.

CHAPTER 4

# PERFORMANCE OF THE INTERACTIVE SPEECH TRANSCRIPTION SYSTEM

In this chapter the proposed interactive system is evaluated. First, section 1 intends to be a general discussion about the possible aspects and features useful to properly evaluate an interactive system for transcription of the speech. Section 2 describes the task to be transcribed during the evaluations. And, finally, section 3 shows the obtained results.

## 1. Evaluation of the performance of an interactive transcription system

Here, the characteristics of the measures to be used in the results section to evaluate the system are detailed. It is clear that the key aspects to be evaluated of an interactive system like the proposed one should be: the decrease in the user effort and, the ability to learn and improve the underlying models and the quality of the resulting transcriptions:

**1.1. Assesment of the user effort.** An straight forward way to asses the user effort is by means of the number of times the user has been asked to supervise a word ("number of supervisions" NS for short) as in [13]. A bit more optimistic would be "the number of correct words corresponding to the parts which has been supervised" ("number of reference supervised words", #rS for short). The latter is preferred in the new results to be reported by the some authors. However, here, the first option is preferred, so the term "%Supervisions" will refer to the ratio of the number of recognized words to the number of words of the reference transcription. It can be argued that , in despite of that the definition allows for percentages greater than the 100% , the #rS may yield misleading results in other tasks: #rS might been motivated with the fact that the deletions require less effort than typing insertions, but for a task in which a user had to perform way more deletions than insertions, #rS would be lower than NS if the corrections would have been "accepting" (equal) operations instead. It is clear that effort would have been almost identical in both cases. And, even worse, yet a real effort in supervision would have existed , NSR counts will be null if the user would had to perform just deletions, . Anyway, #rS and NS in the conducted experiments,

Nevertheless, It would be desirable to have another features from which could be derived a more realistic estimation of the user effort: Let them be the performed number of keystrokes (NK) and the total time of played audio (LT). These features are encouraged by the that the time for manual transcriptions strongly depends on the duration of the speech and the time expended in typing the corrections.

However, it is left for future work the research on the connection of these features and the time a user really needs to fulfill the semi-supervision using the presented method. Moreover, It should be noted that in the case of an automatic evaluation of the system in that the is no real user, i some model must be assumed in order to account for the NK. In this work, it was assumed that the interface would allow to accept (equal operation) and delete with just one keystroke; Insertions and substitutions would require as many as keystrokes as the number of characters of the introduced corrections.

The ratio of the NK to the relative to the total number of characters in the reference, denoted as PK, is also meaningful . Additionally, this estimation can be compared to mPK : the minimum PK necessary if our system were flawless, i.e. if it would never ask to supervise correctly recognized words.

**1.2. Assesment of the improvement of the ASR system due to the interaction.** The WER will be used to asses the performance of the recognitions. This is expected to strongly depend on the tolerance error, since the more supervisions, the better will the models be re-trained. Nonetheless, some improvement of the ASR models could result even when no supervisions are done when the tolerance is too high. This is because of the unsupervised learning , this is, increasing the training sets with the high-confidence parts of the automatic transcriptions. Although, in general, this could lead to worse models, it has been proved that unsupervised learning improves the AM for the WSJ when using an external LM.

Moreover, in order to compare the contributions to the improvement of the acoustic and language parts , experiments using a fixed external LM or AM can be conducted , as in [4].

**1.3. Assesment of the global performance of the method.** In order to depict the overall performance of our interactive system, the WER resulting from the semi-supervision of the recognized blocks versus the percentage of supervisions (PS), explained above, will be used. This is because PS strongly depends on the method (so on the tolerance $W^*$) , while it is not so dependent on the task . Conversely, in despite of providing a more fair picture of the real effort, PK along with LT depend absolutely on the task. Additionally, these results are to be compared with the corresponding "Fully Supervised" experiments. These experiment consists in fully transcribing a certain number of blocks, and then accepting the output of the ASR of the rest of the speech. The effort for these transcriptions is accounted as just the PS, and the residual WER as the WER resulting of the recognized blocks. This serves to compare a non-guided interactive system with ours.

Moreover, it should be highlighted that an ideal interactive system would avoid the supervision of correctly recognized words. Thus, the less equal operations a user would perform, the better. For our system, too many equal operations would mean that the confidence measure cannot successfully locate the errors. In the latter case, the method will ask for to many supervisions, yielding to a residual final WER better than requested. This behavior is preferable compared to a method that may end with errors higher than tolerated.

Thus, it is also convenient to show the behavior of the confidence measures along the whole process of semi-supervision. The performance of the confidence measures is usually described in term of the Classification Error Ratio (CER). Here, however, it is further more relevant the comparison or the difference between the real and the estimated edit cost of the words in each confidence measure rank. Let them be denoted by $E_c$ and $\hat{E}_c$ respectively. Obviously, $\hat{E}_c = \hat{E}_c^+ + \hat{E}^-$, as it can be derived from formulae in chapter 2, . Where $\hat{E}^-$, as formulated in equation 7, is the main driver of the method. And, it is also interesting the ratio of supervisions which consisted of equal operations (NE) to the total number of supervisions (NS). Let this ratio be denoted by PE.

## 2. Experimental set-up

Our method was evaluated for the task of transcribing the whole WSJ0 training set, as well as the WSJ0+1 training sets of the Wall Street Journal (WSJ) speech database [7].

Three different scenarios were studied: 1) Both, the AM and the LM, were trained incrementally as explained in chapter . From now on, this scenario will be referred as as incremental LM or "LM inc". 2) Only the AM was incrementally trained, while the LM was the external WSJ 3-gram LM $5k$ (4.986) word closed vocabulary (LM 5K). And 3) for the External WSJ 3-gram LM $20k$ (19.979) word open vocabulary (LM 20K). The fixed LM remained unchanged for the entire transcription process.

The AM models learned using the fixed LM can be used to recognize the benchmark tests of the Nov'92 ARPA evaluations [7], in order to assess the performance of the trained HMMs . Nonetheless, it should be noted that the fixed LM experimental setups are not only motivated for the sake of the comparison to the WSJ benchmarks. It is also motivated because, in a real case scenario, it is easy to find enough data such as being able to train a fixed LM good enough along the completion of the task. While this is not true for the data needed to build a good AM.

The overall characteristics of the WSJ speech database are shown in Table 2. In addition, the overall difficulty of train and test sets in terms of the perplexity is shown in Table 2 for both language models (LM-5k and LM-20k)

TABLE 1. WSJ speech database statistics.

| | WSJ0 | WSJ1 | nov'93 Hub-1 | Nov'92-5k | Nov'92-20k |
|---|---|---|---|---|---|
| Time (h.) | 15 | 66 | 0.4 | 0.7 | 0.7 |
| Utterances | $7k$ | $30k$ | 213 | 330 | 333 |
| Words per utterance | $18 \pm 8$ | $17 \pm 7$ | $17 \pm 8$ | 16±6 | 16±6 |
| Speakers | 84 | 200 | 10 | 8 | 8 |
| Vocabulary size | $8k$ | $13k$ | $1k$ | $5.4k$ | $5.6k$ |
| Running words | $129k$ | $510k$ | $3k$ | $5.4k$ | $5.6k$ |

TABLE 2. Perplexity on train and test sets depending on the LM.

| Exteral LM | WSJ0 Train | nov'93 Hub-1 | nov'92-5k | nov'92-20k |
|---|---|---|---|---|
| 3-gram LM-5K | – | 115 | 53 | – |
| 3-gram LM-20K | 146 | 170 | – | 142 |

### 2.1. Initialization.

• Segementation of the corpus into blocks

The WSJ0 training set was split into 12 blocks, each one containing utterances from 7 different speakers. Acoustic models were completely re-trained every time the semi-supervision of a block was fulfilled. These Hidden Markov Models (HMMs) consisted of clustered word-internal 3-state triphones with a number of 16 Gaussian-mixtures per state.

For the larger task WSJ1+0, the WSJ1 train set was split into 29 blocks, and then the 12 blocks from the WSJ0 were added. In this case, HMMs were trained with 24 Gaussian-mixtures per state.

The average duration of each block in WSJ0 train set was about one hour, and two hours for the WSJ1 train set.

- Explored Tolerance Errors

Several tolerance values were tested from $W^* = 1\%$ to 50%. On this corpus, these values correspond to roughly allow from one error every 3 utterances ($W^* = 1\%$) up to 9 errors per utterance ($W^* = 50\%$). Additionally, as a comparison baseline, $W^* = 0\%$, which is equivalent to perform a full supervision was also tested.

- Initial models

The initial HMMs, LM, optimal parameters for the decoder, confidence threshold and parameters for $\hat{W}^{\text{semi}}$ (i.e. $\{E_r^{c+}\}$, $\{R_r^{c+}\}$, $\{R_r^{c-}\}$, $\{N_r^{c+}\}$ for $c = 1..4$ ) , were obtained from the initially fully supervised part of the task.

This fully transcribed part is split into 2 blocks, named 0 and 1. For experiments with incremental learning of the ASR models, a pre-initial ASR system is trained with block 0. Then, the block 1 is used as a development set in order to optimize the parameters above. After that, the a new initial HMMs and LM were trained using the whole block. However, For experiments using an external LM or AM will directly use the block 1 for the optimization of the parameters.

Preliminary HMMs are needed to built the initial ASR system. Also, certain ASR parameters have to be properly estimated. With this purpose, the first block was considered as fully transcribed manually and used to train initial HMMs and optimize ASR parameters. In addition, the optimal confidence threshold ($C_\tau$) to select the high-confident parts was estimated by minimizing the confidence error rate (CER) [20] in this block.

On the other hand, the balancing method updates the values of its parameters after each user interaction. Nevertheless, it still requires a good initialization ($\{\hat{E}_0^{c+}\}$, $\{N_0^{c+}\}$, $\{R_0^{c+}\}$, $\{R_0^{c-}\}$) in order to perform properly from the very beginning. These initial values were those resulting of applying the method itself on the first block.

**2.2. Model training.** As explained before, an ASR system depends on an acoustic and a language model (AM, and LM respectively): The AM consisted of Gaussian-mixture HMMs of clustered tri-phonemes with no cross-words. The number of mixtures varied depending on the corpus sets to be transcribed: For the experiments involving just WSJ0 sets, the number of mixtures was 16. For WSJ1 or WSJ1+0 sets, it was 24. which were trained with *HTK* tool-kit [21]. While the LM consisted of 3-ngram models smoothed with interpolation using the Knesser-Ney discount , trained using the *SRILM* toolkit [15].

It should be noted that although this kind of ASR training has been the facto standard over the last decades, it has been recently superseded by other techniques. However, for the purpose of the evaluation of our interactive system, there is no need of achieving cutting-edge performance for the recognition.

Experiments conducted in the fist subsection of the results section above, used the *iAtros* tool-kit [5] for the recognition. However, other experiments were performed using the last version of the *ak* recognizer [3]. The latter achieves better performance and lower computation times for the same level of pruning than iAtros, , however, it became available to us later. Consequently, only the results in the second subsection in the results section have been re-evaluated using *ak* . Nevertheless, the behavior of the system is quite independent of the recognizer and of their performance.

**2.3. Interaction: User simulation.** For the sake of speed, corrections were performed automatically by means of a simulation of a real user:

It was assumed that a real user can always flawlessly understand and transcribe any speech in the audio; although a couple units of WER is usually found in manual transcriptions . This way, the simulation consisted in guessing the edit path a user would apply. Moreover, the number of edit operations a trained user performs was assumed to be always minimum.

Then, two different possibilities were discussed about how best mimicking the the operations a real user would do:
• Straight matching using the temporary alignment: The matching would be limited to reference words spoken in the same temporal range as the recognized word.
• Matching using the edit distance: The matching would be found using the Levenstein algorithm. Insertions, however, should be found over the the time range, as in the option before. It should be noted that this can yield to mismatched substrings in rare cases; Also, the edit distance is not properly defined for sub-strings since several sub-strings from the reference can match a recognized word; And that the way the matching is done varies the set of edit operations (edit path). Nevertheless, in practice, different users will likely perform identical corrections since they benefit from grasping other information that the system does not, as phonetic similarities, semantic and syntactic restrictions, and, mainly, the place of words in time.

The first option may seem more fair and natural. The second, however, may yields optimistic evaluation results. However, as the implementation of the prototype interface (next chapter) allows the introduction of corrections in a utterance not necessarily corresponding to the asked word, the second option is then more realistic in our case.

In any case, the simulation required a whole alignment of the text references to the audio was needed from the beginning. This alignment was carried out by training an ASR model with the full corpus, and then making a forced recognition.

Moreover, in order to implement the correction convention suggested to the users (see sect. 1.3), selected reference words for matching were those which their middle time point laid inside the temporal range of the recognized word (with no audio margins). Then, the selection of the edit path followed no other criterion than choosing the first one returned by the actual implementation of the Levenstein algorithm. Despite the commented issue, this barely biased the experiment since, in almost all cases, there was just one possible edit path. This was because, thanks to the time restrictions, the matched sub-strings contained up to three words at most.

