ROMANIAN JOURNAL OF INFORMATION SCIENCE AND TECHNOLOGY
Volume **18**, Number 1, 2015, 33–53

The SpeeD Grammar-based ASR System for the Romanian Language

Horia CUCU, Andi BUZO, Corneliu BURILEANU

Speech and Dialogue Research Laboratory,
University Politehnica of Bucharest, Romania
E-mail: {horia.cucu, andi.buzo, corneliu.burileanu}@upb.ro

Abstract. This paper describes the grammar-based automatic speech recognition system for the Romanian language developed by the Speech and Dialogue Research Group. The paper links to previous work for the issues related to large vocabulary speech recognition and focuses on the specific optimization work done for several closed-vocabulary, grammar-based speech recognition tasks. Among the specific problems approached, of particular interest is the informal pronunciation modelling of Romanian two-digit numbers. The paper proposes solutions for within-word and cross-word pronunciation modelling of numbers and reports significant relative improvements of the speech recognition word error rates.

1. Introduction

Automatic speech recognition (ASR) is still an unsolved topic for many languages, mainly because (i) there is a lack of acoustic and linguistic resources needed for development (it is the case of so-called under-resourced languages) and (ii) the scientific research community is not stimulated by any national or international evaluation campaigns (as opposed to languages such as English, French or Chinese). The Romanian language is affected by both the aforementioned problems. In this context, the development of speech and language resources for automatic speech recognition is a critical issue that must be addressed to push forward the research in this direction and create ASR systems comparable to those available for other languages. This is one of the main goals of the Speech and Dialogue (SpeeD) research group[1].

[1]Speech and Dialogue Research Group: http://speed.pub.ro

Large vocabulary continuous speech recognition (LVCSR) is a subclass of ASR which aims at transcribing speech possibly containing most, if not all the words in a specific language, or at least a broad sub-domain of it. Depending on the morphological richness of the language, large vocabulary might mean tens of thousands of words (English, French, etc.) or hundreds of thousands of words (Russian, German, Turkish, etc.). To the best of our knowledge, at the moment there are three LVCSR systems developed for the Romanian language. In 2011 we published the first LVCSR results for Romanian [1][2], in August 2012 Google launched their online speech recognizer[2] for Android and Chrome and in December 2012 THINKTech Research Center[3] also published a paper [3] on broadcast news recognition for Romanian.

The automatic speech recognition system developed by the Speech and Dialogue research group is continuously improved and upgraded. Recently, we reported significant improvements (between 30% and 35% relative word error rate reductions) obtained thanks to the extensions of the speech and text corpora and to the implementation of noise robust speech features [4]. This paper builds upon previous work and aims to present the grammar-based speech recognition module which was recently added to SpeeD's ASR system. The emphasis will be on the research and development issues identified and addressed in the process: (i) the design and implementation of rule grammars, (ii) the optimization of key ASR decoding parameters and (iii) informal pronunciation modelling for Romanian numbers. The proof-of-concept system presented in this paper is available online[4].

The rest of the paper is structured as follows. Section 2 is a brief introduction in automatic speech recognition and more specifically in ASR based on rule grammars. This section also describes the main ASR decoding parameters. Section 3 presents the SpeeD LVCSR system, as a starting point for the grammar-based ASR system, and continues by discussing the various rule grammars created in this study. Section 4 warns about important pronunciation variations of Romanian numbers and proposes solutions for both within-word and cross-word variations. Finally, section 5 is dedicated to the assessment of the proposed grammar-based ASR system on various tasks, while section 6 draws the final conclusions.

2. Rule-based Grammar ASR Systems

State-of-the-art ASR systems transcribe the speech into text using three models: an acoustic model, a language model and a pronunciation model. The acoustic model (AM) is used to estimate the probability that a speech signal was produced by uttering a specific sequence of words. The acoustic model does not use words as basic speech units because (i) every new ASR task comes its specific vocabulary (possibly comprising new words for which there is not any available training data) and (ii) the number of different words in a language is too large to model them all independently. Instead of using words as basic speech units, ASR systems model sub-word speech

[2]Google ASR System: `http://officialandroid.blogspot.ro/2012/08`
[3]THINKTech Research Center: `http://thinktech.hu`
[4]SpeeD grammar-based ASR system: `http://speed.pub.ro/speech-to-text`

units (such as phones) or even sub-phone speech units (such as senones). Typically, the acoustic model consists of a set of phone models which are linked, during the decoding process, to form word models and eventually a word sequence model. This model is eventually used to estimate the probability that the speech signal was produced by uttering that specific sequence of words. This generative approach has been proven to be very well served by the Hidden Markov Model (HMM) mathematical apparatus [5, 6, 7, 8].

The language model (LM) is used during decoding to estimate the probabilities of all word sequences in the search space. In general, the purpose of a language model is to estimate how likely is a sequence of words $W = w_1, w_2, ..., w_n$, to be a sentence in the source language. The probability for such a word sequence helps the acoustic decoding in the decision process. For example, in the Romanian language these two phrases: "ceapa roşie este sănătoasă" (red onion is healthy) and "ce apar oşti ied este sănătoasă" (what appear armies kid is healthy) are acoustically very similar, but the second one does not make any sense. The role of the language model is to assign a significantly larger probability to the first word sequence and consequently help the ASR system to decide in favour of the first phrase.

Finally, a pronunciation model is needed to link the acoustic model (which estimates phone acoustic probabilities) to the language model (which estimates word sequence probabilities). Usually, the pronunciation model is a phonetic dictionary that maps each word in the vocabulary to one or more sequences of phones, representing the way in which that word should be pronounced.

2.1. Statistical Language Models vs. Rule Grammars

LVCSR systems are typically required to transcribe speech possibly containing most, if not all the words in a specific language, or at least a broad sub-domain of it. Therefore, LVCSR systems usually use statistical language models (typically based on n-gram), which specify the frequency of occurrence for the words and sequences of up to n words in the language. Theoretically, any sequence of words in the vocabulary has a non-null probability. n-gram language models are created by estimating these occurrence probabilities over large corpora of text. The models are more accurate as the size of the text corpora is greater and as the text is more adapted to the sub-domain (*e.g.* medicine, sports, etc.).

Although statistical language models are state-of-the-art in LVCSR systems, there is a wide range of speech recognition applications for which they are sub-optimal. For example, in speech-based command and control applications, interactive voice response (IVR) systems, home automation systems, and others, the human user is usually restricted to a small-medium set of specific word sequences. In this context rule grammars are more appropriate. As opposed to statistical language models, which allow any word sequence with in-vocabulary words, rule grammars model explicitly all the allowed word sequences (along with their probabilities).

