On the Design of an Automatic Speech Recognition System for Romanian Language

Alexandru Caranica, Horia Cucu, Andi Buzo, Corneliu Burileanu

Speech and Dialogue Research Laboratory, University Politehnica of Bucharest
(E-mail: alexandru.caranica@yahoo.com)

Abstract: For decades, engineers and scientists have studied the phenomenon and production of speech, with an eye on creating more effective and efficient systems for human-computer interaction. This paper presents a large number of experiments made to create an automatic speech recognition system (ASR) for spoken Romanian connected digits. State-of-the-art hidden Markov acoustic models (HMMs) and a finite state grammar language model are used, in order to build and optimize a fully-functional digit recognizer system in Romanian language. The applications of speech recognition in daily life are multiple, and truly there are no limits to the use cases of this technology: from niche applications like medical interfaces and industrial command and control systems to consumer applications, where modern phone operating systems offer speech interfaces to interact with the system. The rich mathematical framework of HMMs makes statistical approaches very feasible for this task, and one of the goals of this paper is to confirm the validity and reproducibility of this method. Another objective is the integration of the components and toolkits necessary to build a recognition system, briefly describe the processes involved in speech representation, the mathematics behind it and the analysis for improving and optimizing the primary evaluation metrics. The results show the advantage of training with a larger speaker database, in order to obtain an independent speech recognizer, with more than 60% WER improvements compared to a dependent model, for a 90 speaker database used for evaluation. The implementation of the system and the experiments, along with the evaluation results for decoding and optimization are provided.

Keywords: signal processing, automatic speech recognition, hidden markov model, phones, senones, MFCC, feature vectors, language models.

1. INTRODUCTION

Speech processing is one of most important branches of digital signal processing. Speech signals can be used for continuous automatic speech recognition (ASR), speaker identification or voice command recognition systems. The goal of automatic speech recognition is to map an audio signal, containing speech, into a text transcription containing a sequence of words. The idea is to match the transcription as close as possible to the audio message, without necessarily understanding the meaning or scope of what was spoken.

Over the past decades, machine learning has become one of the main stays in computer technology and with that, a rather central part of our daily life, when we are interacting with "smart" electronic devices. Hands free applications increased in usage, even in consumer based hardware. Applications such as Apple`s Siri, or Google Now offer speech commands, as a direct interface to the phones operating system. According to Jurafsky (Jurafsky et al., 2008), while many tasks are better solved with visual or pointing interfaces ("keyboard" or "touch"), speech has the potential to be a better human-computer interface than the keyboard for tasks where full natural language communication is useful, in which touch panels are also not appropriate. This can include hands-busy or eyes-busy applications, such as where the user has objects to manipulate or equipment to control, and his attention cannot be detained from the job at hand. This can further be explained by the fact that speech is the most "natural communication method used by humans to exchange information", and the human user is not required to have any additional skills to be able to use a speech enabled device.

Other major applications are in telephony, where speech recognition is already used in many call centres all over the world, for entering digits and identify a person's ID card, recognizing "yes" / "no" to answer certain questions, finding out airplane / train / bus information and call-routing ("To human resources department, please"). In some applications, combining speech and pointing devices in a multimodal interface, can be more efficient than a graphical user interface without speech (Cohen et al., 1998). Medical applications are a good example of a multimodal interface. Chernakova (Chernakova et al., 2006), exemplified the design and improvement of a medical computer visualization system, for diagnostic and surgery operations, with good uses in training of doctors and students in the control of medical equipment. In this multimodal system, two input types are combined: speech, head movements and stereo viewing of images, controlled by voice or head gestures.

Speech-to-speech translation represents, at this moment, another hot topic in many academic and industrial research centres. At WPC 2014 (Microsoft, 2014), Microsoft used a

Skype add-on called "Translator" to demonstrate a live German to English "on-the-fly" translation. This "universal translator" is derived from decades of research in speech-recognition, automatic-translation and machine-learning technologies, and it is developed at Microsoft Research. Finally, automatic speech recognition is applied to dictation, where transcription of an extended monologue by a single specific speaker is transcribed into text for further reference (archival of audio meetings, law trials transcription, etc.).

Some of the major applications of speech recognition have been illustrated in this introductory section. This list is by no means complete, examples have been chosen in an effort to try and cover the most popular applications and use cases. Also, some attention has been paid to covering examples for the different areas of speech recognition research directions.

This paper presents the design of an automatic speech recognition system (ASR), making use of state-of-the-art Hidden Markov/Gaussian Mixture acoustic models (HMM/GMM) and a finite state grammar language model, in order to build and optimize a fully-functional digit recognizer system in Romanian language. Section 2 offers an overview of the current progress in the field of speech processing, including relevant studies for this task. Continuing with section 3, the paper takes a look at the processes involved in speech recognition, briefly describes digital speech representation, the mathematics behind it and the rich mathematical framework of HMMs for acoustic modelling. Section 4 and 5 present in detail the methodology and toolkits necessary to build a speech recognition system, along with information about the training and evaluation data used in this paper. The analysis for improving and optimizing the primary evaluation metrics are covered in section 6. The conclusions and perspectives of future work are formulated in Section 7.

2. OVERVIEW OF ASR

Research in the field of automatic speech and speaker recognition has now spanned more than five decades (Furui, 2009). After all these years of research and development, the problem of automatic speech recognition is still an open issue. To design a machine that mimics human behaviour, particularly the capability of speaking naturally and responding properly to spoken language, in the context of a high degree of correlation in the spoken content, has intrigued engineers and scientists. The end goal of a perfect translation into a word sequence, very accurate and efficient, unaffected by speaker particularities, noisy environment or transmission channel, is very difficult to achieve, and many challenges are to be faced. Figure 1 shows a timeline of progress in speech recognition and multimodal understanding technology, over the past decades.

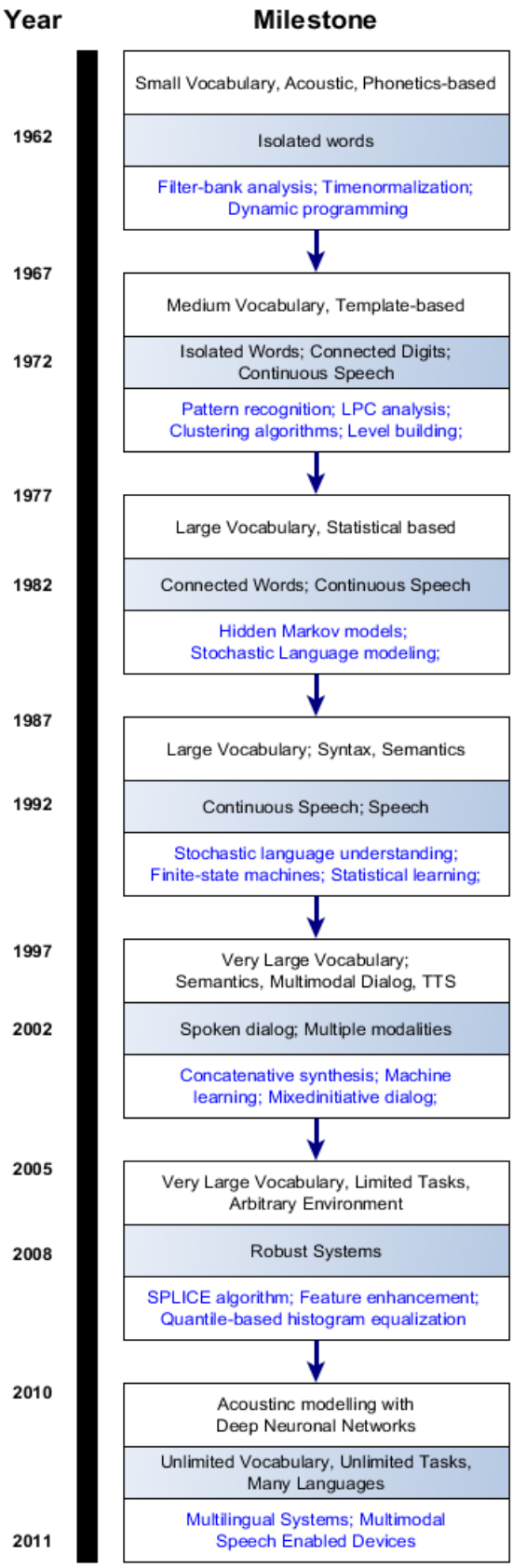

Fig. 1. Milestones in Speech Recognition (adapted Juang, 2005).

