EVALITA-ISTC COMPARISON OF OPEN SOURCE TOOLS ON CLEAN AND NOISY DIGITS RECOGNITION TASKS

Piero Cosi, Mauro Nicolao
Istituto di Scienze e Tecnologie della Cognizione, C.N.R.
Via Martiri della libertà, 2 – 35137 Padova (ITALY)
piero.cosi@pd.istc.cnr.it, mauro.nicolao@pd.istc.cnr.it

1. ABSTRACT

EVALITA is a recent initiative devoted to the evaluation of Natural Language and Speech Processing tools for Italian. The general objective of EVALITA is to promote the development of language and speech technologies for the Italian language, providing a shared framework where different systems and approaches can be evaluated in a consistent manner. In this work the results of three open source ASR toolkits (CSLU Speech Tools, CSLR SONIC, SPHINX) working on the EVALITA clean and noisy digits recognition task will be described together with the complete evaluation methodology.

2. INTRODUCTION

EVALITA[1] provides a shared framework for the evaluation of different systems and approaches on separate natural language and speech processing tasks for the Italian language. In EVALITA 2009 the following tasks have been evaluated.

- Text tasks: PoS-Tagging, Parsing, Lexical Substitution, Entity Recognition, Textual Entailment
- Speech tasks: Connected Digits Recognition (clean and noisy), Dialogue System Evaluation and Speaker Identity Verification (Application & Forensic)

Regarding the Connected Digits Recognition task, systems are required to recognize sequences of spoken Italian digits (numbers ranging from 0 to 9). Two subtasks are defined, and applicants may choose to participate in any of them:

- Clean digits: in this subtask, the test digits sequences are acquired in clean environment;
- Noisy digits: in this subtask, the test digits sequences are acquired in noisy environment. The type of noise may vary from white noise to traffic, room, etc.

The Corpus consists of digit sequences coming from various Italian Corpora. It has been divided into training, development and test. Furthermore, it has been split into clean and noisy speech, according to the presence of many noise events in the signal or low SNR. Audio files are sampled at 16kHz, 16 bit PCM, mono, Windows wav format. For training and development set, transcription at word level has been provided in two separate text files, containing on each row the audio filename followed by the transcription.

The evaluation process is based on Minimum Edit Distance between the transcriptions coming out from the recognizer and the orthographic annotations. Accuracy will be calculated at word and phrase levels and participants which need to enrol the ASR at finer level than phrase have to provide by themselves for the annotation.

Test data have been provided only for one week before the final evaluation workshop in the form of audio files and two lists of filenames (clean and noisy) and test results have

[1] EVALITA web site: http://evalita.fbk.eu/

been provided with the ASR participant system name, the list of filenames and the corresponding recognition results.

2.1 Evaluation Metric

With respect to the results submitted by the participants measurements of Word Accuracy and Sentence Accuracy have been considered.

Word Accuracy is defined as:

$$WA = 100 - \frac{I + S + D}{N} \times 100 \quad (1)$$

where,

- I is the number of inserted words
- S is the number of substitutions
- D is the number of the deletions
- N is the number of words in the reference

Sentence Accuracy: is defined as:

$$SA = \frac{H}{M} \times 100 \quad (2)$$

where,

- H is the number of sentences correctly recognized
- M is the number of sentences in the reference

For this paper's purpose, we focused on Word Accuracy.

3. OPEN SOURCE ASR TOOLS

Three of the most used open source ASR tools were considered in this work, i.e. CSLU Toolkit, SONIC, and SPHINX, simply because promising results were obtained in the past on similar digit recognition tasks.

3.1 CSLU TOOLKIT

The CSLU Toolkit is a comprehensive set of tools for learning about, researching and developing interactive language systems and their underlying technologies. The CSLU Toolkit has been described in several papers (Cole 1999, Sutton *et alii*, 1998) and will not be detailed here[2]. The basic framework of the CSLU Toolkit is represented by an hybrid Hidden Markov Model (HMM) and Artificial Neural Network (ANN) architecture in which the phonetic likelihoods are estimated using a neural network instead of a mixture of Gaussians, which has the advantage of not requiring assumptions about the distribution or independence of the input data, of easily performing discriminative training and of splitting each phoneme into states that are dependent on the left or right context, or are context independent (Bourlard, 1995).