Finally, in case of no recognized words in the line, full insertion of the reference text of the line was performed.

**2.4. Unsupervised Learning.** A brief explanation can be found on the chapter 2, section 2.

## 3. Results

First, in subsection 3.1, it is presented the results of transcribing the smaller task, WSJ0 train set, split in 12 blocks. Several tolerance values were tested from $W^* = 1\%$ to 60% . Then, more detailed results are depicted for the task of transcribing first the WSJ1 plus the WSJ0 train sets, split into 29 and 12 blocks respectively. The results of two representative thresholds $W^* = 6\%$ to 12% are shown. All the recognitions in this subsection were performed using the *iAtros* toolkit using the incrementally trained HMMs of 16 and 24 gaussian mixtures, for the WSJ0 and WSJ1+0 respectively. The parameters for the recognizer and the confidence measure classifier were optimized for the first block, and then never updated. The number of confidence measure ranks was 4. The threshold classifier $C_{Th}$ was optimized in the first block but never updated. The decisions on the selected parameters are justified in the comparison in the following subsection.

In the following subsection, the effect of different parameters of the method on the global performance are exemplified by means of the performance results on the smaller task of WSJ0. The necessary experiments were conducted in order to assess the following features: "Initialization of the $W^-$ parameters" , "The number of confidence measure ranks in a utterance" , "Considering a Utterance as a word or as a sentence" , "Number of blocks" and "Updating ASR and Threshold Classifier Parameters". The recognitions were performed using the *ak* toolkit , but for the last comparison. The task was split into 6 blocks instead of 12, but for "Number of blocks" and "Updating ASR and Threshold Classifier Parameters" comparisons. Several tolerance values were tested from $W^* = 1\%$ to 60%.

### 3.1. Analysis of the performance in transcribeing the WSJ train sets

. • Transcription of the WSJ0 train set

For clarity, only the results of the supervision effort and residual accumulated WER after the transcription of each block of WSJ0 training data using a tolerance error of $W^* = 20\%$ and 5% are depicted in Fig. 1. The rest of the lines corresponding to other tolerance values had an identical tendency to these one, from the fully supervised task ($W^* = 0\%$) to the almost non supervised experiment ($W^* = 50\%$).

The figure shows that all the experiments behaved in a similar manner along the process. However, for LM-20k the reduction in user effort was greater. This was because LM-20k yields better recognition accuracy than LM-5k in WSJ0 training data (fig.fig:WSJ0-WER ). The incremental LM obtained intermediate results, but had a stepper rate of improvement. However, which recognition performance is better is completely dependent on this corpus and the LM: The fixed ones were estimated from a training os million of sentences. The incremental LM is from less than 1/100 that size, and the train data is pronned with errors. But, even so, the incremental is better adapted to task since the tokens (like dots, colons, etc.) were verbalized. Thus, it greatly differs from the regular english text from which were estimated the fixed LM. In the end, both effects seems to compensate.

It is also important to note, that the stated before is not true for the very high tolerated errors. In these cases , almost no semi-supervision was asked to the user. Thus, the ASR models only changed because of the unsupervised learning. The incremental LM yielded catastrophic results. This is the usual when re-training a system with its own output, even the confidence measures prevents some of the worst recognized samples to be added to the new training set. However, the fixed LM the initial recognition WER surprisingly improved block after block, and the absolute difference with the results from the low $W^*$, or even the baseline, was considerably small. Thus, it can be stated that it is desirable stating the method with a huge initial external LM. Which, in turn, might be further adapted using the incremental training.

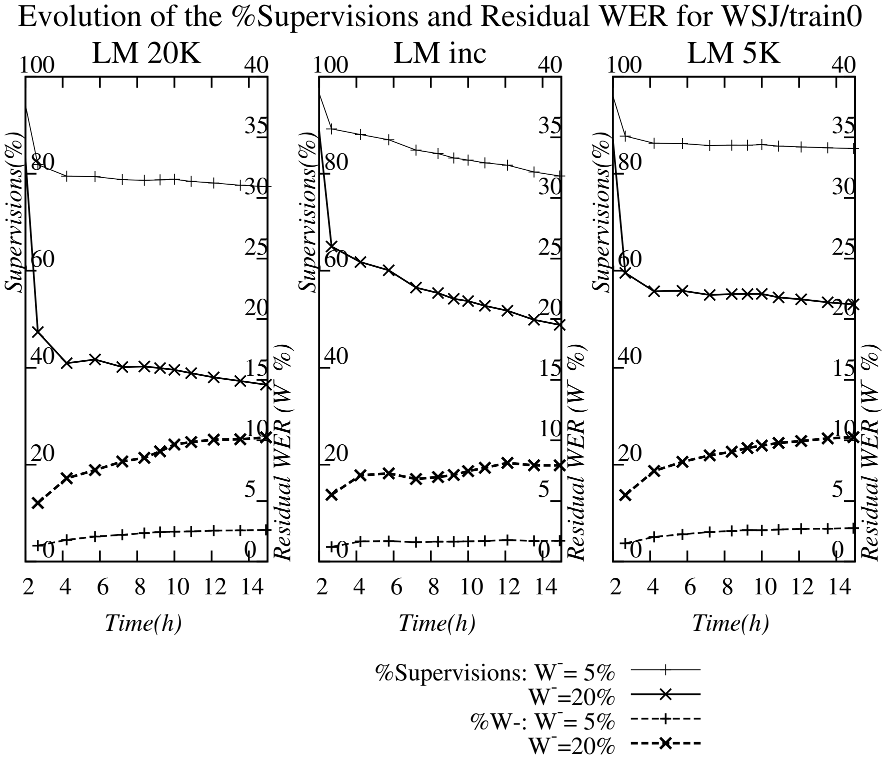


FIGURE 1. User effort and final quality of the transcription of WSJ0 training data using LM-5k, LM-20k, or incremental LM: - Reduction of the user effort in terms of the number of supervisions relative to the recognized words in a block (at top). - Residual accumulated WER after semi-supervision of each one of the blocks (at bottom).

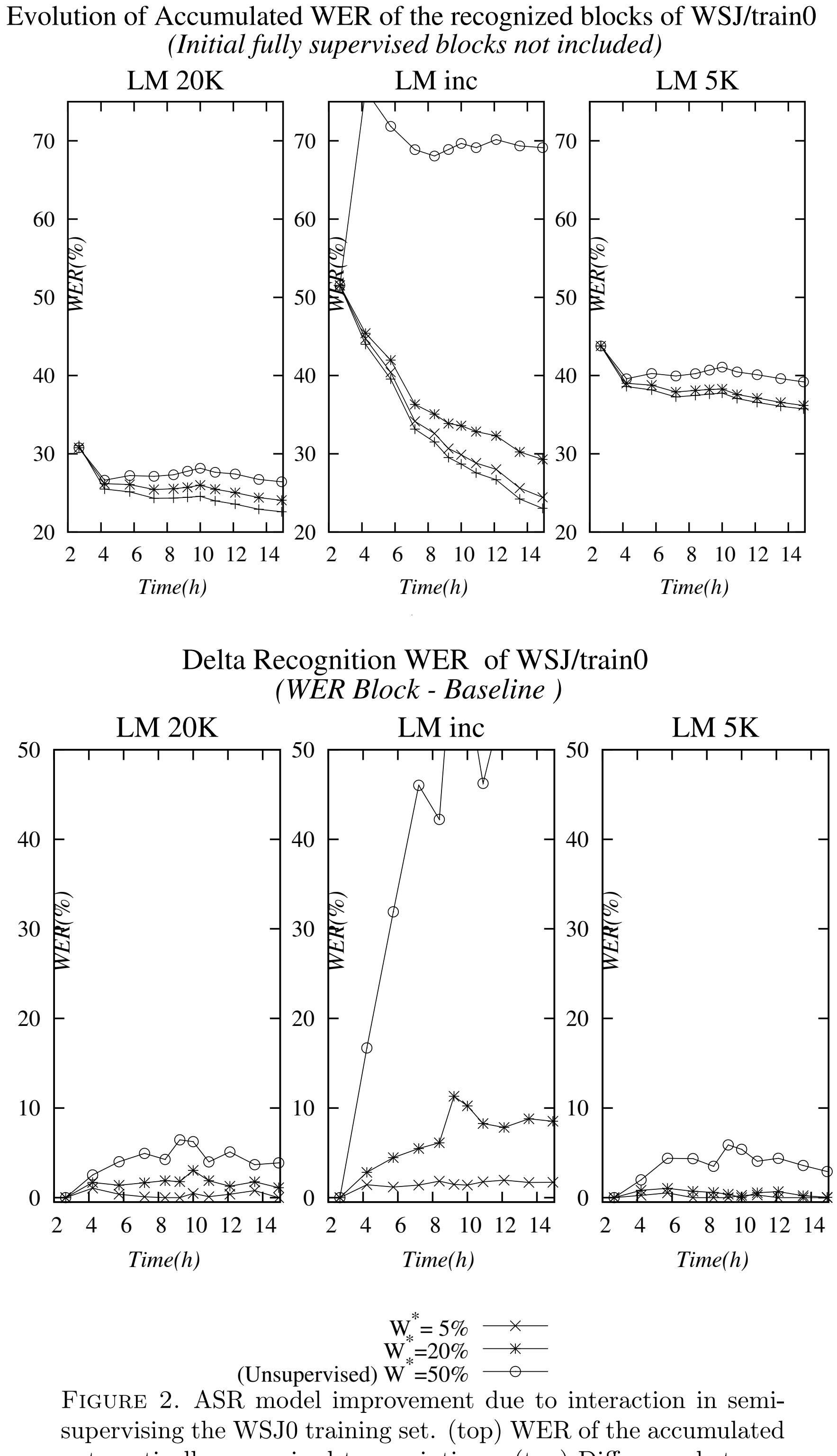


FIGURE 2. ASR model improvement due to interaction in semi-supervising the WSJ0 training set. (top) WER of the accumulated automatically recognized transcriptions. (top) Difference between the WER of each recognized block and its corrsponding baseline ($W^* = 0\%$)

As supplement, Table 3 summarizes results for some significant tolerance errors after the completion of the whole task: the relative number of supervisions (%Sup.); the final residual WER of semi-supervised WSJ0 training set (Wtrain), and the WER on the corresponding external test using the final resulting ASR system (Wtest). Thus, as expected, the more allowed error, the less user interaction was needed.

Table 3. Final results for different $W^*$ tolerances. %Sup. is the relative number of supervisions; Wtrain is the residual WER of the semi-supervised WSJ0 training set; and Wtest is the WER of the corresponding external test using the final resulting ASR system.

| | LM-5k | | | LM-20k | | |
|---|---|---|---|---|---|---|
| $W^*$ | %Sup. | Wtrain | Wtest | %Sup. | Wtrain | Wtest |
| 0 A. Sanchis | 100.0 | - | 6.1 | 100.0 | - | 11.57 |
| 0 | 100.0 | 0.00 | 6.6 | 100.0 | 0.0 | 14.6 |
| 5 | 85.2 | 3.1 | 6.6 | 76.0 | 2.9 | 14.6 |
| 10 | 72.5 | 6.2 | 7.0 | 59.5 | 5.8 | 14.8 |
| 20 | 53.1 | 10.3 | 7.6 | 36.1 | 11.4 | 16.1 |
| 50 | 0.00 | 39.2 | 10.1 | 0.0 | 26.4 | 18.7 |

The resulting quality in the transcriptions after semi-supervision is depicted in terms of the WER in Table 3 (column Wtrain). All experiments resulted in much better transcriptions than requested (WER close to the half to the requested $W^*$ for all experiments). Thus, the system behaved in pessimistic way by requiring more user effort than a flawless predictor would. However, this is preferable rather than systems yielding poorer results than requested.

It should be highlighted that for a moderate error tolerance (20%), a great decrease in the user effort is achieved for both benchmarks (36% of supervisions for LM-20k and 53% for LM-5k). While, final output resulted of 10% of WER (less than 2 words per utterance) for the task of transcribing the whole WSJ0 training set.