The earliest attempts to build ASR systems were made in the 1950s and 1960s. Various researchers tried to exploit fundamental ideas of acoustic phonetics (Bell Laboratories, NEC Laboratories), by using the formant frequencies measured / estimated from the speech signal, to successfully recognise isolated digits. Throughout the decades, digital signal processing advances coupled with increases in computational power, led to the introduction in the 1960's and 1970's of the advanced speech representations, based on LPC analysis and cepstral analysis methods. In 1980's, through the introduction of rigorous statistical methods based on Hidden Markov Models, a shift in methodology took place, from the more intuitive template-based approach (a straightforward pattern recognition paradigm), towards a more rigorous statistical modelling framework. This was possible with significant research contributions from academia, private industry and the governments.

Nowadays, most practical speech recognition systems, commercial or for academic research, are based on the statistical framework developed in the 1980s, by Baker (1975), a team at IBM (Jelinek 1976; Bahl et al., 1983), and a team at AT&T (Levinson et al., 1983; Rabiner, 1989). HMMs are used because speech can be viewed as a stationary signal or a short-time stationary signal, under certain conditions. For short time-periods (up to 20ms), speech can be approximated as a stationary process, and analysed. This way, speech can be thought of as a Markov model for many stochastic purposes. Significant additional improvements were made during the 90s, in the field of pattern recognition, with focus on the optimization problem, involving minimization of the empirical recognition error.

The dominance of GMM-HMM in acoustic modelling, led to an *ecosystem* of speaker adaptation and front-end processing techniques, tailored to maximize the performance under this model. This was hard to challenge over time, until very recently, with a new competing acoustic modelling approach: Deep neuronal network acoustic model for large vocabulary continuous speech recognition systems. (Dahl et al., 2012) reported a 33% relative improvement in WER over a discriminatively trained GMM-HMM on a 300 hour English conversational telephone transcription task. "Deep" comes from using more than one hidden layer, typically three to five, to model context-dependent output distributions directly. In contrast to HMMs, Neural Networks make no assumptions about feature statistical properties. Neural Networks allow discriminative training, in a natural and efficient manner, when used to estimate the probabilities of a speech feature segment. Neural networks have been used in many aspects of speech recognition, such as phoneme classification (Waibel et al., 1989), recognition of isolated words (Wu et al., 1993) and speaker adaptation.

In Romania, the interest for automatic speech recognition and processing manifested since three decades ago, but studies became systematic after 1980. Research teams were organized in major academic centers, such as Bucharest (prof. Corneliu Burileanu, SpeeD group), Cluj (prof. Gavril Toderean, prof. Mircea Giurgiu), Iași (prof. Horia-Nicolai Teodorescu) and Timișoara (prof. Marian Boldea). Areas of interest include automatic speech recognition, speaker recognition and identification, voice synthesis, speech coding, natural language processing (Burileanu et al., 2004), spoken term detection and lately document indexing/retrieval. Besides HMM, other known strategies used by Romanian authors for speech recognition are the neural network connectionist ones, with fuzzy sets (Dumitru et al., 2008). Lastly, there are the hybrid methods. An example is the Fuzzy-HMM approach, based on fuzzy integrals. Fuzzy measures have an essential property monotonicity with respect to set inclusion, far weaker than the usual additive property for probability measures (Militaru, 2014).

This section offered a brief overview in the field of speech recognition. One mention should be that in recent years, the scientific community also focused on increasing the intelligibility of the ASR output, through methods like punctuation and capitalization restoration (Gravano et al., 2009), robust diacritics restoration, in the context of high recognition accuracy and performance.

3. SPEECH RECOGNITION PROCESS

Speech recognition is a difficult task, and is one of most important branches in digital signal processing. Figure 2 presents the general architecture of an automatic speech recognition system, which emphasizes the two main steps of the architecture, namely the training and decoding tasks.

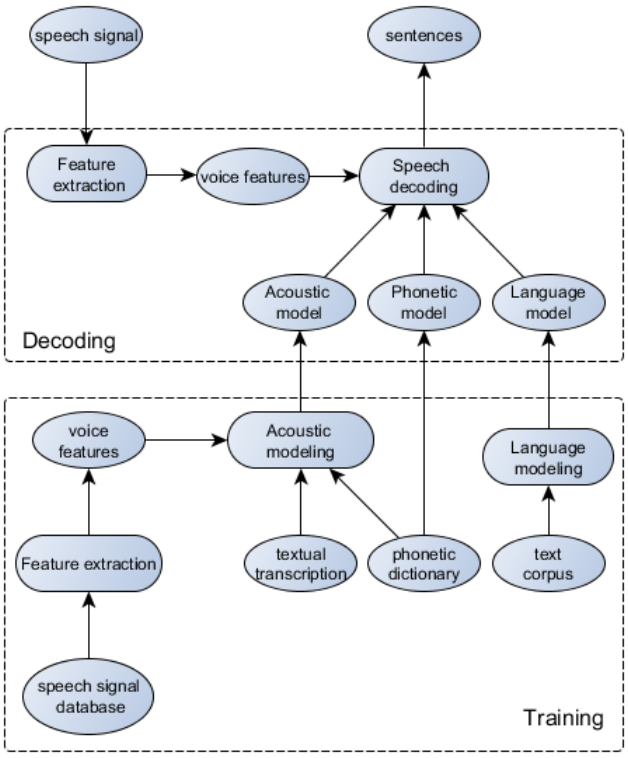

Fig. 2. General architecture of an automatic speech recognition system (ASR).

Figure 2 also shows that the ASR system does not model speech directly (at the waveform level). Here, the feature extraction block is employed to extract specific features,

which are further used in the acoustic model creation. The same block is also used in the recognition or decoding process for speech representation.

3.1 Acoustic Features

The speech signal is processed in order to extract information for further analysis (at a significantly lower sample rate). This is called *feature extraction*, and is the process of extracting unique information from speech files that can later be used to compute some feature vectors which will be eventually modelled by the acoustic model. Speech signal is a quasistationary, slowly timed varying signal. Over a sufficiently short period of time (between 5 and 20ms), its characteristics are fairly stationary. However, over long periods of time, the signal characteristic change to reflect the different speech sounds being spoken. There are many techniques used to parametrically represent a voice signal for speech recognition tasks, for digital use. These techniques include Linear Prediction Coding (LPC), and the Mel Frequency Cepstrum Coefficients (MFCC). Lately, a new technique, noise-robust PNCC features, that implies usage of power law (instead of log) and gammatone filters (instead of triangular) arose. The MFCC features (Figure 3) were used in this paper, as they are implemented in the Sphinx Toolkit used for development of the system. MFCC's are based on the known variation of the human ear's critical bandwidths with frequency. Filters spaced linearly at low frequencies and logarithmically at high frequencies have been used to capture the phonetically important characteristics of speech (Price et al., 2006).