As for feature extraction MFCC, MFCC+Δ, MFCC+Δ +Δ^2 and PLP+MFCC, added by Cepstral Mean Subtraction (CMS) and RASTA processing at 10-msec frame rate, were compared. The input to the network consisted of the features for the frame to be classified, as well as the features for frames at -60, -30, 30, and 60 msec relative to the frame to be

[2] The CSLU Toolkit is available through the CSLU OGI Web site: http://CSLU.cse.ogi.edu/toolkit/.

classified. As an example, in the case of 12 MFCC coefficients plus the energy plus their delta values, the network consisted of 130 input nodes.

Neural-network training was done with standard back-propagation on a fully connected feed-forward network. The training data were searched to find all the vectors of each category in the automatically-labeled training section. The neural network was trained using the back-propagation method to recognize each context-dependent category in the output layer. Each training waveform was then recognized using the best obtained network (baseline), with the result constrained to be the correct sequence of digits. This process, called "forced alignment", was used to generate time-aligned category labels. These force-aligned category labels were then used in a second cycle of training and evaluation was repeated to determine the new best network ("force aligned" network - FA), which was finally evaluated with the same development data.

In order to explore the possibility to further improve the recognition results, the "forward-backward" (FB) training strategy was recurrently applied (three times) (Yan *et alii*, 1997). Like most of the other hybrid systems, the neural network in this system is used as a state emission probability estimator. A three-layer fully connected neural network can be conceived, with the same configuration as that of the baseline and forced-aligned neural networks and the same output categories. Unlike most of the existing hybrid systems which do not explicitly train the within-phone relative likelihoods, this new hybrid trains the within-phone models to probability estimates obtained from the forward-backward algorithm, rather than binary targets. To start FB training an initial binary-target neural network is required. For this initial network, the best network resulting from forced-alignment training (FA) was used. Then the forward-backward re-estimation algorithm was used to regenerate the targets for the training utterances. The re-estimation was implemented in an embedded form, which concatenates the phone models in the input utterance into a "big" model and re-estimates the parameters based on the whole input utterance. The networks would be trained using the standard stochastic back-propagation algorithm, with mean-square-error as the cost function.

3.2 CSLR SONIC

SONIC[3] is a complete toolkit for research and development of new algorithms for continuous speech recognition. The software has been under development at CSLR since March of 2001 at the University of Colorado. It has been written in C and can be executed in multi-process mode.

The recognizer toolkit consists of a core speech recognition engine and programming interface (API). The current implementation allows for two modes of speech recognition:

- Keyword / Grammar Decoding – continuous speech recognition constrained by a finite-state grammar. This mode also allows for keyword and grammar spotting capabilities.
- N-gram Decoding – speech recognition based on statistical n-gram language models.

SONIC is based on continuous density hidden Markov model (CDHMM) technology. The acoustic models are decision-tree state-clustered HMMs with associated gamma probability density functions to model state-durations. SONIC incorporates speaker adaptation and normalization methods such as Maximum Likelihood Linear Regression

[3] SONIC was available through the CSLR Web site: http://sonic.colorado.edu/sonic/download/index.html.

(MLLR), Vocal Tract Length Normalization (VTLN), and cepstral mean and variance normalization.

CSLR has developed an acoustic feature representation known as Perceptual Minimum Variance Distortionless Response (PMVDR) cepstral coefficients (Umit *et alii*, 2003). These features are now the default feature representation in SONIC, version 2.0-beta3. PMVDR cepstral coefficients provide improved accuracy over traditional MFCC parameters by better tracking the upper envelope of the speech spectrum. Unlike MFCC parameters, PMVDRs do not require an explicit filterbank analysis of the speech signal. We have found this new feature representation to not only provide noise robust speech recognition, but also improved accuracy in cleaner speech tasks. A block diagram of the PMVDR feature extraction process is shown in Fig. 1 [see also (Pellom and Hacioglu, 2004) for details].