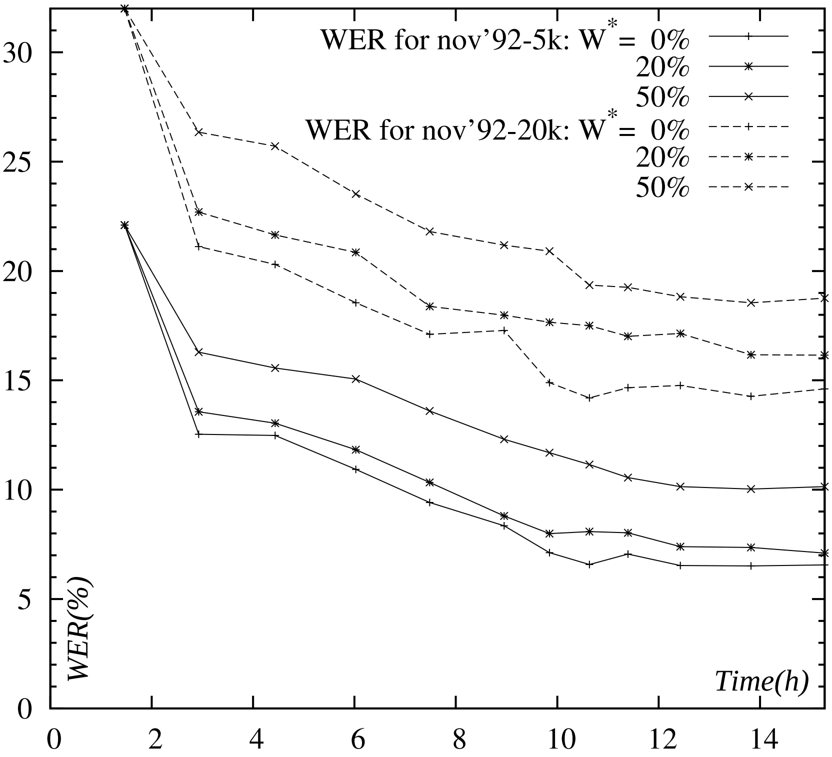


Figure 3. Improvement of the HMMs in terms of the WER of an external test for tolerances $W^* = 0\%, 20\%$ and 50%. Recognition of Nov'92-5k test using the HMMs learnt during semi-supervision of WSJ0 using LM-5k (Solid lines). Recognition of Nov'92-20k, using models learnt with LM-20k (Dashed lines).

Additionally, the improving performance of the trained HMMs for the fixed LM experiments after each step is shown in Fig. 3 in terms of the WER yielded by the recognition of the the Nov'92 benchmark tests. Here, results for 0%, 20% and 50% have been plotted. (See column "Wtest" in Table 3 for the WER results of other tolerance thresholds. Tolerances between 20% and 0% did follow the same tendency. Then, it can be stated that for low tolerances, the method results in HMMs almost indistinguishable to the model built using the full WSJ0 training set for both benchmarks.

It should be noted that, again, the $W^* = 50\%$ experiments also showed a progressive improvement. This is still consistent with the mentioned fact that no user interaction was required for this tolerance, because the models are still improved using the high confidence unsupervised parts, and also thanks to that the fixed LM suited for these tests.

- Task: WSJ1+0

Here, more aspects of the performance of the system are to be detailed:

Figure 4 (top) shows the decrease in the number of supervisions, calculated as explained before in section 1. This estimation can be compared to the minimum keystrokes needed for an ideal system which would flawless ask just for the minimum necessary corrections. Addionally, the time expended in playing one time each of the asked words in units relative to the processed audio length is plotted at the bottom of the figure. Nevertheless, the average number of necessary listenings depends on the user.

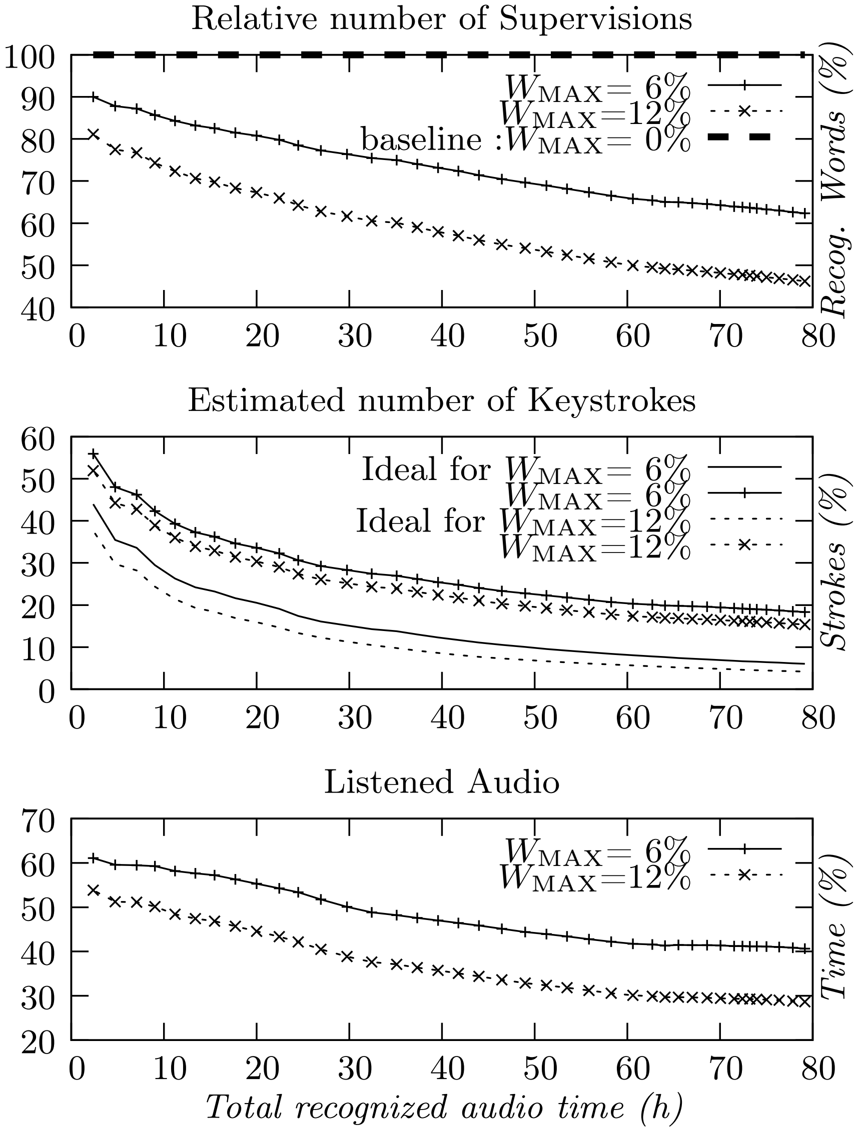


FIGURE 4. Reduction in user effort

Next, the WER values resulting from the concatenation of all already recognized blocks before the semi-supervision are presented. This way , it is to easier the visualization of the trend since each block has a different difficulty level for the recognition. Increasing differences throughout the task between the baseline and experiments on 5 (top), might be due to the accumulation of errors, but also to a possible progressive degrading of the system: On figure 5 (bottom) increments from the WER of each isolated block to the corresponding baseline may help to visualize the system is improving or not.

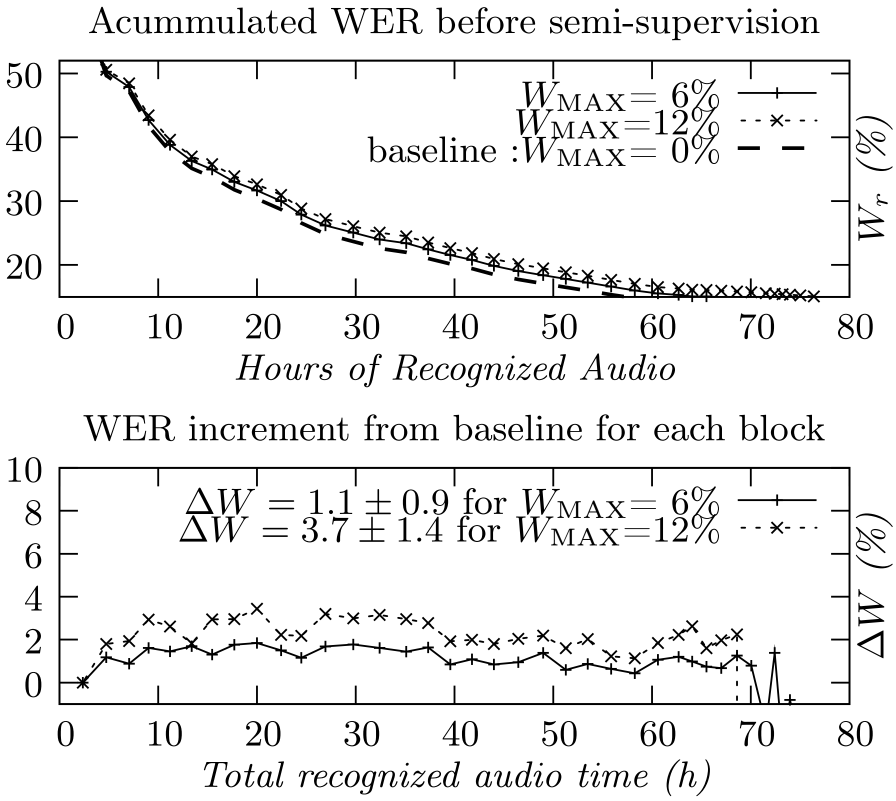


FIGURE 5. Improvement of the ASR system due to interaction

Recognition performance of the ASR models of the different experiments is assessed in figure 5 (top).

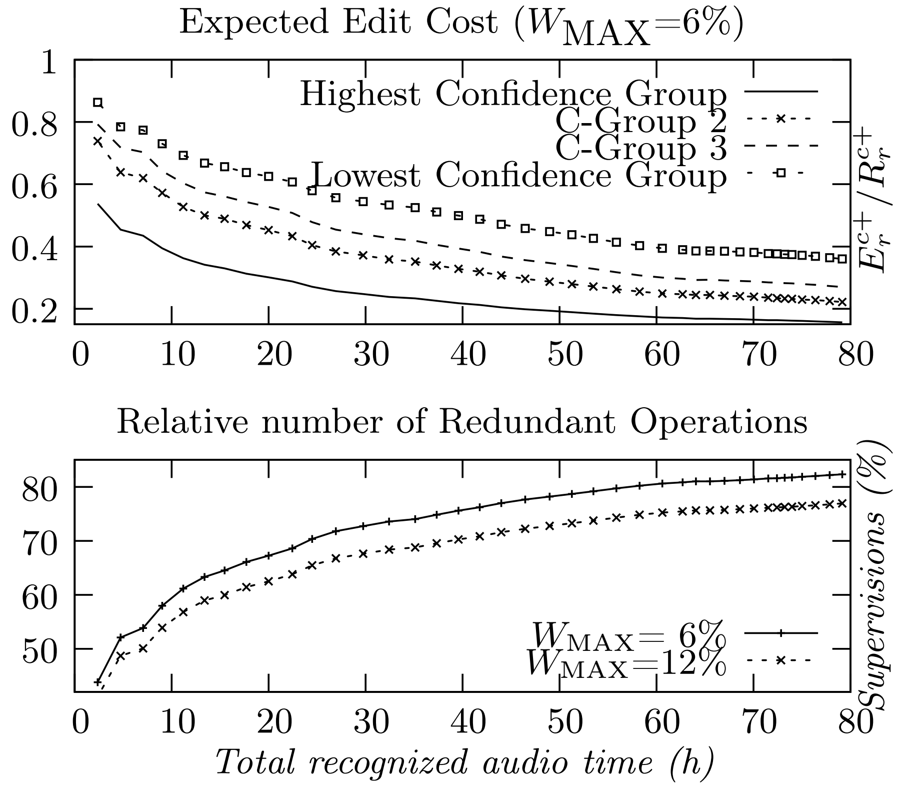
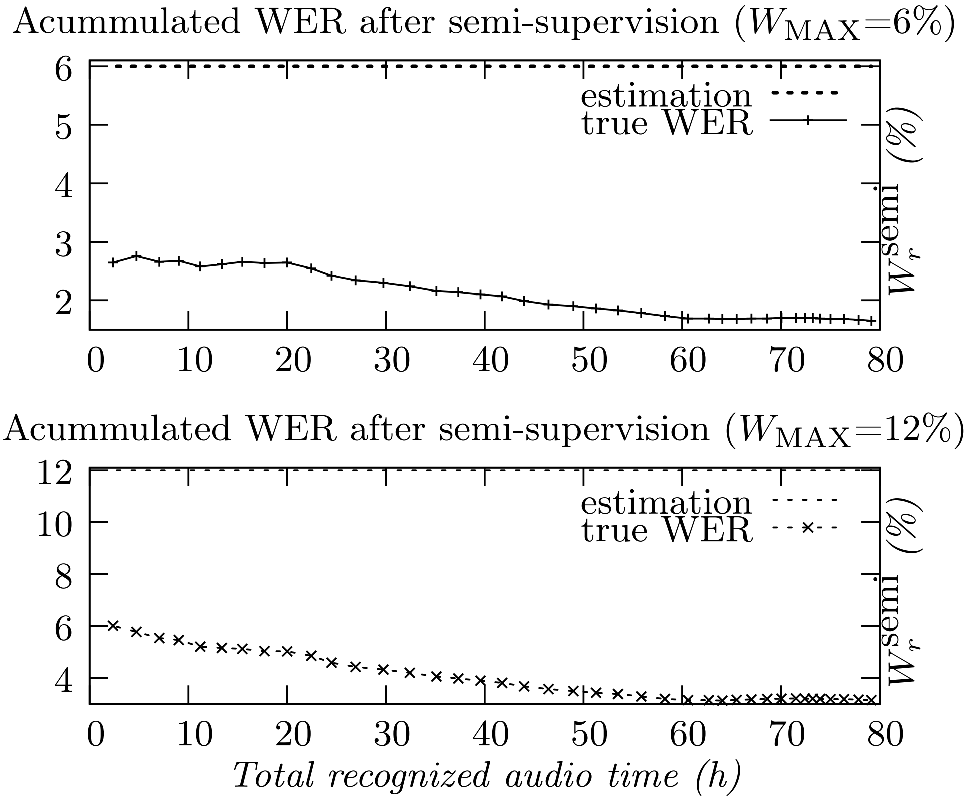


FIGURE 6. *Confidence measure classifier performance*

Figure 6 (top) plots the ratio of total edit distance performed by the user and the number of supervised words for each confidence group. Trends were similar for both experiments, so the plot only shows $W^* = 0.06$ for clearness. A good classifier should yield a zero edit cost for words with high-confidence value, while more than 1 (100%) cost for the lowest. Thus, a perfect classifier should ask for no correctly recognized words (equal operations). The number of supervisions which resulted in equal edit operations are shown in Figure 6 (bottom).