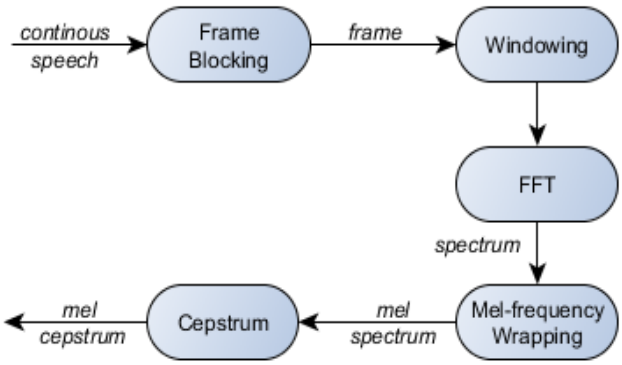

Fig. 3. Block diagram of the MFCC feature extraction module.

In the frame blocking section, the speech signal is broken into frames. The windowing block minimizes the discontinuities of the signal by tapering the beginning and end of each frame to zero. The FFT block converts each frame from the time domain to the frequency domain representation. In the Mel-frequency wrapping block (see figure 4), the signal is plotted against the Mel-spectrum. This mimics the human hearing, as studies have shown that human perception of the frequency contents of sounds for speech signals does not follow a linear scale. The mel-frequency scale is a linear frequency spacing below 1000 Hz and a logarithmic spacing above 1000 Hz.

Figure 4 shows the impact of the Mel-frequency wrapping on the speech utterance "zero", pronounced in Romanian. In the first plot, most of the information is contained in the lower frequencies. This information is then amplified in the second plot (formant frequencies) through Mel filter banks.

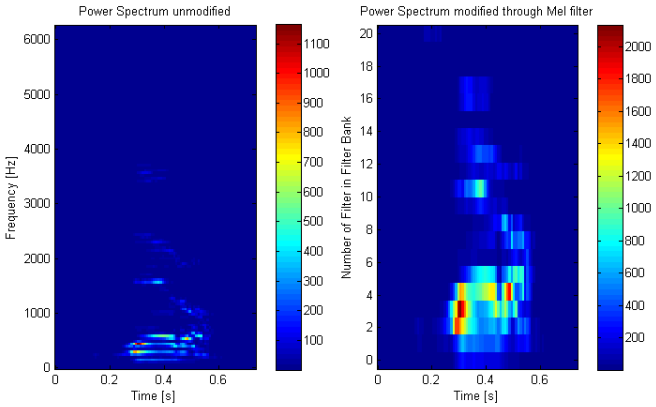

Fig. 4. Spectrum plot of a speech file, before and after the mel-frequency wrapping block. Note that the spectrum is shown in a linear and not a logarithmic scale.

Finally, the Discrete Cosine Transform (DCT) is used to obtain MFCC coefficients (see equation (1)), where n = *0,1,...,K*-1 and $\tilde{S}_0, k = 0,2,...,K-1$ are mel power spectrum coefficients.

$$\tilde{c}_n = \sum_{k=1}^{K} (\log \tilde{S}_k) \cos\left[n\left(k - \frac{1}{2} \right) \frac{\pi}{K} \right] \tag{1}$$

After the above steps, for each speech frame, a set of mel-frequency cepstral coefficients are computed, which are called an *acoustic vector*.

3.2 Language Modelling

These features extracted out of the speech signal are used for further modelling, in the training phase, or for speech recognition, in the decoding phase. The state-of-the-art technique, for modelling basic speech units (sub-word units, phones) in speech recognition, is the statistical Hidden Markov Model. The speech-to-text task can be formulated in a probabilistic manner as follows: *What is the most likely sequence of words W* in the language L, given the speech utterance X:*

$$W^* = \arg\max_w P(W \mid X) \tag{2}$$

$$W^* = \arg\max_w \frac{p(X \mid W) P(W)}{p(X)} \tag{3}$$

$$W^* = \arg\max_w \log p(X \mid W) + \log P(W) \tag{4}$$

Equation (2) specifies the most probable word sequence as the one with the highest posterior probability given the acoustics and the model. Equation (3) is a result of equation (2), through the application of Bayes's theorem, to compute this posterior probability. Since *p(X)*, the probability of the speech utterance, is independent of the word sequence, it can

be ignored. Consequently, equation (3) becomes equation (4) after applying logarithm. P denotes a probability and p denotes a probability density function.

From equation (4), the problem of searching for the most likely sequence of words in an utterance may be split into two separate components: language modelling, which is concerned with estimating the prior probability of a word sequence $P(W)$, and acoustic modelling, in which the likelihood of the acoustic data given the words, $p(X|W)$, is estimated. The parameters of both of these models are learned from large data corpuses. Obtaining the optimal word sequence W^* is the search / decoding problem.

3.3 Acoustic Modelling

For large vocabulary systems, the acoustic model does not model all the words in the vocabulary. Sub-words unit such as phonemes are used in this case, to map the words in the vocabulary to their phonetic representation (phonetic model). Sometimes phones are considered in context. Thus, a phoneme is strongly affected by its immediately neighbouring phonemes, making this units context dependent. Recognition accuracy can be significantly improved if there is enough training data to estimate these context-dependent parameters. Both phonetic and sub-phonetic units have the same benefits, as they share parameters at unit level. Parameter sharing is extended to subphonetic models, to treat the state in phonetic hidden Markov models as the basic subphonetic unit, a senone (Huang et al., 1993). Huang and Hwang (Huang et al., 1993) further generalized clustering: senones are constructed by clustering the state dependent output distributions across different phonetic models. Each cluster thus represents a set of similar Markov states and is called a senone (Huang et al., 1993). A sub-word model is thus composed of a sequence of senones after the clustering is finished.

Therefore, a senone's dependence on context could be more complex than just left and right context, it can rather be defined by a complex function with a decision tree. Phones vary enormously, they are influenced by phones on either side, because of the articulators (tongue, lips) movements during speech production process. Therefore, an articulator may start moving during one phone to get into place in time for the next phone, and so on. Context dependent phones capture an important source of variation, and are a key part of modern ASR systems. But context-dependency also introduces the same problem found in language modelling: training data sparsity. The more complex the model to be trained, the less likely to have seen enough observations of each phone type to train on.

The language model $P(W)$, models a word sequence by providing a predictive probability distribution for the next word based on a history of previously observed words. Since this probability distribution does not depend on the acoustics, language models may be estimated from large textual corpora. The conventional n-gram language model, which approximates the history as the immediately preceding $n - 1$ words, has represented the state of the art for large-vocabulary speech recognition for 25 years. Due to computational reasons, the history of preceding words cannot extend to include an indefinite number of words and has to be limited to m (3 to 5) words. Only a limited number of previous words affect the probability of the next word. Most commonly, trigram language models are used. They consider a two-word history to predict the third word. This requires the collection of statistics over sequences of three words, so-called 3-grams (trigrams).