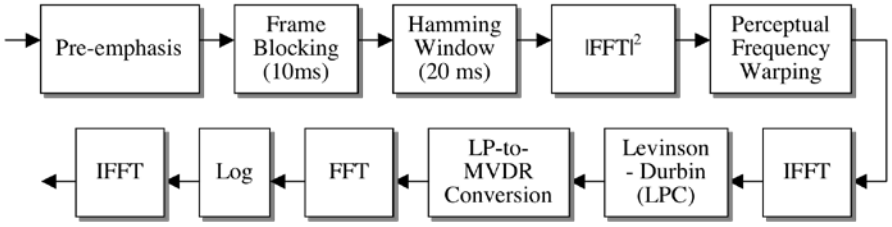

Fig. 1. PMVDR feature extraction process.

PMVDRs are calculated by first pre-emphasizing the speech signal prior to frame blocking and windowing. The power spectrum is warped onto a Mel frequency scale and passed through an inverse FFT to obtain perceptual autocorrelation coefficients. Next, LPC analysis is performed and the resulting LP coefficients are used to compute a perceptually warped MVDR spectrum of the signal (Murthi and Rao, 1999). The log-spectrum is computed prior to taking the inverse FFT to obtain the PMVDR cepstral coefficients. In total, 12 PMVDR cepstral parameters are retained and augmented with normalized log frame energy plus the first and second differences of the features. A final 39-dimensional feature vector is computed, once every 10 ms.

The acoustic modeling for the recognizer consists of decision tree state-clustered continuous density HMMs. The acoustic models have a fixed 3-state topology. Each HMM state can be modeled with a variable number of multivariate mixture Gaussian distributions. The acoustic model trainer uses the Viterbi algorithm for model estimation. This substantially reduces the amount of CPU effort needed to train acoustic models compared with forward-backward training methods. The training process therefore consists of first performing state-based alignment of the training audio followed by an expectation-maximization (EM) step in which decision tree state-clustered HMMs are estimated.

Acoustic model parameters (means, covariances, and mixture weights) are estimated in the maximum likelihood sense. The training process can be iterated between alignment of data and model estimation to gradually achieve adequate parameter estimation.

Support is provided for standard word-based and class-based backoff n-gram language models.

Currently unigram, bigram, trigram, and fourgram models can be applied during the first-pass of recognition. SONIC can process language models that have been estimated

using both the CMU/Cambridge Statistical Language Modeling Toolkit[4] and the SRI language modeling toolkit[5]. Support for finite-state regular grammar based speech recognition is also provided.

3.3 CMU SPHINX

SPHINX system (Lee *et alii*, 1990)[6] is an open-source project which provides a complete set of functions to develop complex Automatic Speech Recognition systems. This software has been developed by Carnegie Mellon University at Pittsburgh and its license is derived from BSD with no restriction against commercial use or redistribution. It includes both an acoustic *trainer* and various *decoders*, *i.e.*, text recognition, phoneme recognition, N-best list generation, etc.

The training step creates the statistical relationships between audio spectrum parameterization and phonemes. The Acoustic Model (AM) is trained from acoustic training data using the SPHINX trainer. This is capable of building acoustic models with a wide range of structures, such as *discrete*, *semi-continuous*, or *continuous*. An aligned corpus of audio and transcriptions is needed. First iteration consists of an equally spaced audio segmentation according to transcriptions and a first raw AM training. Then this AM can be used as a starting point for several loops of Baum-Welch probability density functions (PDF) estimation and alignment so that every alignment output segmentation will be more accurate than the previous one.

In the decoding step, the trained models are used to analyzed new audio speech files. While a unique toolkit for training Acoustic Model is provided, there are different versions of the recognizer: one is specifically designed for embedded applications (Pocket SPHINX), another is a continuous model C-version (SPHINX-3) and finally there is a Java one, designed for Web applications (SPHINX-4). Our choice was to use SPHINX-3, in order to better merge these tools with our test framework.

SPHINX-3 is the CMU's state-of-the-art large vocabulary speech recognition system. The following is a brief summary of its main features and limitations:

- 5-10x real-time recognition on large vocabulary tasks;
- limited to fully continuous acoustic models
- limited to 3 or 5-state left-to-right HMM topologies
- bigram or trigram language model
- the decoder cannot handle arbitrary lengths of speech input; each separate piece (or *utterance*) to be processed by the decoder must be no more than 300 sec. long; typically, one uses a *segmentation tool* to chop up a cepstrum stream into manageable segments of up to 20 or 30 sec. duration.

