ProtoLOGOS, System for Romanian Language Automatic Speech Recognition and Understanding (ASRU)

Diana Militaru, Inge Gavat, Octavian Dumitru, Tiberiu Zaharia, Svetlana Segarceanu
Faculty of Electronics, Telecommunications and Information Technology
University POLITEHNICA
Bucharest, Romania
diana.militaru@gmail.com

The ProtoLOGOS Romanian language automatic speech recognition and understanding system is based at the level of acoustical modeling on the statistical framework of hidden Markov models and at the linguistic level on a finite state grammar or a bigram language model. The system enhanced the speech signal modeling by using for the first time the PLP (Perceptual Linear Prediction) features in a Romanian language recognition system. The main performance indicator for our system is the phrase recognition rate. This paper presents the main results obtained using the two language models already mentioned and the METEO database, constituted by broadcasted forecast news records.

Keywords – speech recognition; speech understanding; Romanian language; HMM; CDHMM; SCHMM; language model; monophones; triphones

I. INTRODUCTION

Language modeling is the attempt to encode linguistic knowledge in a way which is useful for computer systems in dealing with human language. It plays a critical role in speech recognition because it defining the structure of the language representing lexical, syntactic, and semantic information. There are mainly two types of language models: the grammar which restricts sentences in a mandatory format, and statistical language models which illustrate the probabilistic relationships among a sequence of words. In this paper the grammar was represented by a finite state grammar and the statistical language models by a bigram model.

The work focuses on the speech recognition of Romanian forecast radio recordings. The speech is processed with the ProtoLOGOS automatic speech recognition and understanding system [10]. Continuous and semi-continuous HMMs (for monophones and triphones) are used for the acoustic modeling of speech. The rates used to notice the speech recognition improvements are the word recognition rate (WRR), the accuracy and especially the phrase recognition rate (PRR). The final results show the importance of language models for continuous speech understanding.

II. THE PROTOLOGOS AUTOMATIC SPEECH REGOGNITION AND UNDERSTANDING SYSTEM

The ProtoLOGOS automatic speech recognition and understanding system [10] is developed in C#. It is based on HTK (HMM Toolkit, [18]) for the HMMs functions. The elected phonetic unit is the phone in two versions: without context (monophone) and with left-right context (triphone).

The ProtoLOGOS system main modules are: training, recognition and knowledge resources (Figure 1). In the training module, first the raw speech waveforms are parameterized into sequences of feature vectors using one of the listed methods: linear prediction coefficients (LPC), linear prediction cepstra, linear prediction reflection coefficients, mel-frequency cepstral coefficients (MFCC) or perceptual linear prediction coefficients (PLP). Afterwards the spectral features are used to build the HMM acoustic models in two typologies: with semi-continuous and continuous probability densities (with single and multiple mixtures), both designed for monophones and triphones (intra-word and cross-words). The ProtoLOGOS system proposes three HMMs initialization manners: global initialization training (with zero means and unit variances, without the manual speech signal segmentation), global initialization re-training and individual initialization re-training (the re-initialization methods replace the speech signal manual segmentation). Three data normalization methods (cepstral mean, grand variance or speaker's vocal tract length) are proposed to ensure robustness system.

The knowledge resources module has three parts: the phonetic dictionary (all the phonetic transcriptions of every word in the training dictionary), the HMM acoustic models (from speech signal training) and the language models (finite state grammar and bigram model).

The recognition module of the ProtoLOGOS system classifies and decodes the speech features using the knowledge resources to offer the estimated speech sequence. It offers a number of tools to visualize the signal waveforms and their segmentation, interpret the results, display the results in a graphical mode, optimize the training and testing process using some user modifiable parameters, or build the phonetic dictionary and language models.