There are several standards for creating speech recognition rule grammars. Among them, the most used are Speech Recognition Grammar Specification (SRGS) and Java Speech Grammar Format (JSGF). SRGS provides two alternative ways of writing

grammars, one based on XML, and one using Augmented Backus-Naur Form (BNF) format. JSGF allows writing grammars only in an Augmented BNF format. Figure 1 presents an example of a JSGF dates grammar for the Romanian language.

Rule grammars are interpreted by the ASR system as finite state machines. An input state represents the entry of the speech recognition process, transitions between states represent output words and a final state represents the exit from the speech recognition process. Transitions can also have probabilities associated. Figure 2 shows the Romanian dates grammar represented as a finite state machine.

```
public <date> = <day> <month> [<year>];

<day> = <units> | <elevens> | douăzeci [şi <units>] | treizeci | treizeci şi unu;

<month> = ianuarie | februarie | martie | aprilie | mai | iunie | iulie | august |
septembrie | octombrie | noiembrie | decembrie;

<year> = (o mie nouă sute | două mii) <max2digitNumbers>;

<units> = unu | doi | trei | patru | cinci | şase | şapte | opt | nouă;

<elevens> = zece | unsprezece | doisprezece | douăsprezece | treisprezece | pais-
prezece | cincisprezece | şaisprezece | şaptesprezece | optsprezece | nouăsprezece;

<simpleTens> = douăzeci | treizeci | patruzeci | cincizeci | şaizeci | şaptezeci |
optzeci | nouăzeci;

<max2digitNumbers> = <units> | <elevens> | <simpleTens> [şi <units>];
```

Fig. 1. Romanian Dates Grammar (JSGF).

2.2. Decoding Parameters for ASR Systems

The effectiveness and accuracy of the speech recognition process depends directly on some key parameters. As discussed above, the final word sequence produced by the ASR system depends on the relative contributions of the acoustic and language models. In general, the acoustic model has a disproportionately large influence relative to that of the language model and this usually results in a large number of errors due to the insertion of many short words. Since they are short and have large variability a sequence of these models may provide the best acoustic match to short speech segments, even though the word sequence has very low probability according to the language model. The practical solution to this issue is to impose a word insertion penalty such that the probability of transitions between words is penalized. This penalty is modelled using an ASR decoding parameter called Word Insertion

Probability (WIP). For ASR systems based on rule grammars, this parameter is only important if the grammar comprises branches of different lengths or if it contains word loops (*e.g.* a grammar which allows any number of forenames).

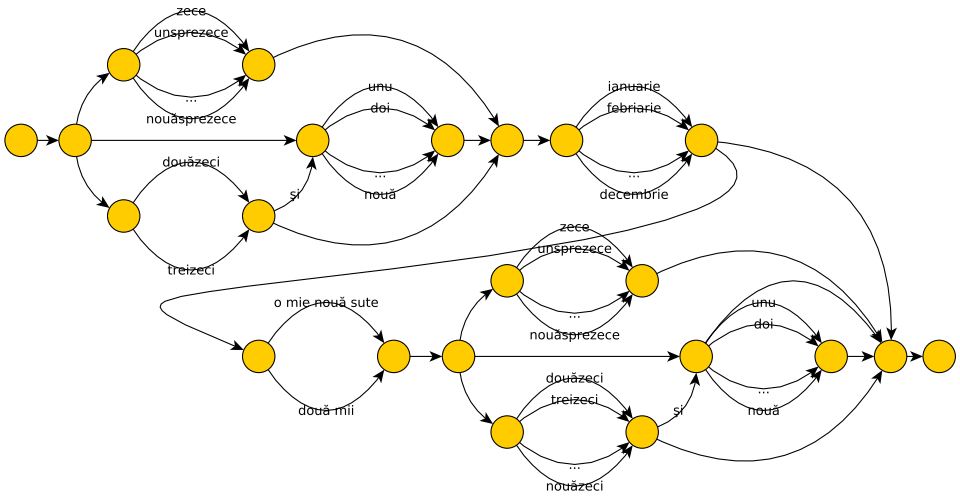


Fig. 2. Romanian Dates Grammar (FSM).

In particular for ASR systems which use rule grammars it is very important to detect and reject out-of-grammar (OOG) utterances. If such a mechanism is not implemented, then any speech utterance, including incorrect or invalid utterances, will be mapped to one of the possible word sequences defined by the grammar. There are three well known methods for rejecting these out-of-grammar utterances. The first one implies training and using a garbage model that will fit well any type of speech. The second method uses an OOG word model implemented as an all-phone self looping network introduced as a parallel branch of the regular grammar (see Fig. 3)[9, 10]. Typically context-independent phones are used to obtain reduced complexity and because it has been found that accuracy seems to be insensitive to context dependency in this all-phone loop. The third method of rejecting OOG utterances is based on confidence scores [11]. Confidence scores are usually computed based on the word lattice resulted from the decoding process. Typical features for computing confidence scores include: average acoustic score, average language score, word length in frames, word length in phones, the number of occurrence of the same word at the same location of the 10-best results, etc.

Out of these three methods the most popular one is the second (the out-of-grammar word model implemented as a phone loop) and, consequently, this will be the one used in our study. The effectiveness of this method is dependent on two key parameters (see Fig. 3):

- the probability of transitioning into the phone-loop (called Out-of-Grammar Probability - OOGP), and

- the probability of looping inside the phone-loop (called Phone Insertion Probability - PIP).

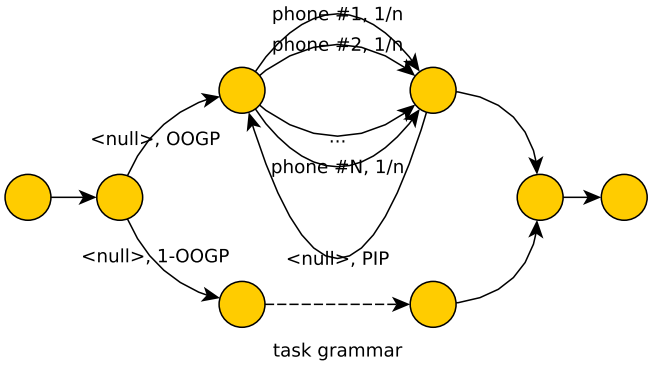


Fig. 3. Phone Loop and Typical Parameters in an OOG-rejection FSM Grammar.