The SPHINX-3 decoder is based on the conventional *Viterbi search* algorithm and *beam search* heuristics, and it uses a *lexical-tree* search structure, too. It takes its input from pre-recorded speech in raw PCM format and writes its recognition results to output text files.

We first give a brief outline of the input and output characteristics of the decoder.

- *Lexical model*: the lexical or pronunciation model contains pronunciations for all the words of interest to the decoder. Like most modern speech recognition systems,

[4] The CMU/Cambridge Statistical Language Modeling Toolkit is available at http://www.speech.cs.cmu.edu/SLM_info.html

[5] The SRI language modeling toolkit is available at http://www.speech.sri.com/projects/srilm/.

[6] The SPHINX system is available at http://cmuSPHINX.sourceforge.net/html/cmuSPHINX.php.

SPHINX-3 uses *phonetic units* to build word pronunciations. Currently, the pronunciation lexicon is almost entirely hand-crafted.

- *Acoustic model*: SPHINX uses acoustic models based on statistical *Hidden Markov Models* (HMMs) with continuous output probability density functions.
- *Language model (LM)*: SPHINX-3 uses a conventional backoff bigram or trigram language model.
- *Speech input specification*: this distribution contains four executable files. We use *SPHINX3_decode* which decodes in batch mode processing. The entire input to be processed must be available beforehand, *i.e.*, the raw audio samples must have been pre-processed into cepstrum files..

The decoder can produce two types of recognition output:

- *Recognition hypothesis:* a single best recognition result (or hypothesis) for each processed utterance. This is a linear word sequence, with additional attributes such as their time segmentation and scores.
- *Word lattice*: A word-graph of all possible candidate words recognized during the decoding of an utterance, including other attributes such as their time segmentation and acoustic likelihood scores.

Models can be computed either for each language phoneme or, considering phoneme context, for sequence of phoneme (usually three). SPHINX acoustic models are trained over MFCC + Δ + Δ^2 feature vectors. MFCCs are the classical perceptual parameterization of audio speech signal spectrum which enhances the most acoustically relevant spectrum band.

3.4 NIST SCLITE

In order to compute the evaluation score, according to EVALITA rules, we used the NIST sclite software. This is part of the NIST SCTK Scoring Toolkit[7] and it is a tool for scoring and evaluating the output of speech recognition systems. The program compares the hypothesized text (HYP) output by the speech recognizer to the correct, or reference (REF) text. After comparing REF to HYP, (a process called alignment), statistics are gathered during the scoring process and a variety of reports can be produced to summarize the performance of the recognition system.

Sclite can use either of two algorithms for finding alignments between reference and hypothesis word strings. The first, and most widely accepted, uses dynamic programming (DP) and the second uses GNU's "diff", a FSF (Free Software Foundation) program for comparing text files. The DP string alignment algorithm performs a global minimization of a Levenshtein distance function which weights the cost of correct words, insertions, deletions and substitutions as 0, 3, 3 and 4 respectively.

When evaluating the output of speech recognition systems, the precision of generated statistics is directly correlated to the reference text accuracy. But uttered words can be co-articulated or mumbled to where they have ambiguous transcriptions. In order to more accurately represent ambiguous transcriptions, and not penalize recognition systems, the ARPA community agreed upon a format for specifying alternative reference transcriptions. For mumbled or quietly spoken words, the ARPA community agreed to neither penalize systems which correctly recognized the word, nor penalize systems which did not. To accommodate this, a NULL word, "@", can be added to an alternative reference transcript. The presence of alternate transcriptions represents added computational complexity to the DP algorithm.

[7] The NIST sclite software is available at the NIST website: http://www.itl.nist.gov/iad/mig/tools/

The computation of elements in (1) was made by the following sclite elements:

$$\text{Sub} = \frac{\text{Number of Substituted words}}{\text{Number of Reference words}} = \frac{S}{N}$$

$$\text{Ins} = \frac{\text{Number of Inserted words}}{\text{Number of Reference words}} = \frac{I}{N}$$

$$\text{Del} = \frac{\text{Number of Deleted words}}{\text{Number of Reference words}} = \frac{D}{N}$$

4. EXPERIMENTS

In this paragraph, we are going to describe the structures and the parameters of our experiments.