Acummulated WER after semi-supervision ($W_{\mathrm{MAX}}$=6%)

Acummulated WER after semi-supervision ($W_{\mathrm{MAX}}$=12%)

FIGURE 7. Global WER performance of the method

Figure 7 shows WER of accumulated final transcriptions after semi-supervision. It can be seen that the improving tendency which happened for the small WSJ0 does also holds for the longer task WSJ1. Moreover, the system achieves in adapting to the change from the WSJ1 train set to the WSJ0, which happens around the 65 on the figures.

Tables 4 to 6 show some results compared to the baseline, for the experiments on the last two audio blocks: Table 4 shows results on block 41. Let us recall that blocks from 1 to 41 contained samples from the WSJ92-93 train partition. In particular, block 41 was built up from the last 607 samples in the WSJ'92 train partition. Which, in turn, is the same as the last block in the WS0 experiments above. Although not explicitly showed, the results for this same block using the models learned incrementally with only 16 HMMs gaussian mixtures, yielded worse results as expected: the baseline was 11.81, 14.09 for $W^* = 5\%$, etc.

TABLE 4. *Semi-supervision on the WSJ1+0 block 41 ( WSJ0 block 12)*

| | Baseline | $W^* = 6\%$ | $W^* = 12\%$ |
|---|---|---|---|
| Supervisions | 10863 | 5781 | 3785 |
| Equal Ops. | 10570 | 5436 | 3440 |
| WER | 4.76 | 6.76 | 9.10 |
| WER$^{\mathrm{semi}}$ | 0.00 | 3.11 | 6.19 |

TABLE 5. *Semi-supervision on block 42 (nov'931-Hub1 test)*

| | Baseline | $W^* = 6\%$ | $W^* = 12\%$ |
|---|---|---|---|
| Supervisions | 3174 | 3113 | 3026 |
| Equal Ops. | 2227 | 2121 | 2063 |
| WER | 37.46 | 36.30 | 36.59 |
| $\text{WER}^{\text{semi}}$ | 0.00 | 3.10 | 6.16 |

Table 5 shows results on block 42. This block corresponded exactly to the WSJ'92-93 Hub-1 test partition. Speech and vocabulary in this block differs slightly from the train. For instance, in the train partition, punctuation symbols are verbalized (ex. '(' is spelt as 'open-paren'). While, this does not happen in test, resulting in a more natural speech.

TABLE 6. *Semi-supervision using external LM 20K on block 42 (nov'931-Hub1 test)*

| | Baseline | $W^* = 6\%$ | $W^* = 12\%$ |
|---|---|---|---|
| Supervisions | 3223 | 2611 | 1919 |
| Equal Ops. | 2631 | 2005 | 1437 |
| WER | 18.99 | 20.35 | 21.09 |
| WER A.Sanchis | 16.1 | - | - |
| $\text{WER}^{\text{semi}}$ | 0.00 | 3.11 | 6.19 |

Table 6 shows results on block 42, but this time an external LM was used for recognition of this last block, instead of the learnt during the semi-supervision. All previous steps of training, semi-supervision and recognition were exactly the same as in table 5. The external 20k-words 3-gram LM was built up from a text supplied along the WSJ speech corpus, which is precisely intended for recognition on Hub-1 test. Additionally, line "WER A.Sanchis", shows the published stated-of-the-art results using iAtros. The differences are due because of the higher pruning for the decoding and the lower number of gaussian mixtures used here.

In summary,

- User effort is greatly reduced as it learns (fig. 4).
- Recognition performance is just slightly worse than the best possible (fig. 5 top).
- WER difference of each recognition between an experiments and the baseline is maintained in average (fig. 5 bottom). This means that the slight worsening in the accumulated WER throughout the task, is due only to the accumulation of errors, not a degrade in the ASR quality.
- Word classifier performance behaves as expected, but its performance degrades over time (fig. 4 top). As a consequence, the number of redundant supervisions increases (fig. 4 bottom).
- $\hat{W}^{\text{semi}}$ insignificantly varies around requested $W^*$.
- $\hat{W}^{\text{semi}}$ is always pessimistic (which is desirable): resulting transcriptions are always better than expected.
- The system effectively adapts, in terms of the number of supervisions, to the true, unknown, quality of recognition: On block 41 (table 4), 50% to 68% of the recognized words were supervised. Next block (table 5), which was poorly recognized, 98% of the words. While, only 60% to 81% when an external LM improved the recognition (Table 5).

**3.2. Analysis of the impact of the parameters of the method.** All experiments started with ASR models trained using the blocks 0 plus 1. However, the impact on the WER and PS (%supervisions) of these fully supervised blocks has been removed from the presented results. By so, eliminating the smoothing effect on the results. This way, the differences amongst the conducted experiments can be spotlighted clearer.

• Initialization of the $W^-$ parameters

The initialization of $\{\hat{E}_0^{c+}\}$, $\{N_0^{c+}\}$, $\{R_0^{c+}\}$, $\{R_0^{c-}\}$ played a key role. This is because an increased number of unsupervised words $R_c^-$ may not increase $W^-$. In fact, for $R_c^- \to \infty$, $W^-$ will tend to the WER of the supervised parts. This issue has several side effects. For instance, if $W^-$ is lower enough than the requested $W^*$, it is likely that no more utterances will be supervised for the remaining speech. This behavior is only desirable when $W^*$ is much higher than the real recognition WER, and it will be still a problem if the nature of the speech would suddenly change.

In order to deal with this issue, the initial values of these parameters were so that $W^-$ was initially about 100%. Then, the first block was used for tuning this initial values, as if the user would have supervised every word in the first block. In order to prevent overtraining on this initial block, or that this block would have no impact on the estimator, the initial values of $R_c^+$ and $N_c^+$ were about the half numbers of words in the first block.

Despite this initialization seem reasonable, the initial $R_c^+$ have some impact on the experiments. Fortunately, the lower $W^*$ , the higher number of supervisions were necessary, so the lower impact on the performance.

• The number of confidence measure ranks in a utterance

As explained before, the estimation of the WER of the unsupervised parts, $W^-$, has been formulated to depend on the confidence measure relative to the other confidence measures values in a utterance. To do so, words are ranked form the lowest confidence value to the highest, and they are assigned a rank value. Let $c = 1..C$ be the number of ranks. If so, then the highest rank $c = C$ includes all the highest confidence values not in the precedent ranks.

It should be spotted that the nature of this ranking will overrate the expected edit cost of the words in the highest rank compared to the the real one. This overrated estimation is due to the lower confidence words in the highest ranked confidence group, because they are more likely to be asked for supervision and been wrong. To overcome this, an obvious manner seems to increase the number of ranks, or, even, to unbound the number of ranks. Unfortunately, for a high number of ranks, the estimation of the edit cost associated to the highest ranks is nether reliable because not all the utterances have so many words, and they are unlikely to be supervised.

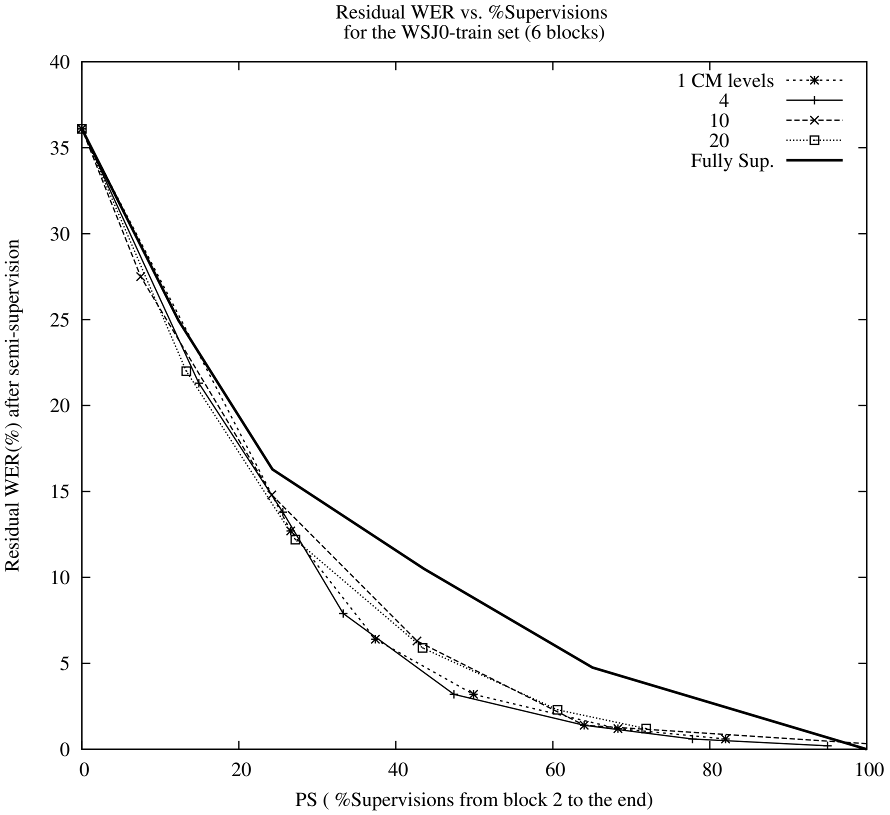


FIGURE 8. Overal performance of the method for several $W^*$. Several Number of CM ranks are compared against the baseline.

From figure 8, it can be stated that for this corpus the number of CM ranks level have little impact. Only for percentage of supervisions (PS) from 20% to 30% (corresponding for $W^*$ from 20% to 30%) there was a clear optimal around the 4 ranks per utterance.

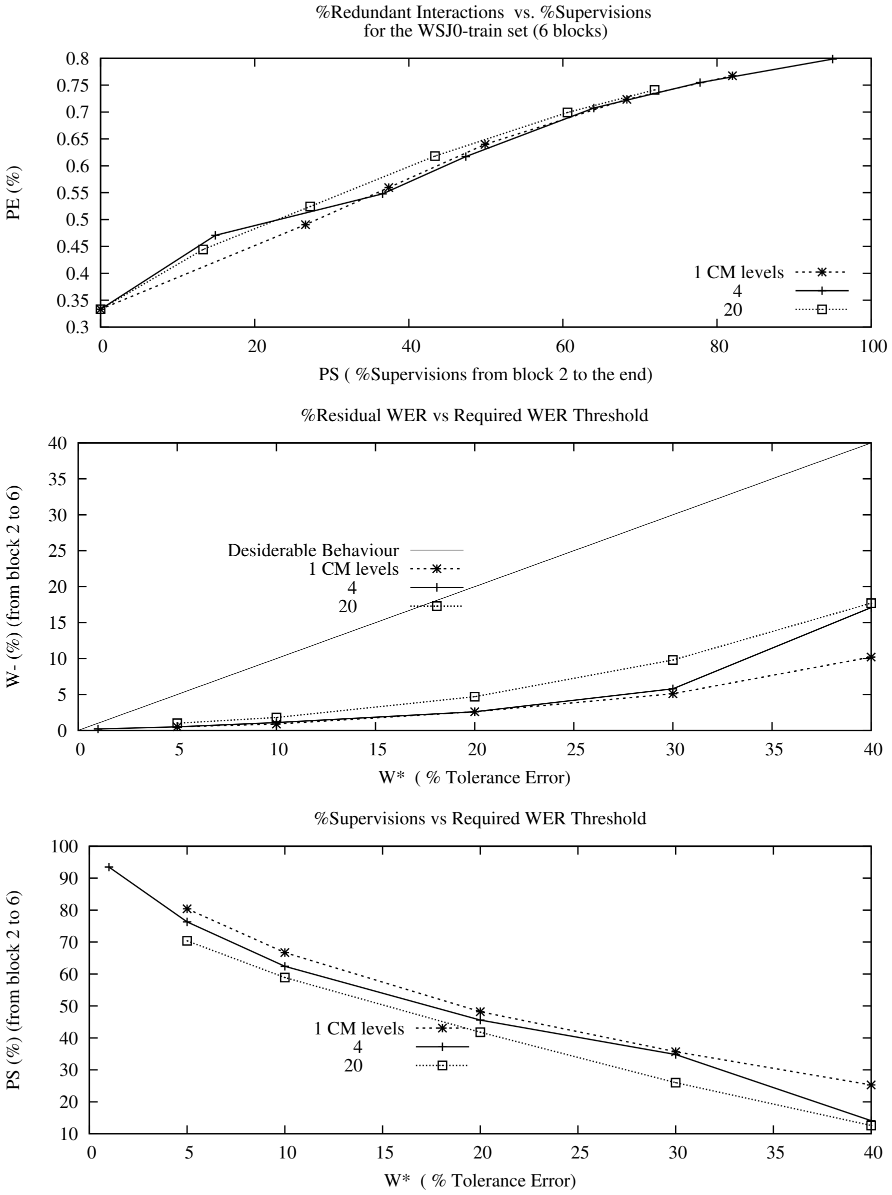


FIGURE 9. Comparision of experiments with different number of CM ranks.