The speech recognition process is a search problem. The models involved in the decoding process are used to create a search graph with all the possible alternative paths corresponding to various hypotheses of the uttered sequence of words. Speech recognition implies finding the most probable path (lowest cost path) through this search graph. Typically, the search graph is so large that it is infeasible to explore it all. Pruning mechanisms are used to eliminate parts of the graph from consideration, thus reducing the search cost. Path expansion and scoring is done progressively starting with the first speech frame and ending with the last one. Pruning involves eliminating some of the low probability partial paths obtained after each expansion and scoring step. Only the remaining paths are expanded further at the next step. There are several pruning methods, but the most popular one eliminates partial paths based on their score relative to the best scoring partial path. The relative threshold used in this pruning method is called Relative Beam Width (RBW) and the search method itself is called Beam Search. This is another key parameter which influences the speech recognition accuracy and performance. Using a too narrow or tight beam (too small RBW) can prune the best path and results in errors. Using a too large beam results in unnecessary computation in searching unlikely paths. One may also wish to set the beam to limit the computation (e.g. for real-time operation), regardless of recognition errors. The basic assumption behind pruning is the following: "as long as the lowest cost path is not eliminated by pruning, the same (optimal) result can be obtained by scoring fewer paths".

3. The SpeeD Grammar-based ASR System

The large vocabulary continuous speech recognition system developed by the Speech and Dialogue research group was recently extended to offer support for rule-

based grammar recognition. The various research and development issues encountered and solved are presented in this section.

3.1. The SpeeD LVCSR System

The SpeeD LVCSR system for the Romanian language was developed in 2011 and it is continuously being updated ever since. Several applications based on it are available of the laboratory's web page[5,6]. The LVCSR system is built upon the CMU Sphinx speech recognition toolkit[13]. More specifically, the decoding system uses the CMU Sphinx 4 Java decoder.

All our acoustic models are speaker-independent, 5-state HMMs with output probabilities modelled with GMMs. As speech features we typically use the classic Mel Frequency Cepstral Coefficients (MFCCs) plus their first and second temporal derivatives (13 MFCCs + deltas + double deltas). For applications in which noise robustness is of critical importance [4] we employ the recently introduced Power Normalized Cepstral Coefficients (PNCCs) plus their first and second temporal derivates (13 PNCCs + deltas + double deltas). In all cases the 36 phonemes in the Romanian language are modelled contextually (context dependent phonemes) with 4000 HMM senones. The number of Gaussian mixtures per senone state is variable (32/64/128), adapted to the size and variability of the training speech corpus. The acoustic models are created and optimized using the CMU Sphinx Toolkit.

The continuous speech language models are back-off, trigram, closed-vocabulary models. The vocabulary size (number of unigrams) is usually limited to 64k words (due to an ASR decoder implementation limitation). The language models are created with the SRI-LM Toolkit [14] using several large Romanian corpora (over 300M words) collected over the Internet, preprocessed and normalized by our research group. Preprocessing and normalization operations include, among others, (i) URLs, emails and abbreviations expansion, (ii) punctuation marks handling, (iii) numbers-to-text conversion, and (iv) diacritics restoration.

The pronunciation dictionary is always generated dynamically by an automatic phonetization system that takes the language model vocabulary and produces phonetic transcriptions for the words based on an already existing phonetic dictionary (for known words) and using a machine translation method for unknown words[12].

Recently, a speaker diarization module was added to the LVCSR system[15]. This feature enables the system to label various paragraphs of the transcriptions with speaker tags and even to identify specific, well-known people for which speaker recognition models were trained. The speaker diarization module is based on the LIUM speaker diarization toolkit [16].

3.2. Extending the Language and Pronunciation Models

The CMU Sphinx Toolkit also offers support for grammar-based speech recognition. The steps to create a grammar-based ASR system starting from the existing

[5]Speech-to-Text application: `http://speed.pub.ro/speech-to-text`
[6]Rich Speech Transcription service: `http://speed.pub.ro/live-transcriber`

LVCSR system are the following:

- create the rule grammars (language models) for the specific speech recognition tasks,
- create the pronunciation models (phonetic dictionaries) for the words in the corresponding vocabularies,
- optimize key decoding parameters discussed in Section 2.2.

As stated before, the SpeeD research group uses an automatic phonetization system to create pronunciation models so the second step can be automatically solved. Step 3 can be approached through experimentation once the rule-based grammars are created. For this proof of concept grammar-based speech recognition system several rule grammars (for the Romanian language) were created:

- a numbers grammar able to recognize rational numbers with up to three decimal places between minus one billion and plus one billion,
- a dates grammar able to recognize dates between 01.01.1900 and 31.12.2099,
- a cities grammar able to recognize Romanian cities,
- a forenames grammar able to recognize Romanian forenames,
- a surnames grammar able to recognize Romanian surnames, and
- a yes/no grammar able to recognize affirmative and negative clauses.

The dates grammar was already exemplified in Section 2.1 in the Figs. 1 and 2. It consists of three parts, one for recognizing the date, one for the month and an optional third part for the year.

The numbers grammar is by far the most complex among these rule grammars. It contains special parts for recognizing 9-digit, 6-digit and 3-digit integer numbers. The 9-digit integer numbers part is eventually composed with the 3-digit integer numbers part and the Romanian word for decimal point ("virgulă") to create a rational numbers rule grammar.

The Romanian cities grammar was created using a list with all the Romanian cities and their population. In this grammar (illustrated in Fig. 4), a probabilistic branch for each city models the probability that the user utters the name of the city. The probability of each city was intended to be proportional to the city's popularity. A rough estimate of the city's popularity is its number of inhabitants, consequently the probability was chosen to be proportional to the city's population.

The Romanian forenames and surnames grammars were created using lists of names collected over the Internet summing up to a total of more than 500 thousands names. Statistics regarding the frequency of occurrence of the names as well as statistics regarding the frequency of occurrence of compound forenames (forenames composed of multiple names) were extracted out of these lists. The statistics were

used in selecting the most popular 200 forenames and 200 surnames and in computing branch probabilities for the names. The forenames and surnames grammars are depicted in Figs. 5 and 6.

Finally, the yes/no grammar is a very simple grammar that allows only affirmative ("da" in Romanian) and negative clauses ("nu" in Romanian) possibly repeated two or three times.