Previous paragraphs detailed the features that are extracted "out of the speech signal, for further modelling (training phase) or for speech recognition (decoding phase)." The approach for modelling basic speech units (phones) makes use of the HMM/GMM framework, introduced in the middle part of this section. A HMM is a probabilistic finite state automaton, consisting of a set of states connected by transitions, in which the state sequence is hidden. Instead of observing the state sequence, a sequence of acoustic feature vectors is observed, generated from a probability density function attached to each state. This is why the Markov process is considered to be "hidden" – the state sequence is not directly available to the observer. This probability density functions are usually a Gaussian mixture models (GMM) density distributions that characterizes the statistical behaviour of the feature vectors within the states of the model (Rabiner et al., 2007).

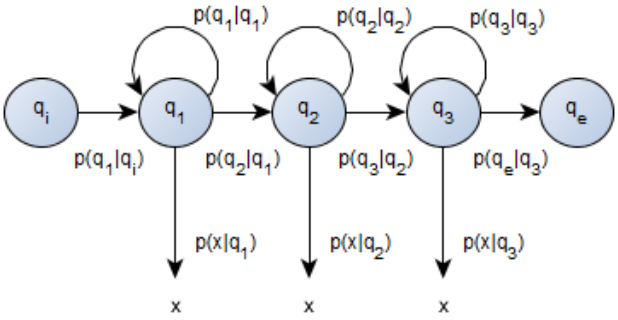

Fig. 5. Representation of a HMM as a parameterized stochastic finite state automaton.

A more detailed representation of an HMM is presented in Figure 5. As the figure shows, an HMM is characterized by these parameters:

- States: a set of states $Q = q_1 q_2 \ldots q_N$;
- Transition probabilities: a set of probabilities $A = a_{11} a_{12} \ldots a_{NN}$. Each $a_{ij} = p(q_j|q_i)$ represents the probability of transitioning from state i to state j;
- Observation likelihoods: a set of observation likelihoods $B = b_i(x_t) = p(x_t|q_i)$, each expressing the probability of an observation x_t being generated from the state i.

In speech recognition, the models are created to disallow arbitrary transitions, just as Figure 5 shows, to model the sequential nature of speech, placing strong constrains on transitions backward or skipping transitions. The use of self-loops allows a sub-phonetic unit to repeat so as to cover a variable amount of the acoustic input.

The observation likelihoods are probability density functions, for a state q_i is a d-dimensional Gaussian, parameterized by a mean vector μ_i and a covariance matrix Σ_i:

$$b_i = \frac{1}{(2\pi)^{d/2} |\sum_i|^{1/2}} \exp(-\frac{1}{2}(x-\mu_i)^T \Sigma_i^{-1}(x-\mu_i)) \quad (5)$$

where d equals 39 for a typical acoustic vector.

The HMM decoding issue, like finding the most likely sequence of states that have generated a sequence of observations, is solved by a variant of the Viterbi algorithm, and the various parameters of a HMM/GMM system are estimated using Forward-Backward algorithm (Baum-Welch).

All of the above aspects of the HMM paradigm play a crucial role in ASR systems.

4. METHODOLOGY

A connected-digits speech recognition system is a limited-vocabulary recognizer. This means the system will only recognize and transcribe the decimal system, in Romanian: *zero*, *unu*, *doi*, …, *nouă*. Speaker characteristics were taken into account in this paper. Theoretically, a speaker-dependent system should be better at decoding speech uttered by the specific user for which it was trained. However, this demands a new system to be constructed and trained individually for each speaker, a time consuming and non-scalable task. To address this issue, a second speaker-independent system was trained with multiple audio files from SpeeD (Speech & Dialogue Research Laboratory) "roDigits" speech database. Results were compared in terms of word-error-rate (WER), sentence-error-rate (SER) and are presented in Section 6. The effects of increasing/decreasing the number of Gaussians per senone and the number of tied-states (senones) for each system were also shown in Section 6.

The CMU Sphinx Toolkit (Lamere et al., 2003) is used to implement the ASR architecture described in Figure 2. CMU Sphinx, also called Sphinx in short, is the general term to describe a group of speech recognition systems developed at Carnegie Mellon University. These include a series of speech recognizers (Sphinx 2 - 4) and an acoustic model trainer (SphinxTrain). The code is available open source for download and use. Another popular speech development toolkit is Hidden Markov Model Toolkit (HTK), also open source. Some studies compare the speech recognition performance of the two toolkits (Kačur, 2006, Ma et al., 2009). They generally conclude that similar systems developed with the two toolkits have a similar performance, but the acoustic modelling performed by Sphinx is slightly better.

4.1 Speech Recording

A speech database comprising recorded audio files is required to build an acoustic model for the speech recognition system. An online speech recorder application, developed by SpeeD research group, is used to record the audio files. Several speakers recorded predetermined audio messages, containing multiple groups of random digits. For the initial training of the speaker-depended system, 100 audio clips were recorded. Each clip contains 12 uttered digits. Integrated laptop microphones were avoided and recording volume was set to high. An initial recording calibration was required (to detect background noise). Figure 6 presents the GUI for the recording application.

Fig. 6. Speech recording application.

4.2 Phonetic dictionary

A phonetic dictionary is a linguistic tool that specifies how to pronounce words in a language. In other words, a phonetic dictionary makes the correspondence between writing and phonetic form of words in a language. In a continuous speech recognition system, a phonetic dictionary is intended to link the acoustic model (which models how to produce language-specific sounds) and language model (which models the succession of words in a language). As a result, the phonetic dictionary should contain all possible words for the given recognition task and, of course, a phonetic transcriptions of these words.

For the current task (digits recognition from recorded audio waves), the phonetic dictionary must contain transcriptions for only the ten digits of the decimal system: *zero, unu, doi, trei, patru, cinci, şase, şapte, opt and nouă* (including Romanian diacritics).

4.3 Acoustic model training

Training the acoustic model requires the following resources:

- audio waves containing speech (previously recorded using the speech GUI on the SpeeD server);
- corresponding textual transcription of the words spoken in the audio waves;
- a phonetic dictionary containing all the words (the dictionary mentioned in the previous paragraph);
- a dictionary with acoustic elements that are not phonemes usually called *fillers* (silence, cough, laugh, music, etc.).

From all the 100 audio waves recorded per speaker, the first 50 and the last 30 file are used for training. The rest will be used for the evaluation of the system.

4.4 Creating a language model (java grammar)

Current systems, trained with a large vocabulary for speech recognition, use an n-gram statistical language model. These language models are built based on large text corpora, specific for the recognition task, estimating the probability of occurrence for words and sequences of words for that task. The n-gram language models are then used in the process of decoding (speech recognition) to select the most likely sequence of words proposed by the acoustic model.

The task of recognizing audio sequences containing digits is a limited vocabulary recognition scenario, which is not suitable for a statistical language model. Furthermore, digits and the succession of digits in the recorded audio clips appear approximately with equal probability (one cannot say that a digit is used systematically more often than the other).

In these circumstances, a finite state grammar (FSG) model is more suitable. A finite state grammar is a graph model in which the nodes represent the language words, and transitions between words are the arcs of the graph. This type of language model explicitly specifies all sequences of words allowed by the recognition task. Moreover, each arc may be assigned a cost specifying the probability that a word is preceded by another (in other words, the probability of the two sequences of words). Figure 7 shows the finite state grammar of our recognition task. 14 nodes make up the model, with only 10 nodes representing the digits. The other four are used for "entering" and "leaving" the graph, respectively for a back trace transition. The transitions show the way to trace this grammar and the word sequences allowed: in every audio clip one or more digits can be spoken.