4.1 Evalita Data

First, the EVALITA data is described. As previously mentioned, EVALITA data is constituted by clean and noisy digits audio file sets. This is divided in three sub sets: train, development and test. First experiments were produced by train and development sets[8].

The first corresponds to 3144 and 2204 digit sequences (10129 and 7376 digits), the latter at the time of the writing of this paper, with the EVALITA clean and noisy digits development sets corresponding to 216 and 299 digit sequences respectively (1629 and 1941 digits).

The only pronounced words are the Italian digits: UNO, DUE, TRE, QUATTRO, CINQUE, SEI, SETTE, OTTO, NOVE, ZERO.

4.2 Experimental Framework

We have trained all our systems only on the EVALITA training data and we have tested them on the EVALITA development data. We decided to analyze clean and noisy data either separately and together by training different acoustic models for each type of audio data. Our test framework was made by three recognition processes with three acoustic models:

- all training file AM;
- only noisy training file AM;
- only clean training file AM.

We have used these models for clean, noisy and complete test set recognition.

One of the main difficulties to set an homogeneous test framework was to find similarity among the three ASR systems in order to choose comparable configurations. Every system has a completely distinct architecture and consequently the configuration parameters are hardly comparable.

Furthermore, language model (LM) role is crucial in ASR system. SONIC and SPHINX are 3-gram LM based and we were able to use the same standard ARPA LM. We have generated a large amount of text data (900000 sentences) containing random digit sequences and we processed it with CMU Statistical Language Modeling (SLM) Toolkit. The aim was to create rules without creating too much bias. On the other hand, CSLU toolkit is a finite state grammar based system. In order to compare SONIC and SPHINX results with CSLU, we have also performed SONIC and SPHINX recognitions with LM

[8] The final results on the EVALITA clean and noisy digits test set will be given during the presentation of this work at the ASRU workshop.

weight set to 0. This should simulate complete independence between connected digits but using words as basic recognition unit.

We did several experiments to find similar configuration and finally we decided to compare results produced by the best WA-score configurations. Results are shown in the next chapter.

The ASR pronunciation lexicon is the same for the three systems. It has the 10 different words shown above, plus 3 special fillers: begin and end sentence markers and silence identifier. The word phonetization derives from the following SAMPA transcription:

0 [dz E r o], 1 [u n o], 2 [d u e], 3 [t r E], 4 [k w a t r o],

5 [tS i n k w e], 6 [s E I], 7 [s E t e], 8 [O t o], 9 [n O v e].

Eventually, multi pronounce entries can be produced to model regional pronunciation differences.

The acoustic model for *CSLU recognizer* was trained on context-dependent categories to account for co-articulatory variations.

Within the EVALITA framework only the orthographic transcriptions are available thus our previously created general purpose recognizer (Cosi and Hosom, 2000) was used to create the phonetically aligned transcription files for each training and development digit sequence.

A simple grammar [<any> (<digit> [silence]) + <any>] allowing any digit sequence in any order, with optional silence between digits, was considered.

A three-layer fully connected feed-forward network was trained to estimate, at every frame, the probability of 98 context-dependent phonetic categories. These categories were created by splitting each Acoustic Unit (AU), into one, two, or three parts, depending on the length of the AU and how much the AU was thought to be influenced by coarticulatory effects. “silence” (.pau .garbage @br) and “closure” are 1-part units, “vowel” (i e E a O o u) is a 3-part unit, “unvoiced plosive” (t k) is 1- part right dependent unit, “voiced plosive” (d), “affricate” (dz tS), “fricative” (s v), “nasal” (n), “liquid retroflex”(r) and “glide” (w) are all 2-part units. AU states were trained for different preceding and following phonetic contexts, and some phonetic contexts were grouped together to form a broad-context grouping. The broad-context groupings were done based on acoustic-phonetic knowledge (see TABLE I).

TABLE I

GROUPINGS OF ACOUSTIC UNITS INTO CLUSTERS OF SIMILAR UNITS, FOR THE ITALIAN DIGITS TASK.