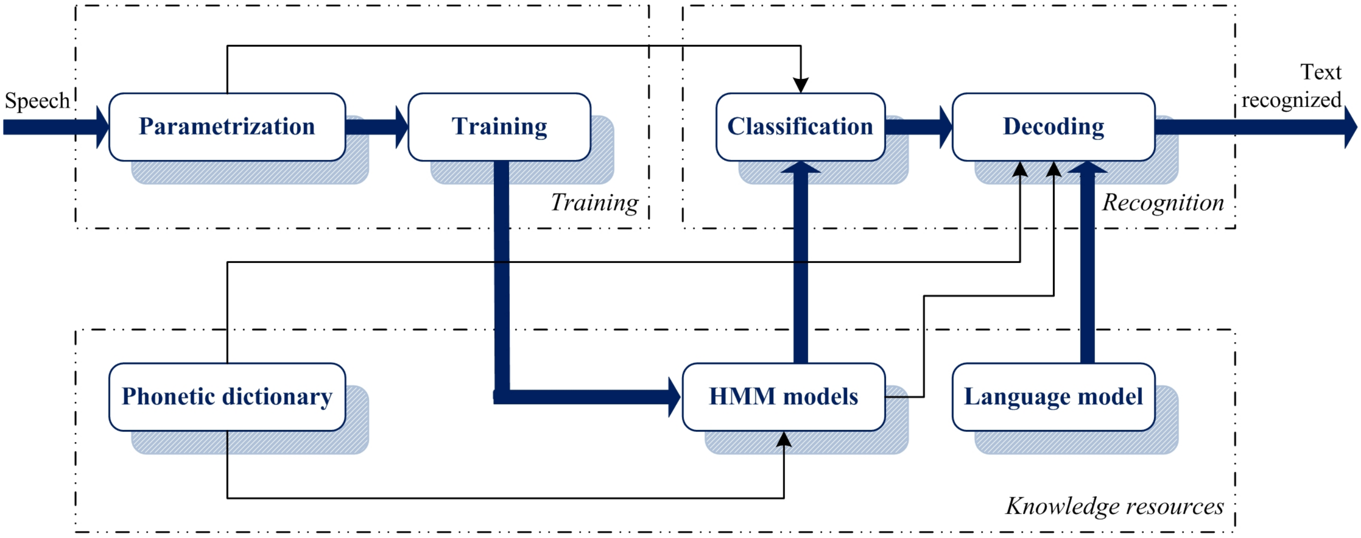

Figure 1. The schema of the ProtoLOGOS automatic speech recognition and understanding system

In this paper, only the main results of running the ProtoLOGOS system using the language models and the PLP parameterization for continuous and semi-continuous HMMs and for all the contexts are presented. For this, we have focused especially on improving the phrase recognition rate.

III. Acoustic Models

For limited vocabulary, word models are widely used, since they are accurate and trainable. In the situation of a specific and limited task they become valid if enough training data is available, but typically it is not possible to generalize.

Monophones constitute the foundation of any training method. But in real speech the words are not simple strings of independent phonemes, because each phoneme is affected by the immediately neighboring phonemes through co-articulation. Therefore context was added to monophones leading for example to triphones – monophones with left and right context, which became the state of the art in automatic speech recognition and understanding for large vocabularies [17].

Based on SAMPA (Speech Assessment Methods Phonetic Alphabet, [20]), Romanian language uses 34 phonemes and for which a model should be trained. For triphones, the training process could easily get out of control due to the large number of models that should be trained – around 40,000. An efficient solution to this problem is to tie the acoustically similar states of different triphone models [17].

The ProtoLOGOS system uses monophoes, intra-word and cross-word triphones. The triphones' similar states are tied for better control of the training process.

The phonetic units are modeled by Hidden Markov Models (HMM), basic entities in the statistical framework [1], [2].

A HMM model is a finite state automata, for which transitions from one state to another are made at equally spaced time intervals with the emission of a . An observation is emitted at each transition, thus two processes are taking place: one that is transparent represented by the sequence of observations, and a hidden one, represented by the sequence of states.

In speech recognition, the left-right model (or the Bakis model) is considered to be the best choice. For each symbol, such a model is constructed; a model corresponding to a word is obtained by connecting corresponding HMMs in sequence [7]. The hidden Markov model incorporates the knowledge about the particular feature constellation corresponding to each of the distinct phonetic units to be recognized.

Based on HMMs the statistical strategies have many advantages, among them being recalled: rich mathematical framework, powerful training and decoding methods, good sequence handling capabilities, flexible topology for statistical phonology and syntax. Due to these advantages, hidden Markov models are today the widest used in practice to implement speech recognition and understanding systems. The disadvantages lie in the poor discrimination between the models and in the unrealistic assumptions that must be made to construct the HMMs theory, namely the independence of the successive feature frames (input vectors) and the first order Markov process [5].

In the learning process, the parameters of the model are estimated to fit the data. The training represents the estimation of every HMMs means and variances. For each phonetic unit a HMM model can be developed deriving efficiently a local maximum likelihood with the Baum-Welch algorithm, also known as the forward-backward algorithm, a particular case of the Expectation-maximization algorithm [14], [15].

The evaluation of the optimal observation sequence's probability generated by the HMM model requires finding a maximum over all possible state sequences, and it can similarly be solved efficiently by the Viterbi algorithm which acts fast and decodes the uttered sequence easier [14], [15].

HMMs are principally concerned with continuous density models in which each observation probability distribution is represented by a mixture of Gaussian densities. On the other hand, discrete HMMs can be used to model speech by using the vector quantization to map continuous density vectors into discrete symbols. A vector quantization depends on a so-called codebook which defines a set of partitions of the vector space. Each partition is represented by the mean value of the speech vectors belonging to that partition and optionally a variance representing the spread. In comparison with continuous HMMs, the discrete HMMs have the advantage of low run-time computation. However, vector quantization reduces accuracy and this can lead to poor performance.

As an intermediate between discrete and continuous, a fully tied-mixture system can be used [4]. These models are called semi-continuous. But to use them effectively in speech recognition systems a number of storage and computational optimizations must be made. The codebook size must be chosen for each data stream. When specific mixtures are tied a Gaussian mixture component is shared across all of the owners of the tie. The effect is to create a set of tied-mixture HMMs where the same set of mixture components is shared across all states of all models. After training the initial set of monophones the information relating to different sources (such as delta coefficients and energy) is split into distinct data streams and tied each individual stream. The data insufficiency problem is addressed by smoothing the distributions.