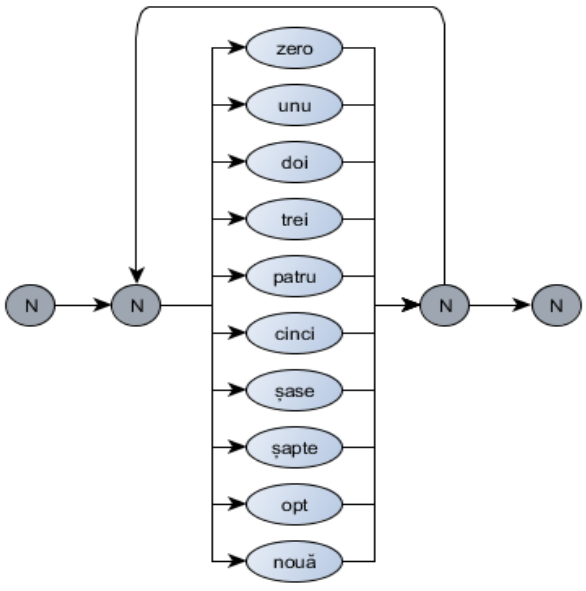

Fig. 7. Finite state grammar for digits task.

4.5 Decoding

The three basic components of a speech recognition system (acoustic model, language model and phonetic model), mentioned in previous subsections, are available for use in the decoding process. As a result, the system can now decode using the evaluation data, and then compare the textual transcription of the decoding process with the reference transcription. A corresponding report file is generated, with statistics and alignment details.

5. EVALUATION SETUP

To build up the connected-digits recognition system, 90 speakers were used to build "roDigits" database (Table 1), comprising of 100 audio files per speaker, with a total of 20 hours of recorded speech. The phrases contain 12 spoken digits, in arbitrary order. From the audio files, 80% were used for training, and the rest for evaluation.

Table 1. Speaker database summary.

Database name:	roDigits
Hours of speech:	20
Number of speakers:	90
Speaker ID:	1 - 90
Audio files per speaker:	100
Training files:	80%
Evaluation files:	20%

Phonemes were modelled in a context-dependent manner. To study the effects of increasing/decreasing the number of senones and Gaussian mixtures per senone, they were varied, according to Table 2.

Table 2. Number of senones and GMMs summary.

Test	No of senones	GMMs
Speaker dependent	100	1/2/4/8/16/32/64/128/256
	200	1/2/4/8/16/32/64/128/256
Speaker independent	100	1/2/4/8/16/32/64/128/256/512

For all ASR experiments presented in this work, we proposed in Table 3 the evaluation setups and their corresponding ids, along with the training and evaluation files used for each setup. These setups where chosen to highlight the importance of training a speaker independent system to avoid mismatch (speakers that are not in the training database), and to select the optimum number of senone states and Gaussian densities, for each task.

The following tests were conducted, for both speaker dependent / independent acoustic models:

Speaker dependent:

- *EvalDepSame1*, *EvalDepSame2* and *EvalDepSame3* setups were especially created to evaluate the performance of training and decoding with the same speaker (speaker dependent), and choose the optimum number of senones for the next experiments (100 / 200). The tests are identical for all 3 randomly chosen speakers, to validate the results.
- *EvalDepRest1*, *EvalDepRest2* and *EvalDepRest3* use the previously trained models to decode speech from the rest of the roDigits database, to emphasize the mismatch between this model and unknown speech.

Speaker Independent:

- *EvalIndepSame* proposes a model trained with 60 speakers from the roDigits database, to compensate for the high WER with the previous speaker dependent trained models.
- *EvalIndepRest* evaluates the speaker independent model, previously obtained, with the remaining 30 speakers from the roDigits database.

Table 3. roDigits evaluation setup.

Evaluation setup	Training files	Evaluation files	Setup id
Speaker Dependent	Speaker ID 1	Speaker ID 1	EvalDepSame1
	Speaker ID 59	Speaker ID 59	EvalDepSame2
	Speaker ID 82	Speaker ID 82	EvalDepSame3
	Speaker ID 1	rest of roDigits	EvalDepRest1
	Speaker ID 59	rest of roDigits	EvalDepRest2
	Speaker ID 82	rest of roDigits	EvalDepRest3
Speaker Independent	Speaker ID 1 - 60	Speaker ID 1 - 60	EvalIndepSame
	Speaker ID 1 - 60	Speaker ID 61 - 90	EvalIndepRest

If the speech recognition problem is posed as the transformation of an acoustic signal to a single stream of words, then there is widespread agreement on word error rate (WER) as the appropriate evaluation measure. The sequence of words output by the speech recognizer is aligned to the reference transcription using dynamic programming. The industry standard SCLITE (NIST, 2014) application is used for scoring and evaluating the output of the system. SCLITE is part of the NIST SCTK Scoring Toolkit. The program compares the hypothesized text (HYP) output by the speech recognizer to the correct, or reference (REF) text. After aligning REF to HYP, statistics are gathered during the scoring to output a performance report.

An example report is further detailed:

```
SENTENCE RECOGNITION PERFORMANCE

 sentences                      20
 with errors                  5.0%   (1)
   with substitions           0.0%   (0)
   with deletions             5.0%   (1)
   with insertions            0.0%   (0)

WORD RECOGNITION PERFORMANCE

Percent Total Error       =    0.4%   (1)
Percent Correct           =   99.6%   (239)
Percent Substitution      =    0.0%   (0)
Percent Deletions         =    0.4%   (1)
Percent Insertions        =    0.0%   (0)
Percent Word Accuracy     =   99.6%

DUMP OF SYSTEM ALIGNMENT STRUCTURE
........
id: (354-354_10_0056)
Scores: (#C #S #D #I) 12 0 0 0
REF:  unu nouă şase şase patru trei trei doi patru doi unu
şase
HYP:  unu nouă şase şase patru trei trei doi patru doi unu
şase
Eval:
```

Word error rate (WER), sentence error rate (SER) can be consulted in the report, along with a comparison between the reference transcription and the hypothetical transcription of the decoded speech. Number of substitutions, deletions and insertions are also shown, along with detailed information about the substituted words, deleted words, etc. The accuracy of the speech recognizer may then be estimated as the string edit distance between the output and reference strings. If there are N words in the reference transcript, and alignment with the speech recognition output results in S substitutions, D deletions, and I insertions, the word error rate is defined as:

$$WER[\%] = \frac{I + S + D}{N} \cdot 100 \tag{6}$$

Sometimes the word error rate can be greater than 100% because the above equation also includes the number of insertions. In some applications, a second evaluation metric, the sentence error rate (SER), might also be important depending on the application. The sentence error rate is based on the word error rate and can be computed as follows:

$$SER[\%] = \frac{Sentences\ with\ at\ least\ one\ error}{Sentences\ in\ the\ reference\ transcription} \cdot 100 \tag{7}$$

6. EVALUATION RESULTS AND DISCUSSION

6.1 Speaker dependent models

a) The following set of results were obtained using a single speaker trained model, with *EvalDepSame1*, *EvalDepSame2* and *EvalDepSame3* setups.

Tables 4 and 5 presents the results for the first trained speaker dependent model, *EvalDepSame1*. Figures 8 and 9 were plotted, to visually represent the results.

Table 4. WER for *EvalDepSame1*.