Group	Acoustic units in group	Description
$sil	.pau, .garbage @br	Silence
$pld	d t tcl	dental plosive
$alv	dz s	Alveolar
$lab	v	Labial
$pal	tS	Palatal
$ret	r	Retroflex
$nas	n	Nasal
$vel	k kcl	Velar
$bck	u o O w	back vowel/glide
$mid	a E	mid vowel
$frn	i, e	front vowel

Training was done for 100 iterations, and the "best" network iteration ("baseline" network - B) was determined by evaluation on the EVALITA clean and noisy digits development sets respectively.

SONIC system also needs a good phonetic alignment to start training process. For other purpose, ISTC-CNR of Padua had previously trained a SONIC Italian acoustic model (Cosi, 2008). We have insert this model in SONIC first alignment process to provide a good segmentation to start from. After this, acoustic model estimation was computed.

At the end of further eight loops of phonetic alignment and acoustic model re estimation, where the AM output of a loop was used to produce the following loop segmentation, the final AM is considered well trained.

SPHINX does not need a precise segmentation to start the training process. Usually, even an uniform phonetic segmentation is sufficient. So, no previously trained AM were used in SPHINX training.

After uniform segmentation and first AM estimation, four loops of re-alignment and contest-independent (CI) AM computation were done. The last CI trained model was used to create the segmentation to train contest dependent (CD) AMs. First an untied AM was computed, then four loops of CD state-tied segmentation–estimation were done. The tying process aims to eliminate the useless states and consequentially reduce computational effort. We choose to limit them to 1000.

A consequence of not using a previously trained AM is that the SPHINX phone set is restricted to those which are in pronunciation lexicon, i.e. only 20.

4.3 Results

In TABLE II the results for the clean, noisy, and clean + noisy experiments for CSLU Toolkit are summarized. It shows, as expected, quite good performance with clean digit sequences, and also quite promising results in the noisy and clean + noisy case. Finally, with the best obtained network a final test will be executed on the EVALITA clean and noisy digits test-sets.

TABLE II
CSLU TOOLKIT ASR RESULTS IN TERMS OF WORD- ACCURACY.

	clean	**noisy**	**clean + noisy**
IA	99,82	90,15	93,86
FA	99,75	90,93	94,12
FB1	**99,94**	**92,11**	94,45
FB2	99,82	91,75	**94,76**
FB3	99,75	91,49	94,51

In the table above, we used the following conventions:

- IA means Initial Alignment,
- FA means Force Alignment
- FBn is the n-th loop of Forward Backward process.

In the following Tables, we used the conventions:

- Full_AM: AM model trained on both clean and noisy audio files.
- Clean_AM: AM model trained on only clean audio files.
- Noisy_AM: AM model trained on only noisy audio files.

- LM: configuration in which Language Model is not 0.
- NO_LM: configuration in which Language Model weight is 0.

In TABLE III the results for the clean, noisy, and clean + noisy experiments for SONIC are summarized.

TABLE III
SONIC ASR RESULTS IN TERMS OF WORD-ACCURACY.

	clean	**noisy**	**clean + noisy**
Full_AM + LM	99,00	92,00	95,19
Full_AM + NO LM	99,70	94,20	96,71
Clean_AM + LM	99,40	89,20	93,85
Clean_AM + NO LM	**99,80**	89,90	94,42
Noisy_AM + LM	99,30	93,10	95,93
Noisy_AM + NO LM	99,70	**94,80**	**97,04**

In TABLE IV the results for the clean, noisy, and clean + noisy experiments for SPHINX are summarized.

TABLE IV
SPHINX ASR RESULTS IN TERMS OF WORD- ACCURACY.

	clean	**noisy**	**clean + noisy**
Full_AM + LM	**99,40**	91,80	95,27
Full_AM + NO LM	**99,40**	**93,30**	**96,08**
Clean_AM + LM	**99,40**	77,40	87,44
Clean_AM + NO LM	**99,40**	78,70	88,15
Noisy_AM + LM	99,10	91,70	95,08
Noisy_AM + NO LM	98,80	92,60	95,43

5. CONCLUSIONS

EVALITA provides a shared framework for the evaluation of different systems and approaches on separate speech processing tasks for the Italian language. Three of the most used open source ASR tools were considered in this work, i.e. CSLU Toolkit, SONIC, and SPHINX, simply because promising results were obtained in the past on similar digit recognition tasks. The clean, noisy and clean+noisy (full) EVALITA digit task was chosen mainly because it is a simple and natural starting point to test a common evaluation methodology within the EVALITA framework.