IV. Language Models

The language model is an important knowledge source, constraining the search for the word sequence that has produced the analyzed observations, in form of succession of phonemes. The language model includes syntactic and semantic constraints. If only syntactic constraints are expressed, the language model is reduced to a grammar. Following, some basic aspects concerning the language modeling and further experimental results obtained in ASRU experiments on a database of natural, spontaneous speech, constituted by broadcasted forecast news, are presented.

Language models intend to capture the interrelations between words in order to gain understanding of the phrases, their meaning. The language models best known are of two kinds: rule-based and statistical. Rule-based models conduct to a certain word sequence based on a set of rules. Statistical models determine the word sequence based on a statistical analysis of a large amount of data [7], [8].

A. Parsing Techniques

Parsing algorithms are applied to search the desired word sequence in an utterance, based on rules or based on statistics.

The rule based parsing technique relies on specialized knowledge from linguists to determine the set of rules (Figure 2). The statistical parser relies on a large training corpus to extract the language model (Figure 3). Efficiency of the statistical parsing can be enhanced by the so called bootstrap training [7], consisting in developing a model for a part of the training corpus and refining this model successively to cover the whole training material.

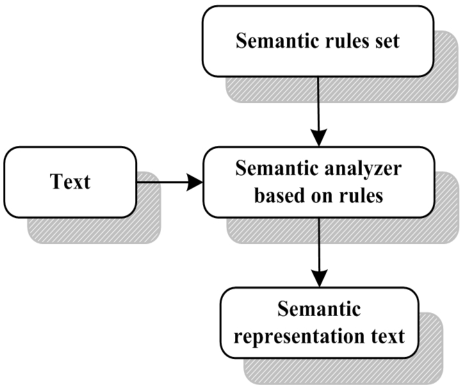
Semantic rules set
Text
Semantic analyzer based on rules
Semantic representation text

Figure 2. Rule based parser

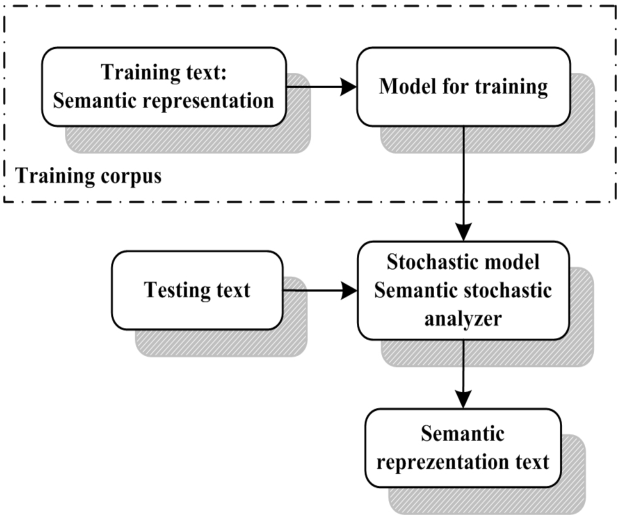
Training text: Semantic representation
Model for training
Training corpus
Testing text
Stochastic model Semantic stochastic analyzer
Semantic reprezentation text

Figure 3. Statistical parser

B. The Lattice Format

The recognition network consists of a set of nodes connected by arcs. Each node is a HMM model instance or a word-end. Each model node is a network consisting of states connected by arcs. Thus, once fully compiled, a recognition network ultimately consists of HMM states connected by transitions.

The decoder must find those paths through the network which have the highest log probability. These paths are found using a *Token Passing* algorithm [19], a token representing a partial path through the network extending from time 0 through to time *t*. As each token passes through the network it must maintain a history recording its route. The amount of the history’s details depends on the required recognition output. All data, at any level, can conveniently be represented using a lattice structure.

When all input observations have been processed, recognition is completed by generating a lattice (Figure 4). The lattice file can contain zero or more sub-lattices followed by a main lattice. Sub-lattices are used for defining sub-networks prior to their use in subsequent sub-lattices or the main lattice. The lattice contains optional information from language models, like alignment and score (log likelihood) information at the word and phone level (for calculation of the alignment and likelihood of an individual hypothesis).

```
N=512 L=2967
I=0  W=!NULL
I=1  W=!ENTER
I=2  W=</s>
I=3  W=<s>
I=4  W=A
I=5  W=ABUNDENTE
I=6  W=ACCENTUAT
…
J=0   S=1  E=0  l=-2.82
J=1   S=2  E=0  l=0.00
J=2   S=3  E=0  l=0.00
J=3   S=4  E=0  l=-1.68
J=4   S=5  E=0  l=-1.10
J=5   S=6  E=0  l=-1.25
J=6   S=7  E=0  l=-0.69
…
```

Figure 4. The bigram model lattice example for METEO database (N = num nodes, L = num arcs, I = node-number, W = word, J = arc-number, S = start-node, E = end-node, l = the log transition probability attached to an arc)

C. The Word Loop Grammar (WLG)

The word loop grammar are simple word loops which put all words of the vocabulary in a loop and therefore the decoder allows any word to follow any other word, without any kind of rules. It is used to point out the testing results using language models and their role in speech understanding.

In Figure 5 is the METEO database word network for word loop grammar.