In figure 9 all the experiments perform similarly for the percentage of equal operations. However, the number of supervisions was lower for the higher number of ranks. As a consequence, the residual WER are higher. But since all are well below the required threshold, that a better estimation of WER comes with more ranks.

In table 7 is shown the relative difference between the expected and the real Edit Costs per rank : $\frac{\hat{E}_c^- - E_c^-}{E_c^-}$ . Consistently, the more ranks, the lower factor for the highest confidence rank (exact estimation is factor 1) .

Table 7. Relative difference between the expected and the real Edit Costs per rank of confidence in a utterance

20 CM ranks:

| $W^*$ | Lowest | 2 | 3 | 4 | 5 | 6 | ... | 17 | 18 | 19 | Highest |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 5% | 1.5 | 0.8 | 0.78 | 0.71 | 0.73 | 0.92 | ... | 7.7 | 8.31 | 8.87 | 38.29 |
| 30% | 0.36 | 0.45 | 0.83 | 1.45 | 3.26 | 6 | ... | 3.77 | 4.2 | 3.44 | 10.39 |

4 CM ranks:

| $W^*$ | Lowest | 2 | 3 | Highest |
|---|---|---|---|---|
| 5% | 0.11 | 0.09 | 0.1 | 99.68 |
| 30% | 0.4 | 0.55 | 0.79 | 98.24 |

Nevertheless, for the rest of experiments 4 ranks will be used since it yielded better in 8. While the performance in terms of the user effort from the 20 ranks is very small.

• Considering a Utterance as a word or as a sentence

In the previous chapter was remarked that our interactive method can work regardless of what a utterance is considered. For the WSJ the natural subsegmentation of the blocks are the sentences. However, it might be considered that an utterance is an only word, as it is published for handwritten recognition.

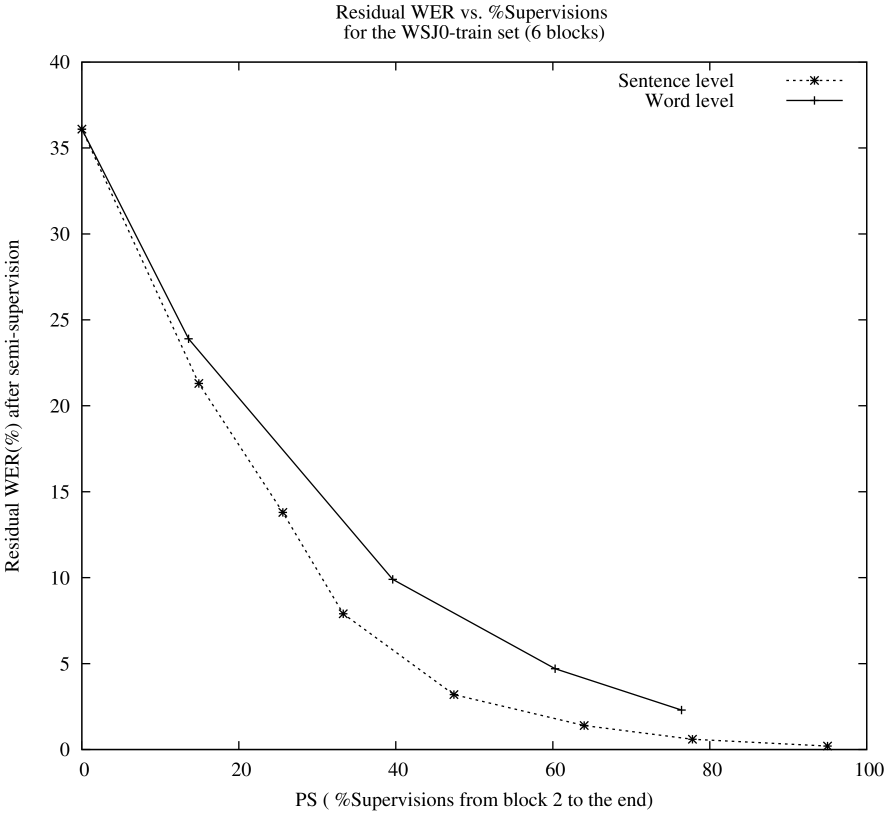


Figure 10. Overal performance of the method for several $W^*$. The utterances of words and samples are compared against the baseline.

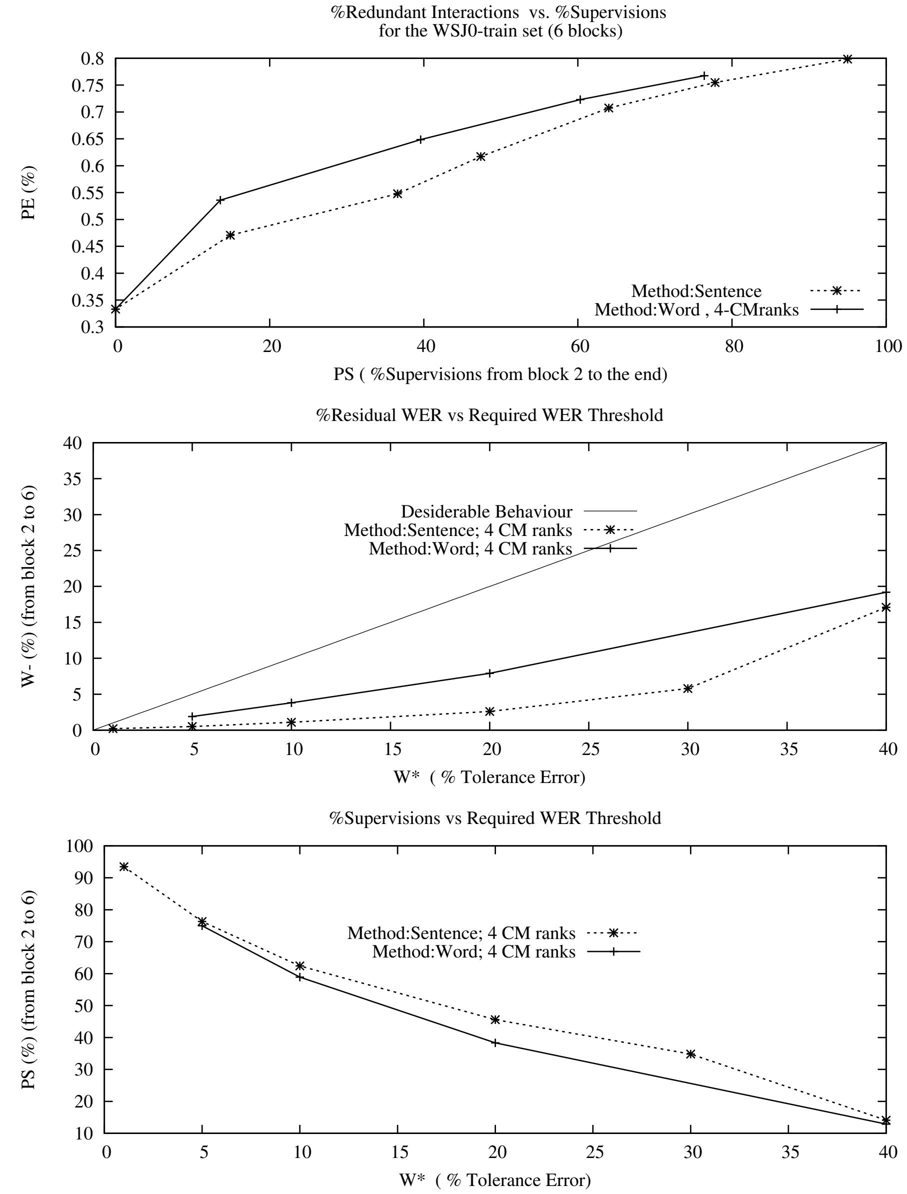


FIGURE 11. Comparision of considering a every word a whole utternace and considering one sentence as a utterance.

In figures 10 and 11, is shown that the sentence level performs better: the equal operations proportion is smaller and overall performance is smaller. It should be noted that the word level utterances has percentage of supervisions. This is because the method asks for high confident words as well as for the low confidence, since when the estimation is lower than the threshold the higher confidence words are not skipped, but just the next. This results in a quite random way of supervision, but the estimation is better since the supervised part have lots of high confidence words. Nevertheless, when putting together the supervisions and the resulting residual WER, the sentence-level utterances do outperform. Thus, from here all experiments will refer to the sentence-level version.

- Number of blocks

It is expected that smaller blocks, so the more number of them, the faster will the system improve. Two conducted experiments comparing a 6 block and a 12 block partition are depicted in figures 12 and 13

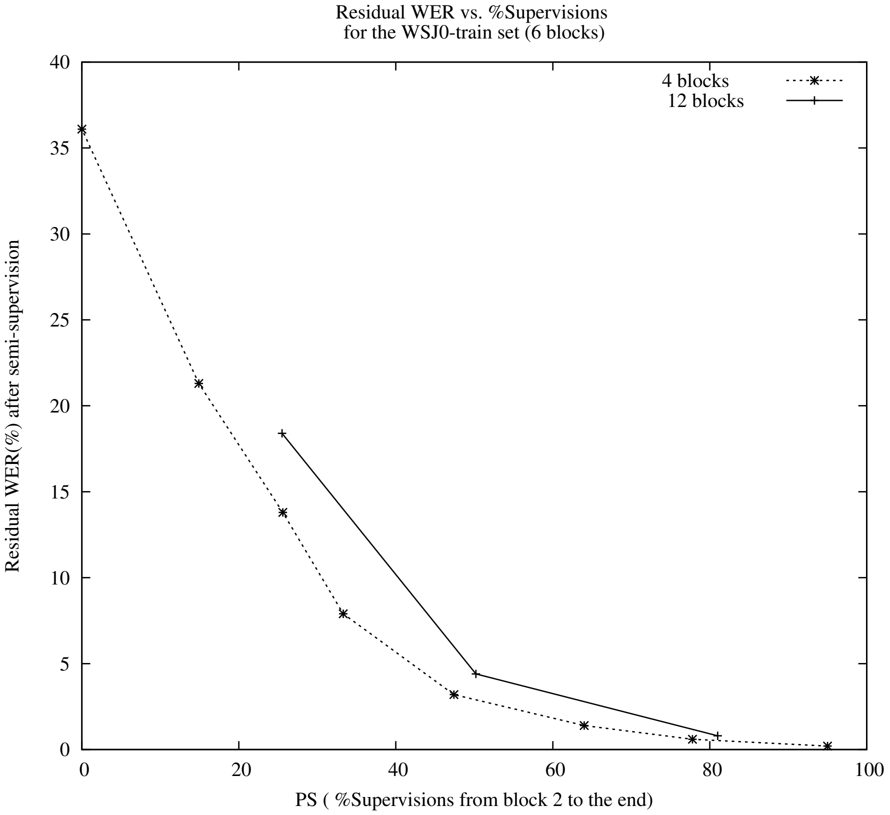


FIGURE 12. Overal performance of the method for several $W^*$. Segmentaton into 6 and 12 Blocks are compared against the baseline.