From various experiment results few simple considerations can be made:

- beyond the fact that one of the main difficulties was that of finding similarity among the three ASR systems in order to choose comparable configurations because every system has its own completely distinct architecture and consequently the configuration parameters are hardly comparable, an homogeneous and unique test framework for comparing different Italian ASR systems was quite possible and effective if the results produced by the best WA-score configuration were compared foe each of the systems;

- in term of better recognition performance the configuration in which Language Model is not used at all (LM = 0) is the best because this should really simulate complete independence between connected digits while using words as basic recognition unit;
- CSLU Toolkit is incredibly good in recognizing clean digit sequences (99.94%), even if we are expecting lower performance with the real test data;
- SONIC is the best system in all noisy and full situations and we believe this is mainly due to the adoption of the PMVDR features;
- SPHINX is quite more sensible to AM specialization than other systems and clean models can not recognize noisy speech;

Finally we should conclude that the EVALITA campaign was quite effective in forcing various Italian research groups to focus on similar recognition tasks working on common data thus comparing and improving various different recognition methodologies and strategies, and we hope more complex task and data will be exploited in the future.

6. ACKNOWLEDGMENT

Our great thanks go the whole CSLU group at OGI Portland Oregon, and in particular to John Paul Hosom and Johan Shalkwyk for their tremendous patience and help in making us "*playing*" with the CSLU Speech Toolkit, and also to the whole CSLR group at Colorado University and in particular to Bryan Pellom (now at Rosetta Stone) for his invaluable help and useful suggestions in developing SONIC experiments, and for their true friendship.

7. REFERENCES

Bourlard, H., (1995), Towards Increasing Speech Recognition Error Rates, in *Proceedings of EUROSPEECH 95*, Madrid, Spain, 1995, Vol. 2, pp. 883-894.

Cole, R.A., (1999), Tools for research and education in speech science, in *Proceedings of ICPhS99*, San Francisco, CA, Aug 1999, Vol. ", pp. 1277-1280.

Cosi, P., and Hosom, J.P. (2000), High Performance "General Purpose" Phonetic Recognition for Italian, in *Proceedings of ICSLP2000*, International Conference on Spoken Language Processing, Beijing, Cina, 16-20 October, 2000, Vol. II, pp. 527-530.s

Cosi, P (2008), Recent Advances in Sonic Italian Children's Speech Recognition for Interactive Literacy Tutors, in *Proceedings of 1st Workshop on Child, Computer and Interaction* (ICMI'08, 10TH International Conference on Multimodal Interfaces, post-conference workshop), Chania, Crete, Greece, October 23, 2008, CD-ROM.

Lee, K.F., Hon, H.W., and Reddy, R., (1990), An overview of the SPHINX speech recognition system, in *IEEE Trans. on Acoustics, Speech, and Signal Processing*, Vol. 38, No. 1. (1990), pp. 35-45.

Murthi, M.N., and Rao, B.D., (1999), MVDR Based All-Pole Models for Spectral Coding of Speech, in *Proceedings of ICASSP99*, Phoenix, 1999.

Pellom, B., and Hacioglu, K., (2004), *SONIC: Technical Report TR-CSLR-2001-01*, Center for Spoken Language Research, University of Colorado, Boulder, revised 2004.

Sutton, S., Cole, R.A., de Villiers, J., Schalkwyk, J., Vermeulen, P., Macon, M., Yan, Y., Kaiser, E., Rundle, B., Shobaki, K., Hosom, J.P., Kain, A., Wouters, J., Massaro, D., and Cohen, M., (1998), Universal Speech Tools: the CSLU Toolkit, in *Proceedings of ICSLP 98*, Sydney, Australia, November 1998, Vol. 7, pp. 3221-3224.

Umit H. Yapanel, John H.L.Hansen, "A new perspective on Feature Extraction for Robust In-vehicle Speech Recognition" Proceedings of Eurospeech'03, Geneva, Sept. 2003.

Yan, Y., Fanty, M. and Cole, R., (1997), Speech Recognition Using Neural Networks with Forward-Backward Probability Generated Targets, in *Proceedings of ICASSP97*, April 1997, Vol. 4, pp. 3241-3244.