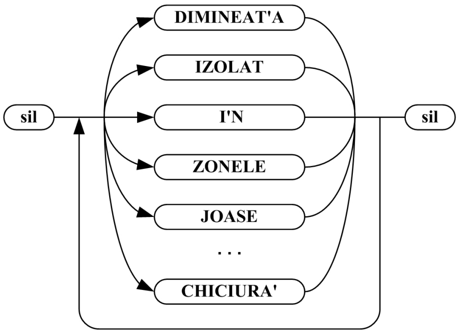

Figure 5. Word network for word loop grammar

This grammar can be used especially for isolated word recognition or reduced vocabulary applications, otherwise the processing time could be very long due to the large search domain.

A. The Finite State Grammar (FSG)

To create a finite state grammar it is necessary a grammar notation. This notation is based on the Extended Backus-Naur Form (EBNF) used in compiler specification and it is compatible with the grammar specification language. It generates word level lattice files using the rewrite rules contained in a syntax description file.

This set of rules is based on words' semantic role and it can not be used for other databases. The finite state grammar building process is complex and laborious.

Expressions used to generate the word level lattice files are constructed from sequences of words and the meta-characters [18]:

| denotes alternatives
[ ] encloses options
{ } denotes zero or more repetitions
< > denotes one or more repetitions
<< >> denotes context-sensitive loop

For these, firstly the variables are identified by a leading $ character. They stand for sub-networks and must be defined before they appear in a rule by adding a silence model before and after all existing sequences. Afterwards each sequence is composed of a sequence of factors where a factor is either a node name, a variable representing some sub-network or an expression contained within various sorts of brackets. Finally, the complete network is defined by a list of sub-network definitions.

In the figure 6 the finite state grammar in the case of the METEO database is illustrated.

```
...
$vbMeteo = VA {MAI} (NINGE | FI | DEVENI | PREZENTA |
     CONTINUA | PLOUA | CA'DEA | RA'MI'NE | I'NNORA | ATINGE);
$nintensificari = {CU | (CU UNELE)} INTENSIFICA'RI;
$inZonele = I'N $zonele {(<$tipZone | $deTipZone> {S'I <$tipZone |
     $deTipZone>} {DIN $punctCardinal S'I $punctCardinal}) | (CU
     CEAT'A')};
$punctuluiCardinal = ESTICE | VESTICE | SUD-ESTICE | NORDICE |
     NORD-VESTICE | CENTRALE;
...
( sil (
…
$nvantul $vbMeteo $nintensificari $inZonele $punctuluiCardinal |
…
) sil )
```

Figure 6. A sequence of finite state grammar for METEO database ([10])

```
...
I'N[208] I'N
  Pred: [U:1] ****[178]
  Succ: [U:2] I'NCA'LZIRE[196] RA'CIRE[190] US'OARA'[202]
US'OARA'[202] US'OARA'
  Pred: [U:3] I'N[208] US'OARA'[202]
  Succ: [U:2] I'NCA'LZIRE[196] RA'CIRE[190] US'OARA'[202]
I'NCA'LZIRE[196] I'NCA'LZIRE
  Pred: [U:3] I'N[208] US'OARA'[202]
  Succ: [U:3] ****[172] US'OARA'[184]
RA'CIRE[190] RA'CIRE
  Pred: [U:3] I'N[208] US'OARA'[202]
  Succ: [U:3] ****[172] US'OARA'[184]
US'OARA'[184] US'OARA'
  Pred: [U:2] US'OARA'[184] I'NCA'LZIRE[196] RA'CIRE[190]
  Succ: [U:3] ****[172] US'OARA'[184]
****[178] <Blank>
  Pred: [U:1] ****[226]
  Succ: [U:1] I'N[208]
...
```

Figure 7. Some examples of finite state grammar words links (Pred = predecessor, Succ = successor) for word group ([10]) *I'N {US'OARA'} (I'NCA'LZIRE | RA'CIRE) {US'OARA'}*

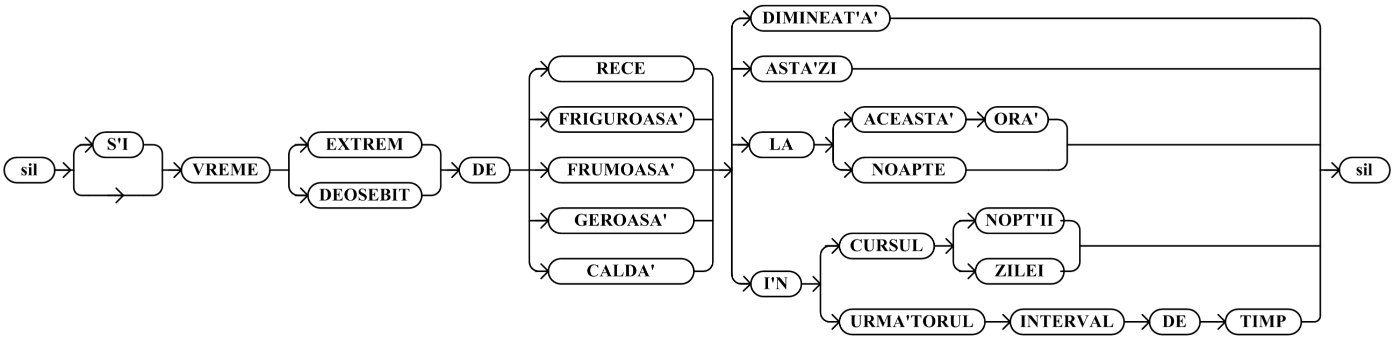

Figure 8. Example of word network using finite state grammar for some phrases ([10]), one of them being
Şi vreme deosebit de rece în următorul interval de timp

In Figure 7 some words' successive links are exemplified for one word group. Here, every word's predecessor, successor and their instantiation number in the training test are noted.