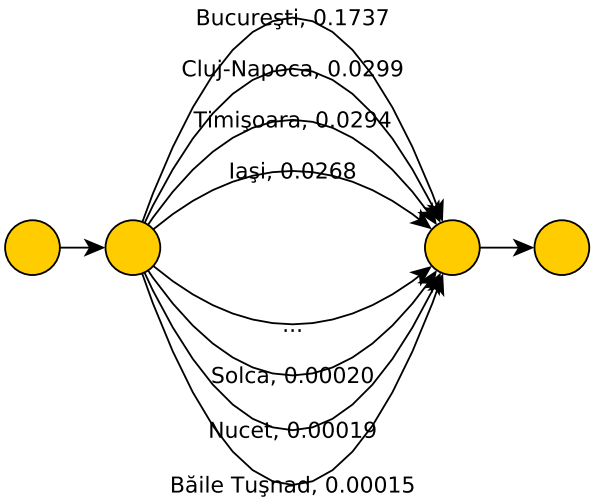


Fig. 4. Romanian Cities FSM Grammar.

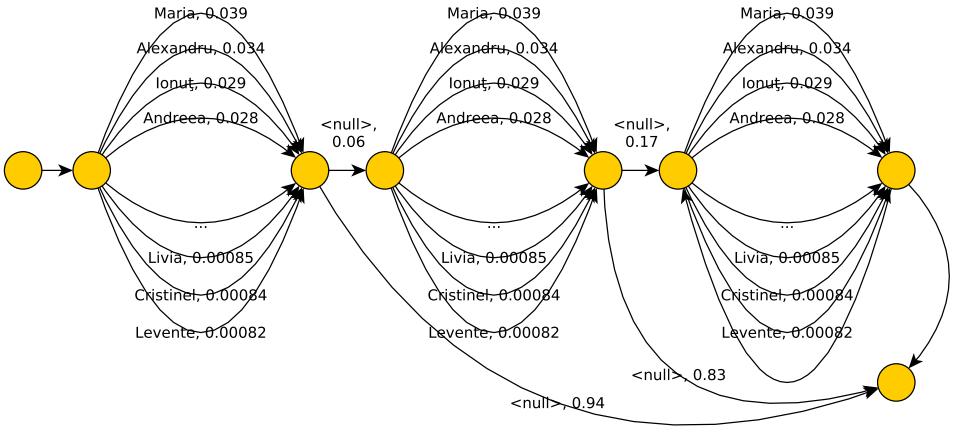


Fig. 5. Romanian Forenames FSM Grammar.

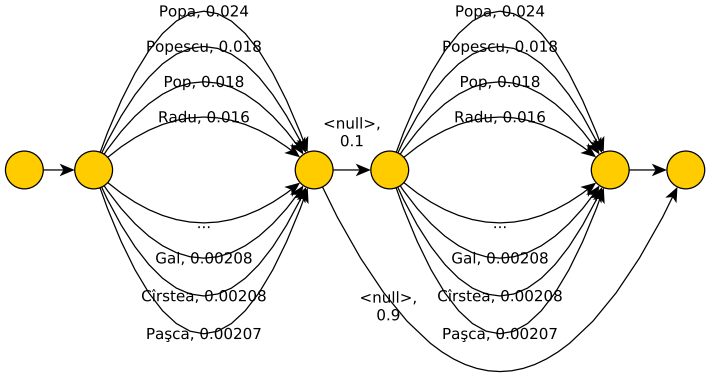


Fig. 6. Romanian Surnames FSM Grammar.

4. Informal Pronunciation of Romanian Numbers

In the previous section we focused mainly on language modelling issues (rule grammar design) and we stated that the pronunciation model for a new grammar (with a new vocabulary) can be easily created using the automatic phonetization tool we previously implemented [12]. However, the above statement does not take into account the following problem: in the Romanian language the numbers are heavily pronounced informally. In other words, although numbers are sometimes pronounced correctly, most of their occurrences in free speech are informal pronunciations. The automatic phonetization tool is able to create canonical pronunciations only and this leaves us with an incomplete pronunciation model for the dates and the numbers rule grammars. After a thorough study of Romanian numbers pronunciations we came to the conclusion that informal pronunciations occur more often in two digit numbers.

4.1. Romanian Two-Digit Numbers

Two digit Romanian numbers are formed similarly to two digit English numbers. Exactly as in English, there are two separate rules for the groups 10 – 19 and 20 – 99.

The numbers between 10 and 19 are written as compound-words formed by concatenating the unit words 1, 2, ..., 9 ("un", "doi", ..., "nouă" in Romanian), with the preposition "to" ("spre" in Romanian) and with the word "ten" ("zece" in Romanian). These numbers are summarized in Table 1. There are two exceptions (14 and 16) for which the unit word is slightly modified "pai" instead of "patru" and "şai" instead of "şase" (similarly to the English "fifteen").

The numbers between 20 and 99 are written as word phrases (separate words) by joining the compound word for tens 20, 30, ..., 90 ("douăzeci", "treizeci", ..., "nouăzeci" in Romanian) with the conjunction "and" ("şi" in Romanian) and with the unit words 1, 2, ...,9. A part of these numbers (30 – 39) are summarized in Table 2.

There are no exceptions to this composition rule.

Table 1. Two-digit Romanian Numbers (10 – 19)

Number	Text Version (English)	Text Version (Romanian)
10	ten	zece
11	eleven	unsprezece
12	twelve	doisprezece
13	thirteen	treisprezece
14	fourteen	paisprezece
15	fifteen	cincisprezece
16	sixteen	şaisprezece
17	seventeen	şaptesprezece
18	eighteen	optsprezece
19	nineteen	nouăsprezece

Table 2. Two-digit Romanian Numbers (30 – 39)

Two-digit Number	Text Version (English)	Text Version (Romanian)
30	thirty	treizeci
31	thirty-one	treizeci şi unu
32	thirty-two	treizeci şi doi
33	thirty-three	treizeci şi trei
34	thirty-four	treizeci şi patru
35	thirty-five	treizeci şi cinci
36	thirty-six	treizeci şi şase
37	thirty-seven	treizeci şi şapte
38	thirty-eight	treizeci şi opt
39	thirty-nine	treizeci şi nouă

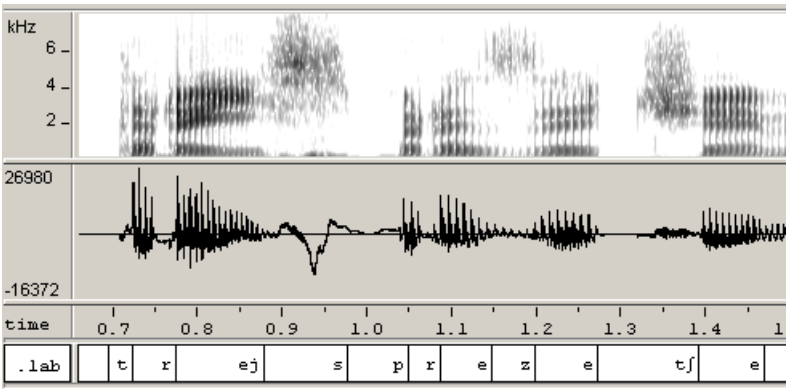
It is worth mentioning that, similarly to English, numbers with a more digits (3, 4, etc.) are formed by joining the hundreds, thousands, etc. words with the two-digit numbers. Consequently, the fact that two-digit numbers are usually pronounced informally affects the pronunciation of all Romanian numbers.