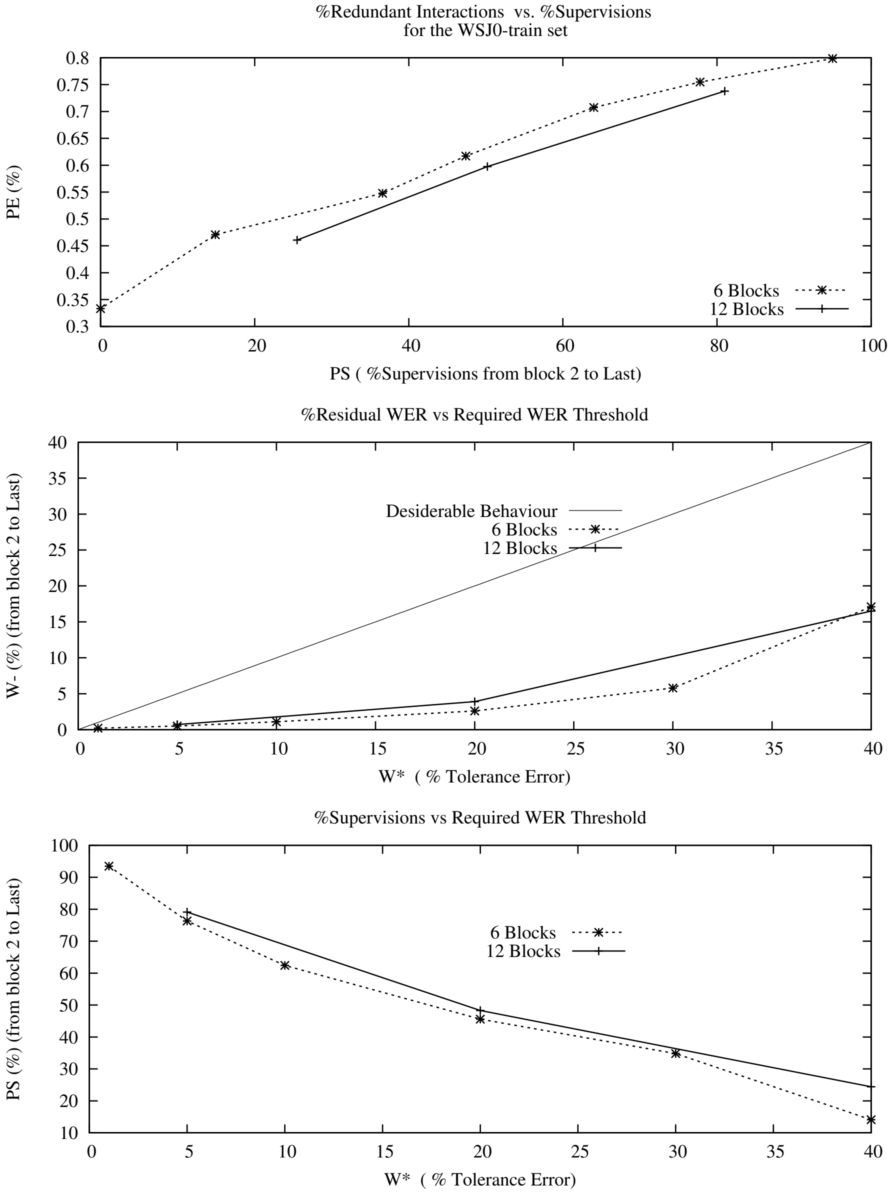


FIGURE 13. Comparision of segmenting the task into 6 and 12 blocks.

Results, however, show the same performance for low tolerances, while a slightly worse performance for the 12 blocks experiment for higher tolerances. Nonetheless, this is because the initial partition on the 6 block segmentation contains also the double quantity of speech than the initial block of the 12-block experiment. Starting with poorer recognition performances has a more detrimental impact when tolerating a high value of errors. From here, experiments conducted for the WSJ0 train set will be split into 12 blocks.

- Updating ASR and Threshold Classifier Parameters

The conducted experiments were the same as in the previous section for the WSJ0 train set (split into 12 blocks, recognized with the *iAtros* toolkit. The update of the parameters were performed as for the initial set, explained above, but using an external development set. The development set was the dev-nov'92-5K for the LM inc and LM 5K , and the dev-nov'92-20K for the LM 20K.

Differences in the results were quite small, in benefit for the update version in average for the WER obtained in the recognition of the blocks of the task. However, there was no significant difference when evaluating the external tests with the HMMs models for the fixed LM experiments. This found is reasonable since the task differs considerably in the language structure to the benchmark tests.

This is why all the presented results have been in the non-updated version; which requires much less computation time. Nevertheless, for other speech tasks, or when reusing the models resulting from one task to another, the update may be mandatory.

In the following, all the graphs corresponding to this assessment can found together. No for further remarks on the meaning of each graph as they are identical to those presented in the previous section for the WSJ0

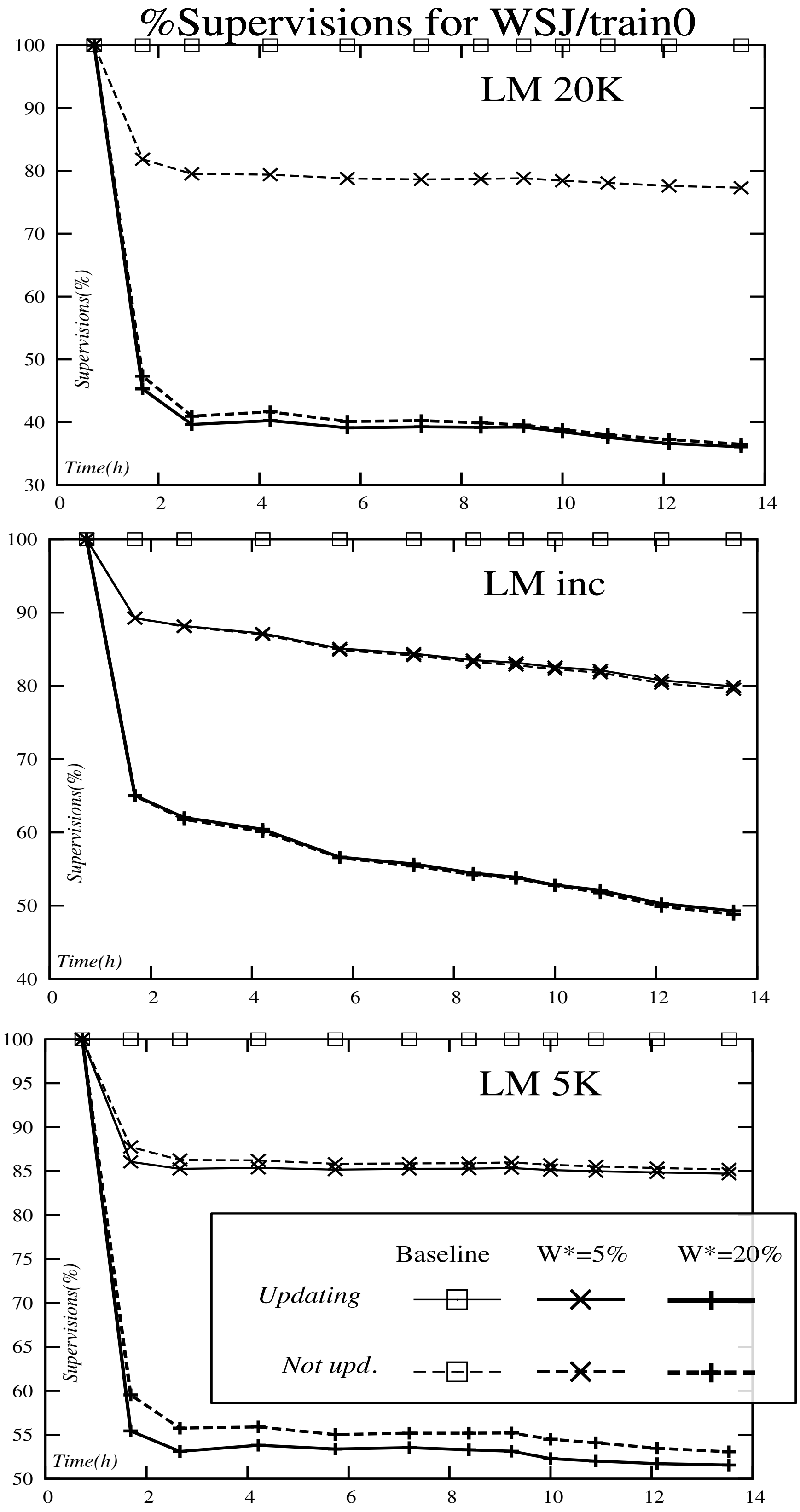


FIGURE 14. %Supervisions: Comparison between updating and not updating parameters

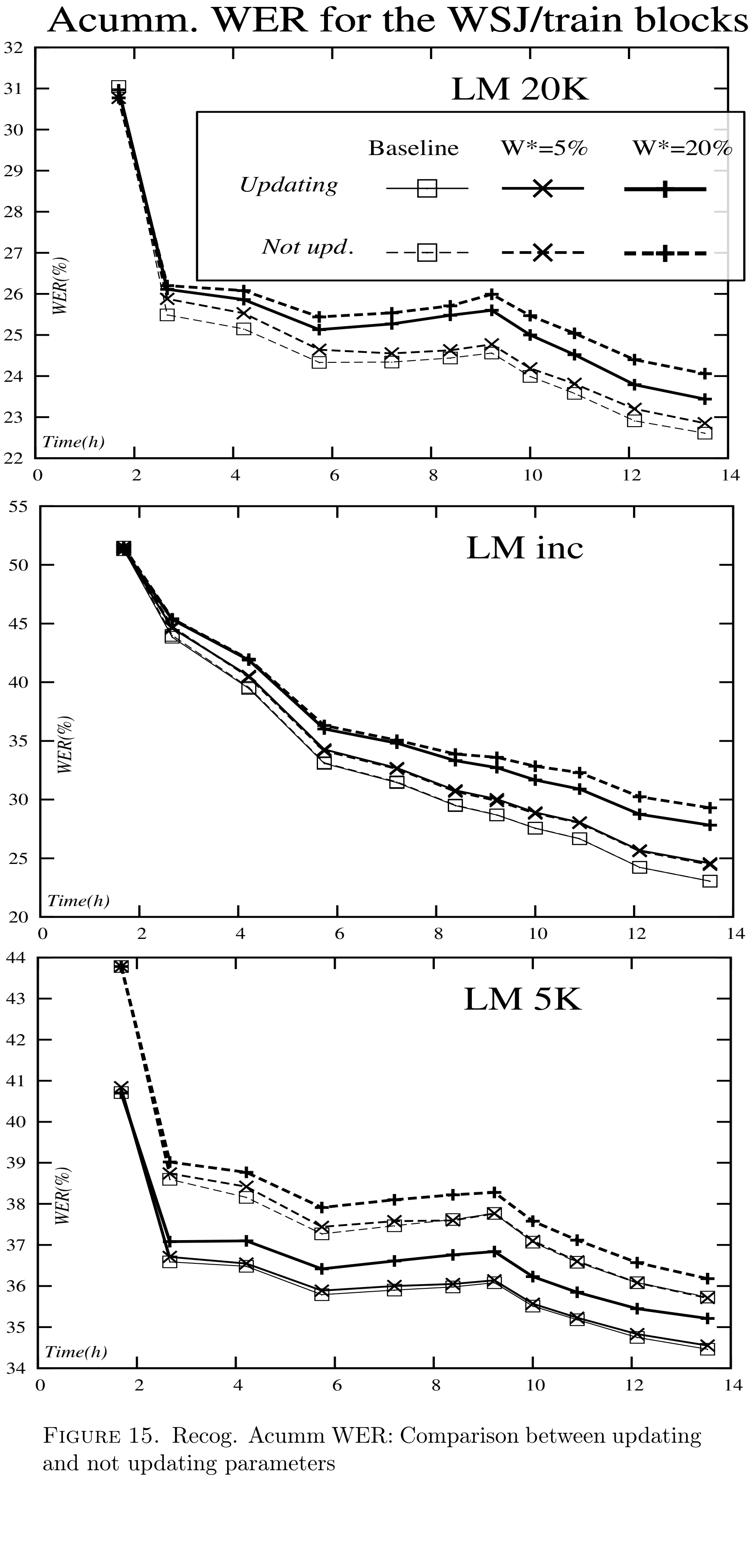


FIGURE 15. Recog. Acumm WER: Comparison between updating and not updating parameters