One example of a word network is in the Figure 8. Here the word network is built using similar phrases.

B. The Bigram Model

The language model is represented by the probability *P(W)* which can be written in the form:

$$P(W) = \prod_{i=1}^{n} P(w_i | w_1, w_2, ..., w_{i-1}) \qquad (1)$$

where $P(w_i | w_1, w_2, ..., w_{i-1})$ is the probability that the word w_i follows after the word sequence $w_1, w_2, ...w_{i-1}$.

The choice of w_i depends on the whole input history. For a vocabulary having of size v there are v^{i-1} different histories; it is a huge number, making practically impossible to estimate the probabilities even for reasonable values of *i*.

To find a solution, shorter histories are considered and the one used in this paper is the bigram model based of a history formed with the predecessor word, ($P(w_i | w_{i-1})$), [7], [18].

The first step in bigram model building is data preparation, dividing the phrases and eliminating the punctuation marks. The bigram model uses the phonetic dictionary to create a bigram table. A small part of disposable probability is deduced from the most frequent bigram probability and assigned them between the least frequent bigrams. This process is called discount. If the bigram model is smaller than the desired threshold, then it is restricted to unigram model probability scaled by a back-off weight.

A measure of language model performance based on average probability can be developed within the information theory field. A speaker emitting language can be considered to be a discrete information source generating a sequence of words w_1, w_2, w_m from a vocabulary set, *W*. The probability of a symbol w_i is dependent upon the previous symbols $w_1, ..., w_{i-1}$. The information source's inherent per-word entropy *H* represents the amount of non-redundant information provided by each new word on average. In this case *H* can be approximated with:

$$H(W) = -\frac{1}{N_W} \log_2 P(W) \qquad (2)$$

where N_w is the text length measured in words. This estimate provides the basis of a metric suitable for assessing the performance of a language model.

Considering a language model as an information source, it follows that a language model which took advantage of all the language's possible features to predict words would also achieve the word entropy of *H*. A measure called perplexity, PP, defined in [18], similar to the entropy, can be used to assess the actual performance of the language model.

$$PP(W) = 2^{H(W)} \qquad (3)$$

Perplexity can be considered to be a measure of how many different equally most probable words can follow any given word on average. Lower perplexities represent better language models. Perplexity is only correlated with performance in a speech recognition system, and it has no ability to note the relevance of acoustically similar or dissimilar words of any particular system; but it will reveal how well a given piece of text is modeled by a language model.

```
ngram 1=537
ngram 2=2014

\1-grams:
…
-3.6256         IARNA'    -0.2710
-3.8017         IAS'I     -0.2734
-3.6256         IERI      -0.2680
…
\2-grams:
…
-0.3010         AFECTEAZA' SUDUL
-0.3010         AJUNGE PI'NA'
-0.7782         AJUNGE S'I
…
```

Figure 9. The bigram model for METEO database ([10])

In Figure 9 a METEO database n-gram model with 537 unigram models and 2014 bigram models is presented. Each unigram definition starts with a probability value stored as $\log_{10}$ followed by the word describing the unigram and a back-off weight which is also stored as $\log_{10}$. For the bigram there are only the probability's value (stored as $\log_{10}$) followed by the words describing the bigram

When a bigram count falls below the selected threshold, the bigram is backed-off to the unigram probability suitably scaled by a back-off weight in order to ensure that all bigram probabilities for a given history sum to one.

V. THE DATABASE

The further displayed results were obtained on the METEO database, constituted by recordings of broadcasted forecast news. In the Table 1 the main database characteristics are presented. The speech is natural. All speech files were sampled at 11025 Hz with 16 bits per sample in mono waveform.

TABLE I. THE MAIN METEO DATABASE CHARACTERISTICS ([10])

Dictionary dimension	**Training set**			
	Number of phrases	***Total number of words***	***Number of speakers***	
			Male	***Female***
535	715	12.666	14	14
	Training set			
	Number of phrases	***Total number of words***	***Number of speakers***	
			Male	***Female***
	47	900	1	1

The training set and the test set are distinct, the training speakers and the test speakers are distinct. There are an average number of 18 words in a sentence and the average speed of the speech is between 2.66 and 3.96 words per second. The database has a total number of 762 phrases and 30 speakers and a number between 4 to 33 phrases per speaker.

```
BIGRAM MODEL
-> perplexity: 12.5957,
-> utterances: 715
-> words predicted: 12666
-> tokens number: 13381

Statistics:
-> words predicted: 12666
-> bigram models: 88.7%
-> backed off: 5.6%
-> unigram models: 5.6%
```

Figure 10. The METEO database characteristics for bigram model ([10])

In the Figure 10 the METEO database characteristics for bigram model are shown. The first part represents the main bigram model data, like perplexity (12.5957), number of utterances (715), number of predicted words (12666) or number of tokens (13381).