4.2. Within-word and Cross-word Pronunciation Variation

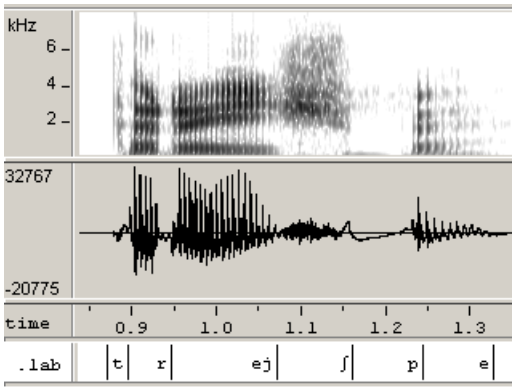
The two-digit numbers in the first group mentioned above (10 – 19) are usually pronounced informally by changing several syllables (within the compound word) into a shorter one. For example, the word "trei.spre.ze.ce" (13), formally pronounced /trej.spre.ze.ʧe/, is usually pronounced /trejʃ.pe/. As you can notice from this example, the syllables "spre", "ze" and "ʧe" have been merged and changed into "pe". Figure 7 illustrates this behaviour, showing both the correct, formal pronunciation

and the informal pronunciation of the number 13. This pronunciation variation can be seamlessly integrated in the pronunciation model by adding a second pronunciation for all these words.

Most of the two-digit numbers in the second group mentioned above (21 – 29, 31 – 39, ... 91 – 99) are written as three consecutive words (*e.g.* the number 36 is written "treizeci şi şase"). They are usually pronounced informally by un-pronouncing one or several syllables. There are two commonly used informal pronunciations for these numbers. The phrase "trei.zeci si sa.se" (36), formally pronounced /trej.zetʃ ʃi ʃa.se/, is usually pronounced /trej.ze ʃi ʃa.se/ (the /tʃ/ in the second syllable is missing) or /trej.ʃa.se/ (the second syllable and the second word are missing). Figure 8 illustrates this behaviour, showing both the correct, formal pronunciation and the informal pronunciation of the number 36. As exemplified, this pronunciation variation spreads across several words and cannot be integrated seamlessly in the pronunciation model, because this model stores words (not word sequences) pronunciations.

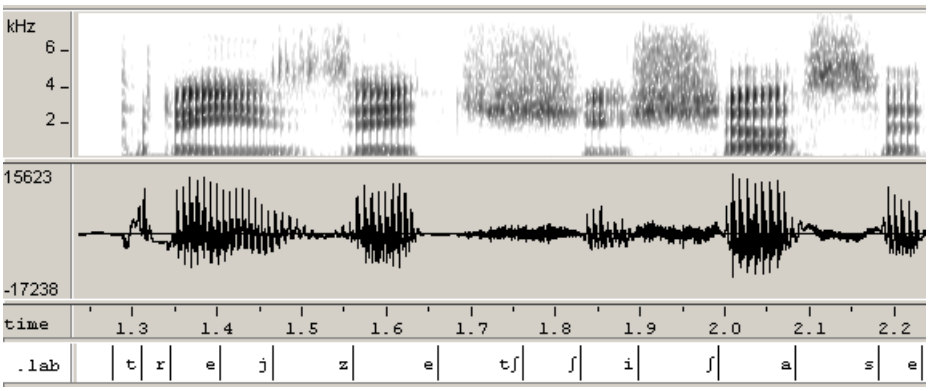


(a) Canonical Pronunciation of the Number 13

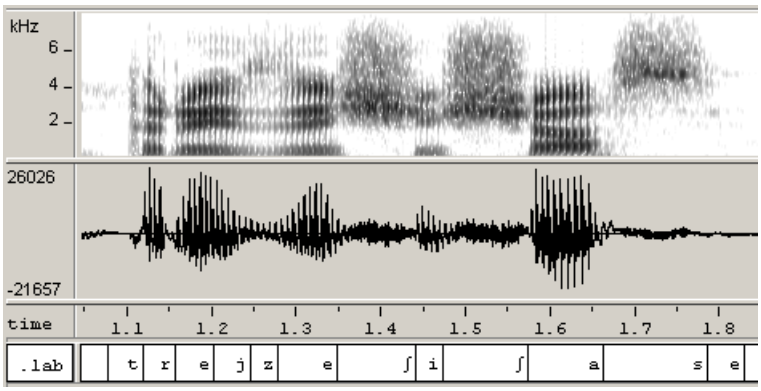


(b) Informal Pronunciation of the Number 13

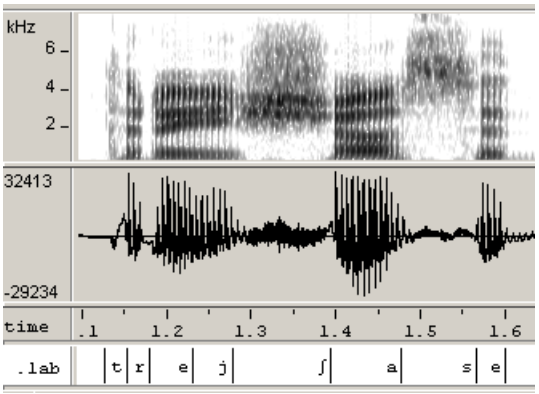
Fig. 7. Pronunciations of the Number 13 ("treisprezece" in Romanian).



(a) Canonical Pronunciation of the Number 36



(b) Informal Pronunciation #1 of the Number 36



(c) Informal Pronunciation #2 of the Number 36

Fig. 8. Pronunciations of the Number 36 ("treizeci şi şase" in Romanian).