WER [%]		# GMMs								
		1	2	4	8	16	32	64	128	256
No. senone	100	2.9	0.4	0.0	0.4	0.8	0.4	2.1	22.5	80.4
	200	9.6	3.3	7.5	12.9	37.9	52.5	77.5	85.8	85.4

Table 5. SER for *EvalDepSame1*.

SER [%]		# GMMs								
		1	2	4	8	16	32	64	128	256
No. senone	100	25.0	5.0	0.0	5.0	10	5.0	15.0	85.0	100.0
	200	40.0	25.0	35.0	50.0	80.0	95.0	100.0	100.0	100.0

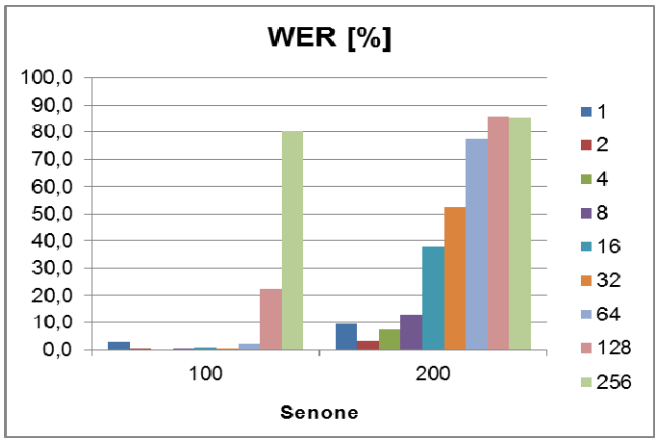

Fig. 8. Comparison of WER depending on the number of senones and GMMs, for *EvalDepSame1*.

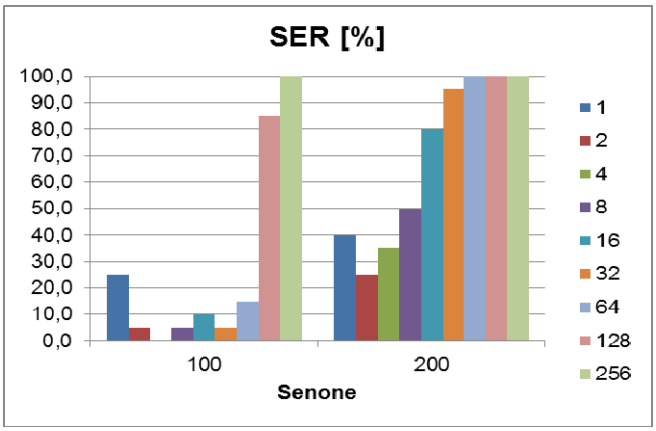

Fig. 9. Comparison of SER depending on the number of senones and GMMs, for *EvalDepSame1*.

Results are consistent with the rest of the randomly selected speakers, as shown in Tables 6, 7, 8 and 9, for *EvalDepSame2* and *EvalDepSame3* setups.

Table 6. WER for *EvalDepSame2*

WER [%]		# GMMs								
		1	2	4	8	16	32	64	128	256
No. senone	100	0.4	0.4	0.4	0.4	0.0	0.4	2.1	24.6	89.6
	200	2.9	8.3	9.2	23.8	29.2	40.8	60.0	75.0	94.6

Table 7. SER for *EvalDepSame2*

SER [%]		# GMMs								
		1	2	4	8	16	32	64	128	256
No. senone	100	5.0	5.0	5.0	5.0	0	5.0	20.0	95.0	100.0
	200	20.0	40.0	40.0	85.0	95.0	95.0	100.0	100.0	100.0

Table 8. WER for EvalDepSame3

WER [%]		# GMMs								
		1	2	4	8	16	32	64	128	256
No. senone	100	2.1	1.3	0.8	0.8	0.8	1.7	9.6	40.0	79.6
	200	3.8	3.8	5.8	7.5	14.6	41.3	70.0	80.0	87.9

Table 9. SER for EvalDepSame3

SER [%]		# GMMs								
		1	2	4	8	16	32	64	128	256
No. senone	100	20.0	10.0	5.0	5.0	5.0	15.0	65.0	100.0	100.0
	200	30.0	30.0	35.0	40.0	85.0	100.0	100.0	100.0	100.0

For the task of single speaker identification, 100 senones seems to be the best option, obtaining the best results between 4 and 16 GMMs, depending on the speaker. The more senones a model has, the better it discriminates sounds, and if a high number of senones are set (more than necessary), the model might not be universal enough to recognize unseen speech. WER will be higher on new data, so it is important not over-train the models. The test was run for 3 randomly selected speakers from roDigits, to validate the results.

It is interesting to see how this model, trained with only one speaker (speaker dependent), performs for new speakers, in terms of WER and SER. Next, the same model is used to decode the audio files in a larger data set, from multiple speakers, to evaluate its performance. Senone are set to 100, for the next experiments.

b) The following results were obtained using the previously single speaker trained model, but the decoding was done with the rest of roDigits database, not part of the training process (*EvalDepRest1*, *EvalDepRest2* and *EvalDepRest3*). Tables in this section present the results. Figures 10 and 11 are plotted for comparison with results from subsection a).

Table 10. WER for *EvalDepRest1*

WER [%]	# GMMs								
	1	2	4	8	16	32	64	128	256
	68.2	74.0	69.4	72.6	73.6	73.9	86.7	94.4	98.8

Table 11. SER for *EvalDepRest2*

SER [%]	# GMMs								
	1	2	4	8	16	32	64	128	256
	97.0	97.0	98.1	98.6	98.7	98.7	98.9	99.8	100.0

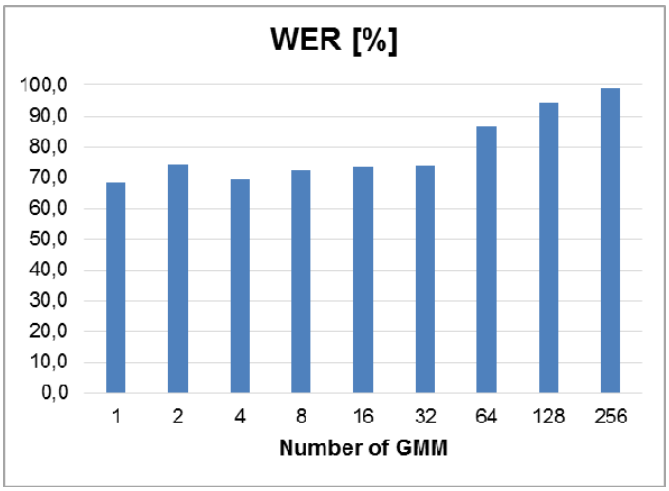

Fig. 10. Comparison of WER depending on the number of GMMs, for *EvalDepRest1*.

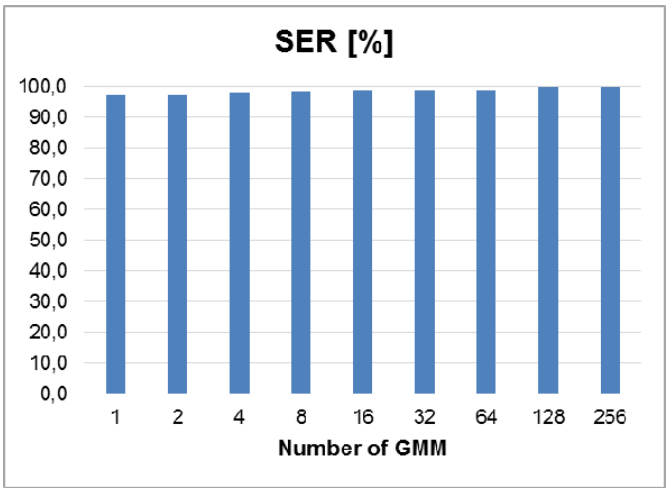

Fig. 11. Comparison of SER depending on the number of GMMs, for *EvalDepRest1*.