The second part represents the statistics using the back-off method. Here 88.7% were found as explicit bigrams in the model, 5.6% were computed by backing off to the respective unigrams and 5.6% were simply computed as unigrams by shortening the word context.

VI. EXPERIMENTAL RESULTS

As speech data, the utterances of the data base were processed by phonetic transcription after the SAMPA standard [20], conducting to the phonetic dictionary of the system. Each word is decomposed in constituent monophones (34 for Romanian language) or triphones (34^3 for Romanian language) and for each monophone or triphone a model must be trained. Of course for monophones the number of necessary models is small, because there are sufficient training data, so that the models will be well trained in a short time. For triphones the situation is more complicated because there number is huge and the training data can become insufficient [16]. Therefore tying procedure must be adopted, combining in a model similar triphones using decision trees. This procedure is based on asking questions about the left and right contexts of each triphone. The decision tree attempts to find those contexts which make the largest difference to the acoustics and which should therefore distinguish clusters [16].

To achieve a balance between speed and accuracy the Viterbi search is replaced with a straightforward Viterbi beam search such that any model whose maximum log probability token falls more than the main beam selected below the maximum for all models is deactivated [19].

The digitized data are further analyzed in order to extract characteristic cues, called features. By short term analysis a set of features is obtained for each speech frame, extracted by a windowing process. The frame duration is chosen by making a compromise between a long interval (20-40 ms) imposed in order to detect the periodic parts of speech and the short time during which the speech can be considered a stationary random process (around 20 ms.). The experimental results further presented are obtained with a Hamming window, with duration 25 ms and the overlapping factor of the windows ½.

The features that can be obtained in this manner to characterize speech segments are of two kinds: static features (perceptive linear prediction coefficients – PLP, [6], [11] obtained for each window, and dynamic features, calculated over a number of windows and representing derivatives of first (delta, D) and second (acceleration, A) degree. Optional normalized energy (E) can be added.

The training procedure uses global initialization, meaning that the training starts with all initial models having zero mean and unitary variance [10], [3]. The following types of continuous density models are trained and tested in the experiments:

- Simple mixture monophones (SMM) (Table 2)
- Multiple mixture monophones (MMM) (Table 2)
- Simple mixture intra-word triphones (SMIWT) (Table 3)
- Multiple mixture intra-word triphones (MMIWT) (Table 3)
- Simple mixture cross-word triphones (SMCWT) (Table 4)
- Multiple mixture cross-word triphones (MMCWT) (Table 4)

The following types of semi-continuous density models used in the experiments are based on:

- Monophones (Table 2)
- Triphones intra-word (Table 3).

TABLE II. COMPARATIVE RECOGNITION PERFORMANCE FOR WLG, FSG AND BIGRAM MODEL IN THE CASE OF MONOPHONES, FOR CDHMM AND SCHMM ([10]).

Number of mixtures/ language models		**PLP + E + D + A**			**PLP + D + A**		
		WRR	***Accuracy***	***PRR***	***WRR***	***Accuracy***	***PRR***
CDHMM							
SMM	WLG	67.28	65.76	2.08	65.98	64.79	4.17
	FSG	95.89	95.11	55.32	96.33	95.67	53.19
	Bigram	97.40	96.97	72.92	98.70	98.37	81.25
10 MMM	WLG	87.64	86.44	10.42	80.15	80.04	10.42
	FSG	97.22	96.44	65.96	96.89	96.00	59.57
	Bigram	98.70	98.59	83.33	98.37	98.16	83.33
SCHMM							
	WLG	83.42	80.82	6.25	78.01	77.03	4.17
	FSG	97.00	96.22	61.70	96.89	96.22	59.57
	Bigram	98.81	98.59	83.33	99.13	98.92	85.42

TABLE III. COMPARATIVE RECOGNITION PERFORMANCE FOR WLG, FSG AND BIGRAM MODEL IN THE CASE OF INTRA-WORD TRIPHONES, FOR CDHMM AND SCHMM [10].

Number of mixtures/ language models		**PLP + E + D + A**			**PLP + D + A**		
		WRR	***Accuracy***	***PRR***	***WRR***	***Accuracy***	***PRR***
CDHMM							
SMIWT	WLG	87.87	81.15	12.50	89.38	85.70	14.58
	FSG	96.67	95.56	65.96	97.33	96.11	68.09
	Bigram	99.35	99.24	89.58	99.24	99.02	85.42
6 MMIWT	WLG	87.87	81.15	12.50	89.11	87.11	17.02
	FSG	98.05	97.14	71.74	97.00	96.33	70.21
	Bigram	99.35	99.13	85.42	99.35	99.02	81.25
SCHMM							
	WLG	92.74	88.52	22.92	90.15	88.42	16.67
	FSG	97.89	96.78	65.96	98.33	97.11	68.09
	Bigram	98.81	98.37	77.08	98.22	98.00	76.60

TABLE IV. COMPARATIVE RECOGNITION PERFORMANCE FOR WLG, FSG AND BIGRAM MODEL IN THE CASE OF CROSS-WORD TRIPHONES, FOR CDHMM [10].