4.3. Solution to Cross-word Pronunciation Variation

To the best of our knowledge this is the first study that deals with cross-word pronunciation for Romanian. To solve this problem we propose the following steps:

- identify the phrases for which cross-word pronunciation variation occurs (some two-digit numbers),
- merge the sequence words for which the variation was observed into a single compound word (the merging is done by inserting an underscore character, so that splitting is still possible),
- specify the various pronunciations for these compound words in the pronunciation model,
- modify the rule grammars to use these compound words instead of original sequences of words.

According to the above algorithm, the Romanian dates grammar represented in Fig. 2 was modified as shown in Fig. 9. The figure shows that the days and the last two digits of the year are modelled differently: using a single set of alternatives instead of compounding these numbers as sequences of one, two or three words. The Romanian numbers grammar mentioned in Section 3.2 was also updated accordingly.

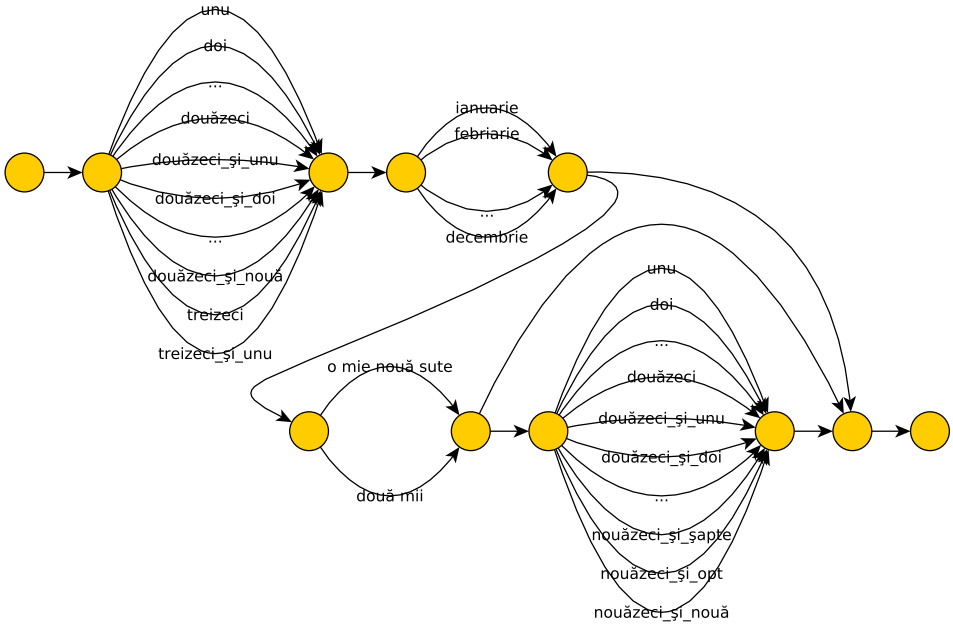


Fig. 9. Updated Romanian Dates Grammar (FSM)

5. ASR System Assessment

This section is intended to present the optimization process for the decoding parameters of the SpeeD grammar-based ASR system. The envisioned decoding parameters were described in Section 2.2.

All the experiments were performed using the acoustic model described in Section 3.1 and the various rule grammars described in Section 3.2.

The typical performance figures for automatic speech recognition: Word Error rate (WER) and Sentence Error Rate (SER) were used to assess and compare the accuracy of the system using different parameter configuration sets. The SER is the percentage of incorrectly transcribed sentences relative to the total number of sentence. The WER is the ratio between the number of word errors (substitutions, insertions, and deletions) and the total number of words in the reference transcription (Equation 1).

$$WER = \frac{S + D + I}{N} * 100. \tag{1}$$

The actual evaluation was done on a newly recorded speech corpus created specifically for assessing the grammar-based ASR system. This evaluation speech corpus is briefly presented in the next sub-section.

5.1. Evaluation Speech Corpora

The evaluation of rule grammar ASR systems can only be done on speech corpora which comprise task-specific utterances. For example, to assess the Romanian cities ASR system, one needs in-grammar utterances with city names and out-of-grammar utterances (with any other speech). The continuous speech corpora which were already available could not be used for this experiment, because they contained only out-of-grammar utterances. Consequently, we developed several new speech corpora, one for every ASR task: numbers, dates, cities, forenames, surnames and yes/no.

The corpora were created by recording various predefined phrases representing in-grammar utterances for the six ASR tasks mentioned above. The phrases were chosen randomly with the goal of covering as much as possible the rule grammars designed for these tasks. The recordings were made using an online recording application previously developed by the SpeeD research group. The 14 speakers were involved voluntarily in the speech corpus development process and were mostly university students. Some of them recorded all the phrases for all ASR tasks, while others recorded only partially these phrases. The details of the speech corpora are provided in Table 3.

It is worth mentioning that the utterances for the Numbers ASR task were recorded in a special manner. The speakers were asked to pronounce informally the first 150 utterances and formally the other 100 utterances. The Numbers speech corpus was designed and recorded in this manner so that it can be used to assess the (possible) negative effects of informal pronunciation of Romanian numbers.

Table 3. Evaluation Speech Corpora

ASR task	Utterances	Words	Speakers
Numbers	3000	12672	14
Dates	381	3452	7
Cities	480	600	4
Forenames	1120	1680	7
Surnames	720	840	6
Yes/No	250	380	5

5.2. Experimental Results

The three decoding parameters which were optimized in these experiments are: the word insertion probability (WIP), the out-of-grammar probability (OOPG) and the relative beam width (RBW). Their initial values in the Sphinx 4 decoding recipe were WIP=1E-50, OOGP=1E-85 and RBW=1E-70. The first series of experiments aimed at optimizing the WIP on in-grammar utterances. The experiments were done on separate in-grammar utterances for each ASR task (i.e. numbers utterances for the Numbers task, dates utterances for the Dates task, etc.). The results are summarized in Table 4.