Table 12. WER for *EvalDepRest2*

WER [%]	# GMMs								
	1	2	4	8	16	32	64	128	256
	57.8	59.9	54.9	60.4	62.7	66.8	79.7	89.4	98.0

Table 13. SER for *EvalDepRest2*

SER [%]	# GMMs								
	1	2	4	8	16	32	64	128	256
	87.0	85.6	86.7	89.7	92.0	95.8	99.6	100.0	100.0

Table 14. WER for *EvalDepRest3*

WER [%]	# GMMs								
	1	2	4	8	16	32	64	128	256
	65.5	58.4	59.0	65.5	66.0	77.2	81.7	89.5	99.0

Table 15. SER for *EvalDepRest3*

SER [%]	# GMMs								
	1	2	4	8	16	32	64	128	256
	92.0	95.8	97.6	98.0	99.6	99.8	100.0	100.0	100.0

As expected, results show a big difference in error rate, when using multiple speakers for decoding. Results are consistent for all the 3 models trained with data from one speaker, and show a weaker recognition rate in identifying utterances from multiple speakers. A model trained with only one speaker (speaker dependent) cannot be successful in decoding utterances from multiple speakers. A different model must be constructed, with a larger training dataset.

6.2 Speaker independent model

A speaker independent ASR system requires a bigger database for the training process. For this purpose, 60 speakers, from roDigits, are used for the training process. Multiple tests were conducted: after training, a different set of audio waves are used for decoding. The results are compared afterwards with a different batch of 30 speakers, which were not used in the training process, to evaluate the performance of decoding unseen speakers.

a) Results from *EvalIndepSame* evaluation setup, trained with multiple speakers, to compensate for the high error rate the previous models offered. The evaluation is done with the same speakers, used for training.

Table 16. WER for *EvalIndepSame*

WER [%]	# GMMs									
	1	2	4	8	16	32	64	128	256	512
	10.0	4.9	3.3	2.2	1.8	1.4	1.1	0.9	0.6	0.5

Table 17. WER for *EvalIndepSame*

SER [%]	# GMMs									
	1	2	4	8	16	32	64	128	256	512
	53.1	36.5	26.8	18.7	16.1	12.2	10.4	9.3	7.9	5.6

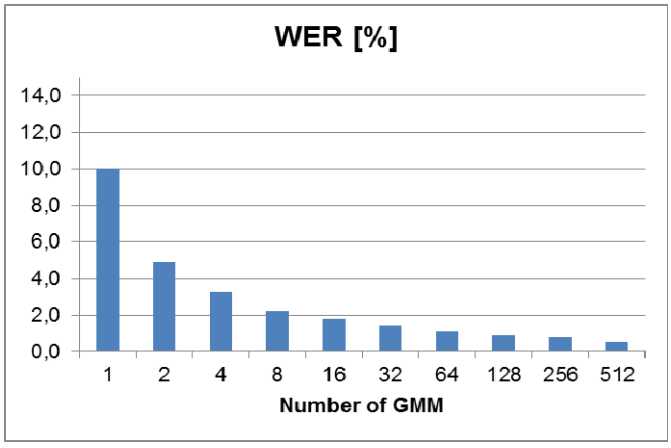

Fig. 12. Comparison of WER depending on the number of GMMs, for *EvalIndepSame*.

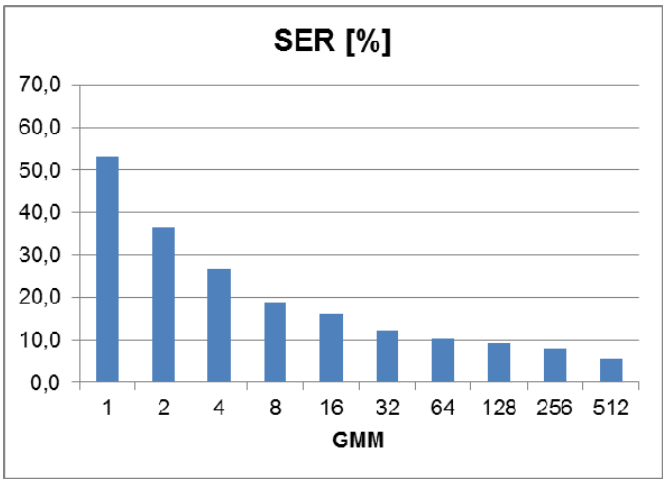

Fig. 13. Comparison of SER depending on the number of GMMs, for *EvalIndepSame*

The speaker-independent ASR system obtains better results (a lower WER) than the speaker-dependent ASR system. This means that speakers that are contained in the training database are better recognized. In general, the little the mismatch (be it speaker, environment, encoding, etc.) between the training and the evaluation data, the better the results. The best WER is obtained for around 256-512 GMM densities, and there is not much incentive to go further, as the results are in the error interval and the training and decoding time does not justify the gained improvements. These speech recognition results were for "known speakers" (speakers which were also part of the training process), but they might not be as good for "unknown speakers" (speakers to which the system was not exposed during training).

Consequently, the next experiment aimed to evaluate the speaker-independent ASR system on speech uttered by 30 other speakers, which were not part of the training batch.

b) Using the previously speaker independent training data, the decoding is done with the rest of the 30 roDigits speakers to evaluate the model performance. The setup is *EvalIndepRest*, described in table 3.

Table 18. WER for *EvalIndepRest*

WER [%]	**# GMMs**									
	1	**2**	**4**	**8**	**16**	**32**	**64**	**128**	**256**	**512**
	20.1	13.8	11.3	10.4	9.4	9.0	8.0	7.6	7.2	7.3

Table 19. SER for *EvalIndepRest*

SER [%]	**# GMMs**									
	1	**2**	**4**	**8**	**16**	**32**	**64**	**128**	**256**	**512**
	65.3	48.7	40.0	33.3	24.8	25.2	24.2	24.2	24.0	26.5