Number of mixtures/ language models		**PLP + E + D + A**			**PLP + D + A**		
		WRR	***Accuracy***	***PRR***	***WRR***	***Accuracy***	***PRR***
SMCWT	WLG	81.15	79.20	2.08	85.98	81.74	2.64
	FSG	96.11	95.56	59.57	97.67	96.56	61.70
	Bigram	99.35	99.24	87.50	99.13	98.92	85.42
6 MMCWT	WLG	82.20	77.69	6.25	81.78	78.89	4.26
	FSG	97.71	97.37	69.57	96.67	96.22	68.09
	Bigram	99.44	99.44	89.36	99.00	99.00	87.23

TABLE V. TRAINING AND AVERAGE TESTING DURATION FOR DIFFERENT LANGUAGE MODELING AND ACOUSTICAL MODELING TECHNIQUES ([10]).

HMMs		**Training duration (sec.)**	**Average testing duration/word (sec.)**		
			WLG	***FSG***	***Bigram***
CDHMM	SMM	162	3.220	1.18	0.102
	10 MMM	672	3.930	1.43	0.130
	SMIWT	259	2.979	0.037	0.072
	6 MMIWT	361	1.250	0.069	0.097
	SMCWT	261	3.289	0.078	0.122
	6 MMCWT	652	7.390	0.143	0.205
SCHMM	Monophones	3.250	4.004	0.902	0.108
	Triphones	8.040	4.031	1.552	1.002

The language models applied in the presented research are:

- a simple word loop grammar, (WLG), permitting ever word sequence, without any restrictions
- a restrictive finite state grammar (FSG), allowing only certain word sequences
- a bigram statistical model, extracting valid word sequences based on the bigram probabilities

For the monophone SCHMMs a 512 codebook size was experimentally chosen. For the triphone SCHMMs the codebook size was different for each stream (from 1024 to 256) and the mixture weight 2 (for PLP+D+A) and 4 (for PLP+E+D+A). The information related to static, delta and acceleration coefficients (as well as the energy for PLP+E+D+A) was split into 3 distinct data streams for PLP+D+A and into 4 distinct data streams for PLP+E+D+A. The threshold was set to 20.

The experiments had as objective to determine the influence of the language model on the ASRU performance, expressed in WRR, accuracy and PRR.

For monophones, the best results are obtained for multiple mixture CDHMMs in the case of FSG and WLG (PLP+E+D+A) and for SCHMMs in the case of bigram model with 85.42% (PLP+D+A). FSG is better than WLG with around 54.5% and the bigram model with 70.86% - 81.25%.

For triphones, the best results are for SCHMMs in the case of WLG with 22.92%, for multiple mixture CDHMMs in the case of FSG with 71.74% and for single mixture CDHMMs in the case of bigram with 89.58%, all of this for PLP+E+D+A. The improvements of language models are 43.04% - 59.24% for FSG and with 64.23% - 77.08% for bigram model.

The cross-word triphones bring significant improvements if they are used with language models, between 57.49% - 63,83% for FSG and around 84% for bigram model. The multiple mixtures have the best result with 89.36%.

Investigation of training duration and average testing duration/word were also done and the obtained results are displayed in Table 5. It is necessary to mention the training and the testing process using the ProtoLOGOS system ([10]) ran on a Pentium IV platform with 1 GHz processor and 1 GB DDRAM.

The longest training time is for SCHMMs and the shortest is for CDHMMs using monophones. For monophone CDHMMs and SCHMMs, the shortest testing duration is for bigram model, but for triphone CDHMMs it is for FSG. The longest testing time for WLG is for cross-word triphone CDHMMs. In the case of language models, the larger testing time is for triphone SCHMMs and CDHMMs for monophones.

From the HMM modeling point a view it has been observed that monophone SCHMMs are situated between single and multiple mixtures CDHMMs for WLG and FSG. On the other hand, the bigram model SCHMMs yields better rates. For intra-word triphones, if the results of SCHMMs were higher than single and multiple mixtures CDHMMs for PLP+E+D+A in case of WLG, for bigram model the situation is inverted. In this case, because the database is small isn’t necessary the use of multiple mixture for CDHMMs and the best results is obtained with SCHMMs for WLG. A reduced number of cross-word triphones requires an increase of Gaussian mixtures. The fluent and continuous speech is reflected in the results of cross-word triphones and bigram model which are the best.

From the parameterization point a view, the influence of energy normalization is visible for WLG and FSG in the case of monophone models and for bigram model in the case of triphone models. In general, the results show that it is better to use energy normalization for real noise conditions.

Given the WLG, the influence of language models over the phrase recognition rate is overwhelming, the results being with 44-63% better for FSG and with 60-85% better for bigram model.

Despite of the large processing time, the SCHMMs, with a correct selected configuration, can offer competitive results.

VII. Conclusions

For these experiments a natural spontaneous spoken language database was used, namely METEO, recorded in a laboratory, under not severe noise and the ProtoLOGOS speech recognition and understanding system. The training and testing speakers pronounced a number of different phrases. The training speakers and phrases were different from the test speakers and phrases.

To make evident the improvements of language models in comparison with word loop grammar, the phrase recognition rate was especially analyzed.