Table 4. WIP optimization for RBW=1E-70 and OOGP=1E-85

WIP	**WER [%] on in-grammar uterrances**					
(1E-)	**Numbers**	**Dates**	**Cities**	**Forenames**	**Surnames**	**Yes/No**
2	5.1	1.0	1.5	15.1	10.0	27.9
5	4.9	1.0	1.5	12.2	8.6	24.7
10	4.8	0.9	1.5	9.9	6.5	17.6
20	5.0	0.9	1.5	7.7	4.5	7.4
30	6.4	1.2	1.5	6.7	3.3	2.1
40	9.7	3.8	1.5	5.9	3.2	1.1
50	17.3	12.3	1.5	6.2	2.6	1.1

As the results in Table 4 show, the optimum choice of the WIP parameter is not trivial. If the rule grammar of the ASR system allows word repetitions (word loops), then this parameter should be relatively small in order to penalize erroneous insertions. In our case, the Forenames, Surnames and Yes/No ASR tasks are such examples. The Forenames rule grammar allows the repetition of up to three forenames to accommodate multi-word forenames such as “Maria Ioana”. The Surnames rule grammar allows the repetition up to two surnames to accommodate multi-word surnames such as “Popescu Tăriceanu”. The Yes/No rule grammar allows the repetition of the affirmative or negative clause to accommodate utterances such as “da da da”. The Forenames rule grammar was illustrated in Fig. 5 and the Surnames grammar was illustrated in Fig. 6. Having in mind the above, please note that small values for the WIP trigger high word error rates (WERs) for all these three ASR tasks (e.g. WERs of 15.1, 10.0 and 27.9 for WIP=1E-2). Moreover, it is worth noting that for small WIPs the highest WER is obtained for the Yes/No task, for which the

insertion errors are more probable due to the shortness of the words: "yes", "no". On the opposite side, the WERs for the Surnames task are smaller because the number of allowed repetitions in this case is smaller.

Going further with the discussion on WIP optimization, if the rule grammar of the ASR system allows sporadic or scattered word insertions, then this parameter should be relatively high. The reason is not to wrongly penalize many-words utterances by merging multiple short words into one longer word. In our case, the Numbers and the Dates ASR tasks are such examples. A relatively high WIP for these ASR tasks leads to word-merging errors such as "optsprezece milioane" instead of "opt sute zece milioane".

Finally, if the rule grammar of the ASR system does not allow word insertions at all, then this parameter is irrelevant. In our experiment, the Cities ASR task is based upon such a grammar (see Fig. 4) that restricts the number of words per utterance (only one city name in this case). Therefore, the WER for this ASR task is always the same, irrespective of the WIP.

The second series of experiments aimed at optimizing the relative beam width (RBW) on in-grammar utterances. The results are summarized in Table 5. As the results in Table 5 show, irrespective of the ASR task, for small or narrow beams (RBW between 1E-30 and 1E-50) the word error rate is quite high. As expected, as the beam is increased the accuracy of the speech recognition system gets higher (the WER is lower). However, the results also show that for beams larger than a certain threshold (RWB $>$ 1E-70) the decrease in WER is insignificant. Given that any beam enlargement transposes into computational costs, it is clear that a RBW around 1E-70 is a good compromise if the purpose is to obtain maximum ASR accuracy.

Table 5. RBW optimization for OOGP=1E-85 and WIP=1E-10

RBW	**WER[%] on in-grammar utterances**					
(1E-)	**Numbers**	**Dates**	**Cities**	**Forenames**	**Surnames**	**Yes/No**
30	10.9	5.7	5.2	19.0	13.6	8.2
40	6.7	3.0	1.8	13.0	7.6	8.2
50	5.4	1.7	1.8	10.8	7.3	8.2
60	4.8	1.0	1.5	10.1	6.5	8.2
70	4.8	0.9	1.5	9.9	6.5	8.2
80	4.7	0.9	1.5	9.8	6.7	8.2
90	4.6	0.8	1.2	9.8	6.8	8.2
100	4.6	0.6	1.2	9.8	6.8	8.2
110	4.6	0.6	1.2	9.8	6.8	8.2

The third series of experiments aimed at optimizing the OOGP on in-grammar and out-of-grammar utterances. The results are summarized in Table 6. As the results show, the out-of-grammar probability (OOGP) has little effect when decoding in-grammar utterances. This is understandable because for in-grammar utterances the cost of passing through the task grammar is much lower than passing through the OOG word model (the all-phone loop) and the cost of entering the OOG word model (expressed by the OOGP) is almost irrelevant. Of course, the OOGP cannot be too high (i.e. larger than 1E-2) because this would lead to an unnatural bias towards

the all-phone loop. However, for out-of-grammar utterances the OOGP is of critical importance. Only high values for this probability (between 1E-2 and 1E-10) lead to a relatively effective rejection of out-of-grammar utterances. For low values, the cost of passing through the task grammar and generating an output is small enough and triggers high WERs. A particular negative case is the Forenames ASR task whose rule grammar allows an indefinite number of forenames to be uttered. Consequently, the Forenames ASR system usually finds a close matching forename for every word in the OOG utterance and outputs a list of names as the hypothesis for any utterance.

Table 6. OOGP optimization for WIP=1E-10 and RBW=1E-70

OOGP	**WER[%] on IG utterances**						**WER[%] on OOG utterances**					
(1E-)	**Num**	**Dat**	**Cit**	**For**	**Sur**	**Y/N**	**Num**	**Dat**	**Cit**	**For**	**Sur**	**Y/N**
2	5.2	0.9	1.8	9.9	7.1	12.6	4.8	2.4	25.7	43.8	4.7	1.6
5	5.1	0.9	1.8	9.9	6.9	11.6	6.0	3.5	28.4	52.2	5.8	4.7
10	5.0	0.9	1.8	9.9	6.9	10.3	8.6	4.6	34.0	72.2	8.4	5.5
20	4.8	0.9	1.8	9.9	6.9	9.2	17.8	6.8	46.8	155.1	25.7	15.0
30	4.7	0.9	1.7	9.9	6.7	8.7	38.9	17.4	66.9	336.2	76.6	37.0
40	4.7	0.9	1.5	9.9	6.5	8.2	74.2	40.6	92.6	598.7	152.2	78.7
50	4.7	0.9	1.5	9.9	6.5	8.2	380.4	342.8	118.0	890.0	200.0	299.0

Finally, the last series of experiments aimed at assessing the (possible) negative effects caused by the informal pronunciations of Romanian numbers. This experiment was performed only for the Numbers ASR task and only on in-grammar utterances (only these are relevant). The decoding parameters optimized above were set for this experiment as follows: OOGP=1E-10, WIP=1E-20, RBW=1E-70.

As mentioned in the previous section, the speakers who recorded the Numbers evaluation speech corpus were asked to pronounce informally a part of the utterances and formally the other part. Three experimental setups were evaluated on these two parts of the Numbers speech corpus (see Table 7). In the first setup informal pronunciations are not modelled at all (neither within-word, nor cross-word variations). In the second setup within-word informal pronunciation variations are inserted into the pronunciation model for the two-digit numbers between 11 and 19 (these numbers are written with single words, see Table 1). Finally, in the third setup, the rule grammar is modified to model two-digit numbers between 20 and 99 as single, compound words (these numbers are normally written with several words, see Table 1) and cross-word informal pronunciation variations (for these numbers) are inserted into the pronunciation model.