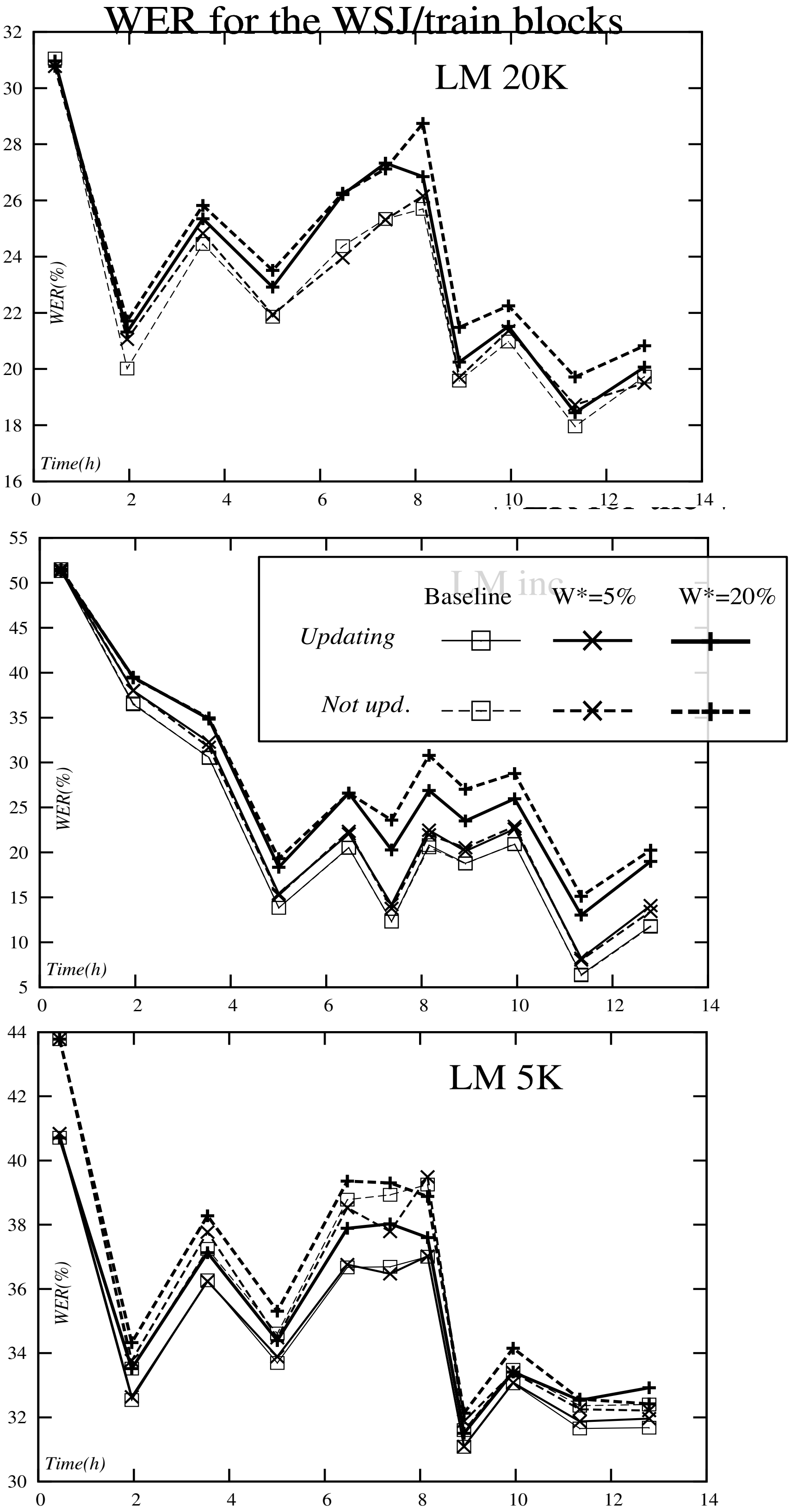


FIGURE 16. Recog. WER: Comparison between updating and not updating parameters

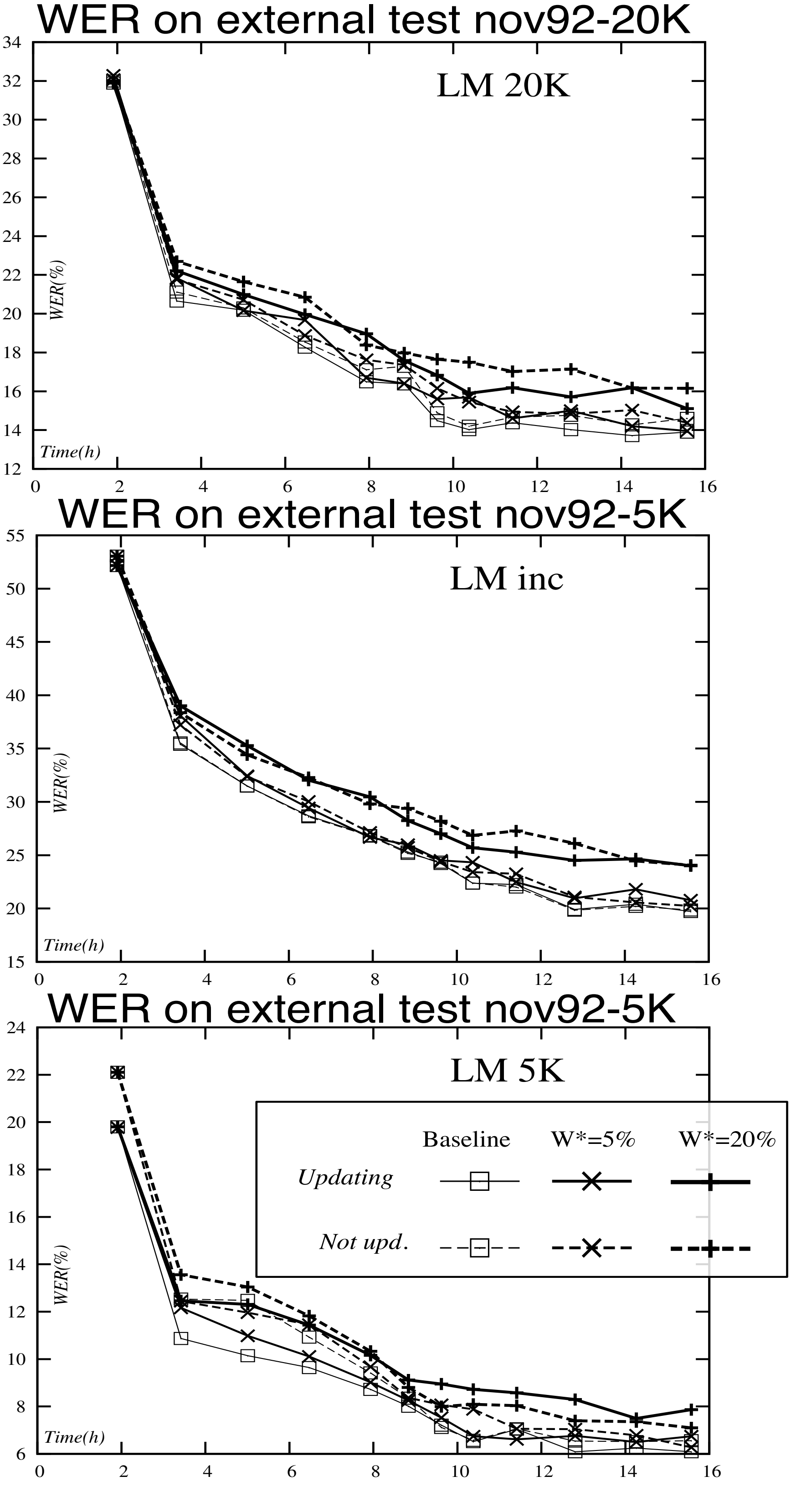


Figure 17. WER Test: Comparison between updating and not updating parameters

# CHAPTER 5

# A PROTOTYPE INTERFACE TO INTERACTIVE SPEECH TRANSCRIPTION

## 1. System Overview

The IST prototype has been thought to be an extension for tools for speech transcription.

A transcription is fulfilled iteratively by performing the following three steps, as explained in chapter 3. In summary, "Recognize", "Semi-supervise" and "Re-train". To do so, the user must first load a piece of the audio (a block), preferably already segmented into utterances with a duration such as those of a typical sentence. First, the user activates "Recognize". Once recognized a part of the speech, the minimum acceptable quality of the final transcript is established by a maximum tolerated Word Error Rate (WER), $W^*$. Then, the user starts "Semi-supervise" in order to be assisted by the system. Additionally, at anytime, the user can make the system to "Re-train" the ASR model in order to improve subsequent recognitions.

**1.1. Recognize.** ASR is performed based on a state-of-the-art statistical speech recognizer [5]. The IST prototype asks for a configuration file to read the location of audio samples and different parameters of the ASR system. Automatic transcriptions of a batch of utterances are obtained along with word-level confidence measures computed over word-graphs [20].

**1.2. Semi-supervise using Balacing Method.** The supervision of the automatically obtained transcriptions is performed as explained in chapter 3. Let us recall here the procedure along with some additional details concerning to the actual implementation of the prototype:

Sequentially from the first to las recognized utterance , in a word by word basis:

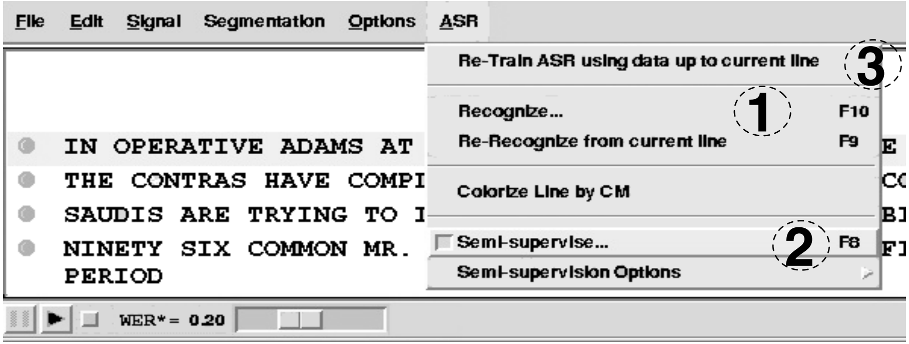
- The error, $\hat{W}^-$, of the unsupervised parts of the transcriptions up to the current utterance under revision is estimated If $\hat{W}^- > W^*$:
  - The system asks the next lowest-confidence word in the utterance:
    - The system catches the attention of the user over the new word. This is important, because the supervision is not performed sequentially.
    - The region to supervised is highlighted in the text-box, as well as in the audio graph. The audio portion is enlarged by a fixed margin of 15 ms to avoid chopping.
    - The corresponding audio is played once by default.
  - The user can replay the audio, validate the word or type the proper correction. However, the user is also let to correct the surrounding

words to the current under supervision. This is explained below in section sec:usability.
  - $\hat{W}^{-}$ is updated accordingly. If $\hat{W}^{-} < W^{*}$ it proceeds to next utterance.
- Add new samples to the training data using the supervised and high-confidence parts of the utterance.

**1.3. Re-Train models.** Acoustic and language models can be re-estimated any time after a new utterance is supervised. The system uses HTK [21] and SRILM [15] to train new acoustic and language models, respectively. Initial ASR models can be obtained from external sources similar to the task. Since this is not usually feasible a small part of the speech must be initially fully transcribed. The recognition of the non-supervised utterances can be performed every time the recognizer models are re-trained since it is expected to adapt progressively and perform better.

## 2. User interface of the prototype

The prototype is implemented as an extension to the *Transcriber tool* [2]. This extension is incorporated as an additional menu entry called "ASR" . This menu lets the user to follow the proposed interaction paradigm (see fig. 1): (1) First, recognize speech automatically , (2) Start/stop the assisting process using a method that allows up to some degree of errors in the final transcription. (3) Finally, Re-training the ASR models with the semi-supervised utterances.

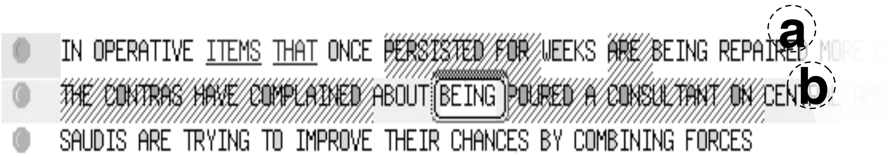


IN OPERATIVE ADAMS AT
THE CONTRAS HAVE COMPI
SAUDIS ARE TRYING TO I
NINETY SIX COMMON MR.
PERIOD

FIGURE 1. Screenshot of the ASR menu addition. 1) Recognition. 2) Interaction. 3) Re-training.

To start the recognition, the user must press the "F10" key or to select the "Recognize..." entry in the ASR menu. Then, a dialog for choosing a configuration file will appear. This configuration file defines the paths to the ASR model files; optimal configuration for the recognizer; word classifier and WER prediction method parameters; and the list of the audio files to recognize. These samples should be segmented at sentence level. After all, the recognized utterances will then appear in the application.

The "F8" key starts the correction guidance (interactive transcription): The text cursor is placed to the right of the word the system has decided that it deserves to be supervised ; The word background flashes in a striking color ; And the audio graph shows the corresponding region of the utterance corresponding to the selected word . After that, the word remains highlighted (see fig. 2 ).