Comparing the results obtained in [2] using also HMMs in speech recognition for the Romanian language. Here, related to a different database, recorded in a laboratory and with the same phrases for each speaker, the results are comparable for the word recognition rate. But this paper brings a new perspective about the possible phrase recognition improvements using the language models, especially bigram model, as first step in the implementation of statistical language models.

To evaluate the performance of ProtoLOGOS system we compared the results with a similar experiment for Croatian language described and discussed in [9] where HMMs and bigram models are also applied. There are some differences: in the database (e.g. 5 speakers and 252 minutes of speech material), in using another parameterization (mel-frequency cepstral coefficients, MFCC+E+D+A), but, finally, the phrase recognition results are also enhanced by the language model. The obtained PRR performance is comparable with that of the ProtoLOGOS system but there are situation when our system acts better.

Concerning the future, after the first step in developing the bigram language model for Romanian, we will try to develop more performing language models (based on trigrams and quadrigrams), in order to enhance the PRR capability of the Protologos system. But therefore, the development of professional databases for Romanian language becomes mandatory, because our own constructed databases are in a way local, containing especially speakers from better educational background and not sufficient large for such experiments.

References

[1] Gavăt, I., Zirra, M., Grigore, O., Vâlsan, Z., “Speech synthesis and recognition elements” (Elemente de sinteza şi recunoaşterea vorbirii), Ed. Printech, Bucharest, 2000.

[2] Gavăt, I., Dumitru, C.O., Costache, G., Militaru, D., Continuous Speech “Recognition based on statistical methods”, Proceedings of Speech Technology and Human-Computer-Dialog (SPED2003), Bucharest, pp. 115-126, 2003.

[3] Gavăt, I., Militaru, D., Dumitru, C., “Knowledge resources in automatic speech recognition and understanding for Romanian language”, Speech Recognition, I-Tech Education and Publishing KG, Vienna, Austria, pp. 241-260, 2008.

[4] Giurgiu, M., “On the use of semi-continuous hidden Markov models for speech recognition”, SPED 2003, Bucharest, Romania, 10-11 April, pp. 187-192, 2003.

[5] Goronzy, S. “Robust adaptation to non-native accents in automatic speech recognition”, Springer-Verlag, Berlin, 2002.

[6] Hermansky, H., “Perceptual linear predictive (PLP) analysis of speech”, Journal of the Acoustical Society of America, 87(4), pp. 1738-1752, 1990.

[7] Huang, X., Acero, A., Hon, H.W., “Spoken language processing – a guide to theory, algorithm and system development”, Prentice Hall, 2001.

[8] Juang, B.H., Furui, S., “Automatic recognition and understanding of spoken language – a first step toward natural human–machine communication”, Proceedings of the IEEE, Vol. 88, No. 8, pp. 1142-1165, 2000.

[9] Martinčić-Ipšić, S., Žibert, J., Ipšić, I., Mihelič, F., “Speech recognition of Slovenian and Croatian weather forecasts”, Proceedings B of the 5th International Multi-Conference of Information Society, IS'2002, Language Technologies, Ljubljana, Slovenia, pp. 106-110, 2002.

[10] Militaru, D., “Contribution of vocal signal processing methods for speech recognition in Romanian language” (Contribuții la metodele de prelucrare a semnalului vocal în vederea recunoaşterii vorbirii în limba română), PhD thesis, University Politehnica of Bucharest, 2008.

[11] Militaru, D., “Speech analysis with perceptual linear prediction”, The 33-th International Scientific Symposium of METRA, Bucureşti, 2002.

[12] Militaru, D., “Statistical models for speech recognition”, The 35-th International Scientific Symposium of METRA, Bucureşti, 2004.

[13] Oancea, E., Munteanu D., Burileanu, C., “Continuous speech recognition system improvements”, SPED 2005, Cluj-Napoca, Romania, pp. 81-91, 2005.

[14] Rabiner, L., “A tutorial on hidden Markov models and selected applications in speech recognition”, Proceedings of IEEE, Vol. 77, No. 2, pp. 257-286, February 1989.

[15] Rabiner, L., Juang, B.H., “Fundamentals of speech recognition”, Prentice Hall, Englewoods Cliffs, New Jersey, 1993.

[16] Young, S.J., “Tree based state tying for high accuracy acoustic modelling”, ARPA Workshop on Human Language Technology, Princeton, 1994.

[17] Young, S.J., “The general use of tying in phoneme-based HMM speech recognizers”, Proceedings ICASSP’92, Vol. 1, San Francisco, pp. 569-572, 1992.

[18] Young, S.J., Evermann, G., Kershaw, D., Moore, G., Odell, J., Ollason, D., Povey, D., Valtchev, V., Woodland, PH., “The hidden Markov modelling toolkit book”, version 3.4, Cambridge University Engineering Department, 2006.

[19] Young, S.J., Russell, N.H., Thornton, J.H.S., “Token passing: a simple conceptual model for connected speech recognition systems”, Technical Report, Cambridge University Engineering Department, 1989.

[20] ***, “SAMPA: The machine-readable phonetic alphabet by SAMPA (Speech Assessment Methods Phonetic Alphabet)”, http://www.phon.ucl.ac.uk/home/sampa/romanian.htm.