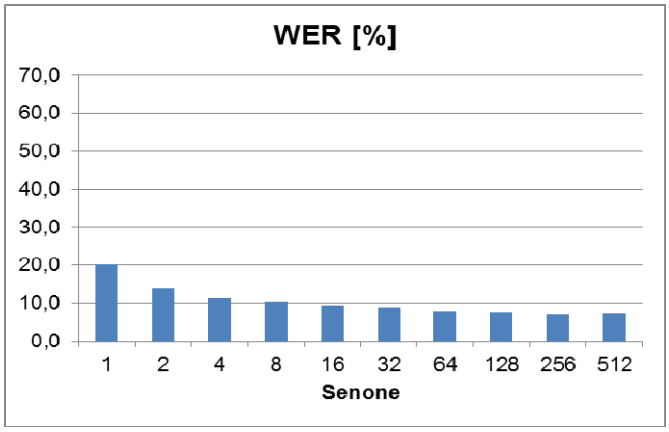

Fig. 14. Comparison of WER depending on the number of GMMs, for *EvalIndepRest*

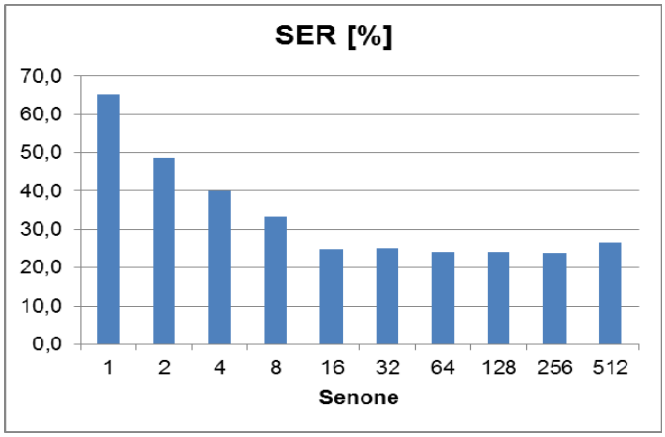

Fig. 15. Comparison of SER depending on the number of GMMs, for *EvalIndepRest*

As the above results showed, to obtain a connected-digits recognition system in Romanian language, to successfully recognize unseen speakers, the acoustic model needs to be trained against a larger set of speakers. The number of senones and the number of Gaussian mixtures per senone are variables of the system, to be optimized based on each specific database. For this last experiment, 256 GMMs offers the best results, in terms of error rate, after which more densities cannot successfully model the output distributions, requiring a more detailed model. Depending on the desired error rate, smaller GMMs can be used, which offer faster decoding speed.

Other interesting observations can be taken by looking at word confusion pairs, for this last experiment:

No. of confusions	Confusion pair
5	șapte ==> șase
3	șase ==> șapte
2	nouă ==> unu

The 7 digit (şapte) is often confused with 6 (şase), as only two phones are different in the phonetic transcription. This mistake is usually made by human speakers also, and the language model can be improved by including, then training the acoustic model, to contain the alternate „şepte" spelling. This alternate spelling is used in numerous telephone conversations, to avoid confusion between these two digits.

7. CONCLUSIONS

This paper presented the processes involved in building a fairly representative speech recognition system for decoding connected digits, using speech recorded from multiple speakers to train and evaluate the system. The paper offered a quick overview of the processes involved in ASR, and in particular, the trainable hidden Markov/Gaussian mixture model (HMM/GMM), for acoustic modelling. Information for improving the models and the training set, along with decreasing word error rate (the primary evaluation metric) are provided. The proposed ASR system can be used in commercial applications, to recognise connected-digits from multiple speakers. An example would be automatic recognition of National Identification Number (CNP – "Cod Numeric Personal") for certain applications. The commercial success of these speech recognition systems in general, is an impressive testimony to how far research in ASR has come.

The work has been funded by the Sectoral Operational Programme Human Resources Development 2007-2013 of the Ministry of European Funds through the Financial Agreements POSDRU /159 /1.5/ S/ 132395, and POSDRU /159 /1.5/S/134398.

REFERENCES

Burileanu D., Fecioru A., Ion D., Stoica M., Ilas C. (2004), An Optimized TTS System Implementation Using a Motorola StarCore SC140-Based Processor, *Proc. of ICASSP 2004*, Montreal, Vol. 5, pp. 317-320, 2004, ISSN: 1053-587X.

Chernakova S., Nechaev A., Karpov A. and Ronzhin A. (2006). Assistive Multimodal Interface for Medical Applications. *Proc. of the SPECOM'2006 Conf*.

Cohen, P.R., Johnston, M., McGee, D., Oviatt, S. L., Clow, J., and Smith, I. (1998). The efficiency of multi-modal interaction: a case study. *In ICSLP-98*, Sydney, Vol. 2, pp. 249-252.

Cucu, H. (2011). *Towards a speaker-independent, large-vocabulary continuous speech recognition system for Romanian,* Ph.D. Thesis, Universitatea Politehnica din Bucureşti, Romania, http://speed.pub.ro/speed3/wp-content/uploads/2013/02/2011HoriaCucuThesis.pdf

Dumitru C. O., Gavat I. (2008). Progress in Speech Recognition for Romanian Language, *Advances in Robotics, Automation and Control*, pp. 472, Vienna, Austria

Dahl G. E., Yu D., Deng L. and Acero A. (2012). Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Trans. Audio Speech Lang. Processing*, vol. 20, no. 1, pp. 30–42, 2012.

Furui, S. (2009). 40 Years of Progress in Automatic Speaker Recognition, *Proc. of the ICB 2009 Conf,* 1050-1059.

Gravano A., Jansche M., Bacchiani M. (2009). Restoring punctuation and capitalization in transcribed speech, *in ICASSP 2009*.

Huang X., Alleva F., Hwang M., Rosenfeld R. (1993). An overview of the SPHINX-II speech recognition system. *HLT '93 Proceedings of the workshop on Human Language Technology*.

Juang, B.H, Rabiner, L. (2005). Automatic Speech Recognition - A Brief History of the Technology Development, *Elsevier Encyclopedia of Language and Linguistics*.

Jurafsky, D., Martin J. H. (2008). *Speech and Language Processing, 2nd Edition*, Ch. 9. Prentice Hall, USA.

Kačur J. (2006). *HTK vs. SPHINX for SPEECH Recognition*, Department of telecommunication, FEI STU, Bratislava Slovakia.

Lamere, P., Kwok, P., Gouvêa. E., Raj, B., Singh, R., Walker W., Warmuth. M., Wolf, P. (2003). The CMU SPHINX-4 Speech Recognition System. *Proc. of ICASSP 2003*.

Ma, G., Zhou W., Zheng J., You X and Ye W. (2009). A Comparison between HTK and SPHINX on Chinese Mandarin. *In Artificial Intelligence, 2009. JCAI '09*, pp 394 – 397.

Microsoft, Skype Translator Add-on, (2014). Online, at Microsoft Worldwide Partner Conference - WPC 2014, https://www.youtube.com/watch?v=cJIILew6l28

NIST, Speech Recognition Scoring Toolkit (SCTK), 2014. Online, National Institute of Standards and Technology, http://www.nist.gov/itl/iad/mig/tools.cfm

Militaru D., Gavat I. (2014). A Historically Perspective of Speaker-Independent Speech Recognition in Romanian Language, *Sisom & Acoustics 2014*, Bucharest, Romania.

Petrică L., Cucu H., Buzo A. and Burileanu C. (2014). A Robust Diacritics Restoration System using Unreliable Raw Text Data, *SLTU-2014*, St. Petersburg, Russia.

Price J. and Eydgahi A. (2006). Design of Matlab®-Based Automatic Speaker Recognition Systems. *In the 9th International Conference on Engineering Education*.

Rabiner L. R., Schafer R. W. (2007). Introduction to Digital Speech Processing, *Foundations and Trends in Signal Processing*, Vol. 1, Nos. 1–2 (2007) 1–194.

Renals, S., Hain, T. (2013). Ch. 12: Speech Recognition. In Alexander Clark, Chris Fox, Shalom Lappin, *The Handbook of Computational Linguistics and Natural Language Processing,* 300-332. Wiley-Blackwell, UK.

Waibel A., Hanazawa T., Hinton G., Shikano K. and Lang K. J., (1989). Phoneme recognition using time-delay neural networks. *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 37, pp. 328-339.

Wu J. and Chan C. (1993). Isolated Word Recognition by Neural Network Models with Cross-Correlation Coefficients for Speech Dynamics. *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, pp. 1174-1185.

Zahorian S. A., Zimmer A. M. and Meng F. (2002). Vowel Classification for Computer based Visual Feedback for Speech Training for the Hearing Impaired. *In ICSLP 2002*.