Table 7 shows that the pronunciation modelling approaches proposed in this paper have significant beneficial effects for the task of recognizing informally pronounced numbers. Within-word pronunciation modelling of informal numbers brings a relative WER improvement of 14% over the baseline. Furthermore, cross-word pronunciation modelling of informal numbers brings a relative WER improvement of 63% over the previous pronunciation modelling technique. As expected, on formally pronounced numbers the results are more or less the same regardless of the pronunciation modelling technique. However, it is interesting to see that even in the third setup the WER

on informally pronounced numbers is much higher than the WER on formally pronounced numbers. This means that there is still room for improving the recognition of informally pronounced numbers.

Note that all the optimization experiments discussed earlier in this section for the Numbers ASR task were performed using the third setup.

Table 7. The effects of informal pronunciations of Romanian numbers

Pronunciation Model	**Rule Grammar**	**WER[%] on IG utterances**		
		Num (inform)	**Num (form)**	**Num (all)**
Formal pronunciations	Regular numbers grammar	26.2	2.4	16.9
+ within-word informal pronunciations	Regular numbers grammar	22.5	1.8	14.5
+ cross-word informal pronunciations	+ 20-99 modelled with compound words	8.2	2.2	6.0

6. Conclusions

This paper presented the process employed by the SpeeD research group to create a proof-of-concept for a grammar-based ASR system. The effort started from the already existing large vocabulary continuous speech recognition system and involved creating rule grammars for a series of common ASR tasks (numbers, dates, names, etc.), and extending the phonetic dictionary.

A more subtle issue identified and solved was the problem of informal pronunciation for Romanian numbers. More specifically, the pronunciation of two-digit numbers poses interesting and difficult problems because the pronunciation variation also extends over several words. As the experimental results section showed, the pronunciation modelling approaches proposed in this paper have significant beneficial effects for the task of recognizing informally pronounced numbers. Within-word pronunciation modelling of informal numbers brings a relative WER improvement of 14% over the baseline. Furthermore, cross-word pronunciation modelling of informal numbers brings a relative WER improvement of 63% over the previous pronunciation modelling technique.

The study also compared statistical language models and rule grammars (as alternative linguistic supports for ASR) and described the main ASR decoding parameters. In the experimental section these decoding parameters were optimized (on various tasks: numbers, dates, cities, etc.) in order to find the best setup for the grammar-based ASR system. This proof-of-concept system is available online.

Further research could be done to assess the negative effects of informal Romanian numbers pronunciations for continuous speech recognition. Although numbers do not appear in continuous speech as often as in the Numbers ASR task, they are almost always pronounced informally and thus the impact on word error rate could also be significant.

Acknowledgements. This work has been partially funded by the Sectoral Operational Programme "Human Resources Development" 2007-2013 of the Ministry of European Funds through the Financial Agreement POSDRU/159/1.5/S/134398 and partially by the PN II Programme "Partnerships in priority areas" of MEN - UEFISCDI, through project no. 32/2014.

References

[1] CUCU H., *Towards a speaker-independent, large-vocabulary continuous speech recognition system for Romanian*, PhD Thesis, University Politehnica of Bucharest, 2011.

[2] CUCU H., BESACIER L., BURILEANU C., BUZO A., *Enhancing Automatic Speech Recognition for Romanian by Using Machine Translated and Web-based Text Corpora*, Proc. Int. Conf. Speech and Computer (SPECOM), Kazan, Russia, 2011, pp. 81–88.

[3] TARJAN B., MOZSOLICS T., BALOG A., HALMOS D., FEGYO T., MIHAJLIK P., *Broadcast news transcription in Central-East European languages*, Proc. Int. Conf. Cognitive Infocommunications (CogInfoCom), Kosice, Slovakia, 2012, pp. 59–64.

[4] CUCU H., BUZO A., PETRICA L., BURILEANU D., BURILEANU C., *Recent Improvements of the SpeeD Romanian LVCSR System*, Proc. Int. Conf. Communications (COMM), Bucharest, Romania, 2014, pp. 111–114.

[5] Rabiner L.R., *A tutorial on hidden Markov models and selected applications in speech recognition*, Proc. IEEE, vol. **77**, no. 2, pp. 257–286, 1989.

[6] JELINEK F., *Statistical Methods for Speech Recognition*, The MIT Press, Cambridge, MA, 1998.

[7] JURAFSKY D., MARTIN J., *Automatic Speech Recognition*, *Speech and Language Processing: An introduction to natural language processing, computational linguistics, and speech recognition* (2nd Ed.), Pearson Education, 2009.

[8] RENALS S., HAIN T., *Speech Recognition*, *Handbook of Computational Linguistics and Natural Language Processing*, 2010.

[9] BAZZI I., GLASS J., *Modeling out-of-vocabulary words for robust speech recognition*, Proc. Int. Conf. on Speech and Language Processing (Interspeech), 2000.

[10] BAZZI I., GLASS J., *A multi-class approach for modelling out-of-vocabulary words*, Proc. Int. Conf. on Speech and Language Processing (Interspeech), 2002.

[11] JIANG H., *Confidence measures for speech recognition: A survey*, Speech Communication, Vol. **45**, No. 4, pp. 455–470, 2005.

[12] CUCU H., BUZO A., BESACIER L., BURILEANU C., *SMT-based ASR Domain Adaptation Methods for Under-Resourced Languages: Application to Romanian*, Speech Communication, Vol. **56** - Special Issue on Processing Under-Resourced Languages, pp. 195–212, 2014.

[13] LAMERE P., KWOK P., WALKER W., GOUVEA E., SINGH R., RAJ B., WOLF P., *Design of the CMU Sphinx-4 decoder*, Proc. Int. Conf. Interspeech, Geneve, Switzerland, 2003, pp. 1181–1184.

[14] STOLCKE A., *SRILM - an extensible language modeling toolkit*, Proc. Int. Conf. on Speech and Language Processing (Interspeech), 2002, pp. 257–286.

[15] BUZO A., CUCU H., PETRICA L., BURILEANU D., *An Automatic Speech Recognition Solution with Speaker Identification Support*, Proc. Int. Conf. Communications (COMM), Bucharest, Romania, 2014, pp. 119–122.

[16] ROUVIER M., DUPUY G., GAY P., KHOURY E., MERLIN T., MEIGNIER S., *An Open-source State-of-the-art Toolbox for Broadcast News Diarization*, Proc. Int. Conf. Interspeech, Lyon, France, 2013.