IN OPERATIVE ITEMS THAT ONCE PERSISTED FOR WEEKS ARE BEING REPAIRED MORE

THE CONTRAS HAVE COMPLAINED ABOUT BEING POURED A CONSULTANT ON CEN

SAUDIS ARE TRYING TO IMPROVE THEIR CHANCES BY COMBINING FORCES

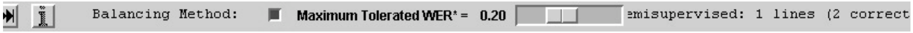
FIGURE 2. Outline of an screenshot of the textbox with the transcriptions. Shaded regions: Words recognized with low confidence measure values. Circled word: Word under supervision.

It should be noted that has been published that highlighting low confidence words helps user in the detection of the errors when the confidence estimation are correct ([17])

The selected audio is automatically played once. It should be noted that the enlargement of the piece of audio to be played influences the user decisions, especially for the number of insertions to be done.

Then, the user can enter the proper correction. At the same time, the user can re-play the piece of the audio as many times as required by pressing the tabulator key. Once finished, the user should press the enter key. The correction is underlined, and it assigned the highest possible confidence.

After each correction, the estimator of the residual WER ($W^{-}$) is updated. Then the system will ask the user to correct the next lowest confidence word in the sample, or it will jump to the next utterance if the estimator is below the tolerance error threshold ($W^{*}$). It should be noted that the required quality on the transcription was specified in the configuration file in term of $W^{*}$. This threshold can also be set by the user using a slider at any time (fig. 3).



Balancing Method: ■ Maximum Tolerated WER* = 0.20 emisupervised: 1 lines (2 correct

FIGURE 3. Screenshot of the textbox with the transcriptions. Shaded regions: Words recognized with low confidence measure values. Circled word: Word under supervision.

Finally, in order to improve the recognition of remaining utterances of the task, the speech recognizer model can be improved by retraining models (fig. ). New data pairs of audio an text will be added to the training set. These pairs are the built from the contiguous subsegments not in maroon background (i.e., only those supervised and high-confidence parts): Once a new model is available, the remaining utterances can be re-recognized. This should help in reducing the number of corrections the system will ask to the user.

## 3. Demonstration

The prototype was showcased and demonstrated on a special session during the IUI 2012 congress. A small speech corpus allowed the attendants to test our interactive speech transcription prototype. The corpus was intended to be automatically recognized in a laptop in very short time. It posed some typical recognition errors. Users were able to perform the corrections while being assisted by the system, to change the tolerance error, and to re-train the models to check how the recognition has improve and it whether it has learned new vocabulary.

A video with a short overview of the system is included in the CD of this master thesis.

## 4. Usability

No formal tests on the usability of the interface have been conducted. However, the overall impression of the authors and other non-professional transcripts who tested the demo at the congress was satisfactory. The main concerns came from a small percentage of the words asked for supervision which segmentation was wrong. This mainly happened for the extra inserted words (i.e. when a deletion operation should be performed). Even so, although non performed deletions increase the resulting WER, they are preferable for a real user from the language understanding point of view. Also, incorrectly placed extra words are preferable for indexing and term search applications that would use the resulting transcriptions. Instead, the omission of words is a critical issue.

In order to avoid the most of the chopped words an enlargement of the portion of the audio to be played of 15 ms was found to be the best option. Nevertheless, it would be desirable a cleverer way to modify the margins in order to increase the chances of understanding.

In the previous chapter it was stated that a user should attach strictly to some conventions. For instance, a user should deleted a word whenever less about the half of the word is uttered even if it can be figured out. However, while this servers properly for the evaluation purposes, allowing corrections more than those for the asked words to be supervised greatly improves the user experience and the final quality of the transcriptions. This behavior was implemented by simply performing an additional estimation of which parts of the introduced corrections corresponded to which words in the utterance under supervision, instead of assuming the correction corresponds strictly to the word under supervision. The estimation is performed by means of the of the Levenstein algorithm . The update of the $\hat{W}^-$ estimator is performed as if the extra corrected corresponding recognized words would have been asked for supervision, and that the corresponding corrections were introduced. It should be noted that this extra behavior is usually useful because the user can rapidly figure out some corrections just by listening the word under supervision and reading the recognized surrounding words. This way, in case that more than one recognized word was corrected, the system will skip the supervision of those recognized words if they would have been later selected for supervision. Thus, in those cases no increase of the user effort would have happened.

Finally, it should be noted that for better comprehension of the speech, the more audio context the better and the more sequentially performed corrections , the better. Thus, this is an important issue that is left as a future work.

# CHAPTER 6 CONCLUSIONS

## 1. On the method of balancing error and the user effort

In summary:
A simple yet effective method to find an optimal balance between recognition error and supervision effort has been applied to interactive speech transcription. Empirical results confirm previous works on handwriting recognition showing that this strategy is effective to reduce the supervision effort by allowing a maximum tolerance error in the speech transcriptions. Moreover, results show that a tolerance error in the transcriptions does not affect critically on the incremental learning of acoustical models. Thus, this method can be used also for producing ASR models resulting in similar performance to those generated using fully manually transcribed corpora.

General Remarks Concerning the evaluations :

- The estimation of the residual WER ($\hat{W}^{-}$) is always pessimistic, which is preferable.
- The pessimistic behavior is due to nature of the method, that is based solely in the supervised low-confidence parts.
- The pessimistic behavior yielded to many spurious supervisions (equal operations). Nevertheless, equal operations do not require as much effort as other necessary operations; and , in turn, they help to improve the estimation which would be too pessimistic if not.
- The poor performance of the confidence measures, especially at the end of the tasks, worsens the $\hat{W}^{-}$; Fortunately, the effect is insignificant since $\hat{W}^{-}$ is robustly estimated throughout the previous blocks. Also, it makes the system more likely to asks for more words that it are correct.
- The initialization poses an effect over the performance, but a reasonable initialization is easy to find.
- The update of the parameters of the recognizer and the confidence measure classifier yield no significant improvement, but it seems that in a general scenario it would.
- The procedure performs better when it works at sentence level.
- In practice, badly segmented words (which is usually the cause of the wrong recognition), poses a difficulty for the user. Although it might be alleviate enlarging the margins by just 15ms.

Remarks on the large task WSJ1+0:

- User effort is greatly reduced as it learns (fig. 4).
- Recognition performance is just slightly worse than the best possible (fig. 5 top).
- WER difference of each recognition between an experiments and the baseline is maintained in average (fig. 5 bottom). This means that the slight worsening in the accumulated WER throughout the task, is due only to the accumulation of errors, not a degrade in the ASR quality.
- Word classifier performance behaves as expected, but its performance degrades over time (fig. 4 top). As a consequence, the number of redundant supervisions increases (fig. 4 bottom).
- $\hat{W}^-$ insignificantly varies around requested $W^*$.
- $\hat{W}^-$ is always pessimistic (which is desirable): resulting transcriptions are always better than expected.
- The system effectively adapts, in terms of the number of supervisions, to the true, unknown, quality of recognition: On block 41 (table 4), 50% to 68% of the recognized words were supervised. Next block (table 5), which was poorly recognized, 98% of the words. While, only 60% to 81% using an external LM improved the recognition (Table 5).

## 2. Future Work

Although a great reduction in the effort can be achieved with this simple method, there are three main aspects that should be addressed:

• The issue of the isolated words: The lack of audio context and the badly automatically segmented words makes harder for the user figuring up the proper correction. Furthermore, the system keeps jumping from one place in the sentence to another until it decides to jump to the next utterance. To alleviate this problem the driver of the method should include rules and cost functions in order to group into one larger segment words that are likely to be wrongly regonized. Also, it will be better if the segments were asked from left to the right.

• The confidence measures: A bad performance of the confidence measures mislead the overall process. Thus better performance should be achieved. Our next research will precisely be centered on this topic by finding regions of low confidence instead of isolated words. This will improve the capturing of insertions on regions with no recognized words. Also it will help in using a segment driven approach, instead of the word-driven, as just proposed.

• The estimation of the error: The estimation might be further refined using more features than the rank level of confidence measure. It could be modeled depending on the words itself, the relative position in the sentence, the continues value of the confidence measure, etc.

# BIBLIOGRAPHY

[1] Behrouz Abdolali and Hossein Sameti. A Novel Method For Speech Segmentation Based On Speakers' Characteristics. *arXiv.org*, cs.AI, May 2012.

[2] C. Barras, E. Geoffrois, Z. Wu, and M. Liberman. Transcriber: a Free Tool for Segmenting, Labeling and Transcribing Speech. In *Proceedings of LREC*, pages 1373–1376, 1998.

[3] A. gimenez. AK: Adriaś kit . *prhlt.iti.upv.es*, pages –, 2012.

[4] D. Hakkani-Tür, G. Riccardi, and G. Tur. An active approach to spoken language processing. *ACM Transactions on Speech and Language Processing (TSLP)*, 3(3):1–31, 2006.

[5] M. Luján Mares, V. Tamarit, V. Alabau, C.D. MartÄśnez-Hinarejos, M.P. i Gadea, A. Sanchis, and A.H. Toselli. iATROS: A speech and handwritting recognition system. *V Jornadas en TecnologÄśas del Habla (VJTH'2008)*, pages 75–78, 2008.

[6] Saturnino Luz, Masood Masoodian, and Bill Rogers. Interactive Visualisation Techniques for Dynamic Speech Transcription, Correction and Training. In Stuart Marshall, editor, *Proceedings of CHINZ 2008, The 9th ACM SIGCHI-NZ Annual Conference on Computer-Human Interaction*, pages 9–16, Wellington, New Zealand, 2008. ACM Press.

[7] D.S. Pallett, J.G. Fiscus, W.M. Fisher, and J.S. Garofolo. Benchmark tests for the DARPA spoken language program. In *Proceedings of the workshop on Human Language Technology*, pages 7–18. Association for Computational Linguistics, 1993.

[8] B Ramabhadran, O Siohan, and A Sethy. The IBM 2007 speech transcription system for European parliamentary speeches. In *IEEE Workshop on ASRU*, pages 472–477, 2007.

[9] Luis Rodríguez-Ruiz, Francisco Casacuberta, and Enrique Vidal. Computer Assisted Transcription of Speech. In *Proceedings of the 3rd Iberian Conference on Pattern Recognition and Image Analysis, Volume 4477 of LNCS*, pages 241–248, 2007.

[10] R. Sánchez Sáez, J.A. Sánchez, and J.M. Benedí. Confidence measures for error discrimination in an interactive predictive parsing framework. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 1220–1228. Association for Computational Linguistics, 2010.

[11] A. Sanchis, A Juan, and E Vidal. A Word-Based Naïve Bayes Classifier for Confidence Estimation in Speech Recognition. *Audio, Speech, and Language Processing, IEEE Transactions on*, 20(2):565–574, 2012.

[12] Alberto Sanchis. *Estimación y aplicación de medidas de confianza en reconocimiento automático del habla.* PhD thesis, Departamento de Sistemas Informáticos y Computación, 2004.

[13] N. Serrano, A. Sanchis, and A Juan. Balancing error and supervision effort in interactive-predictive handwriting recognition. In *Proceedings of the 15th international conference on Intelligent user interfaces*, pages 373–376. ACM, 2010.

[14] G Stemmer, S Steidl, E Nöth, H Niemann, and A. Batliner. Comparison and combination of confidence measures. *Text, Speech and Dialogue*, pages 561–582, 2006.

[15] A. Stolcke. SRILM - An Extensible Language Modeling Toolkit. In *ICSLP*, 2002.

[16] Martin Sundermeyer, Markus Nußbaum-Thom, Simon Wiesler, Christian Plahl, Amr El-Desoky Mousa, Stefan Hahn, David Nolden, Ralf Schlüter, and Hermann Ney. The RWTH 2010 Quaero ASR evaluation system for English, French, and German. In *ICASSP*, pages 2212–2215. IEEE, 2011.

[17] Keith Vertanen and Per Ola Kristensson. On the benefits of confidence visualization in speech recognition. In *CHI '08: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems.* ACM Request Permissions, April 2008.

[18] Y.Y. Wang, A. Acero, and C. Chelba. Is word error rate a good indicator for spoken language understanding accuracy. *Automatic Speech Recognition and Understanding, 2003. ASRU'03. 2003 IEEE Workshop on*, pages 577–582, 2003.

[19] F. Wessel and H Ney. Unsupervised training of acoustic models for large vocabulary continuous speech recognition. *IEEE Transactions on Speech and Audio Processing*, 13(1):23–31, 2005.

[20] F. Wessel, R. Schlüter, K. Macherey, and H Ney. Confidence measures for large vocabulary continuous speech recognition. *IEEE Transactions on Speech and Audio Processing*, 9(3):288–298, 2001.

[21] S.J. Young, Woodland, P.C., and W J Byrne. *HTK: Hidden Markov Model Toolkit V1.5.* Cambridge Univ. Eng. Dept. and Entropic Research Labs Inc., Cambridge, UK, 1993.