On Formatting Transcriptions of Romanian Speech

Horia Cucu*, Andi Buzo, Alexandru Caranica, Corneliu Burileanu
Speech and Dialogue Research Laboratory, University Politehnica of Bucharest, Romania

Abstract: Automatic speech recognition (ASR) is the process of automatically transcribing a spoken audio message into text. Besides the fact that the transcription generated by an ASR system is not 100% correct, it is also characterized by low reading intelligibility, because it is not organized in paragraphs, it does not contain any punctuation marks, all text is lowercased, the numbers and dates are written with words instead of digits, etc. This paper reports on the recent advances made by the Speech and Dialogue research group in the field of ASR transcription formatting for the Romanian language. The complete rich speech transcription system developed by our research group is briefly discussed and the transcription post-processing framework, which is responsible for text formatting, is presented in detail. Concrete examples of transcription formatting are given and insightful details on the algorithms designed and used are provided. The impact of transcription formatting and future work is also discussed.

Key Words: automatic speech recognition, transcription formatting, speech recognition intelligibility

1. INTRODUCTION

Automatic speech recognition (ASR) addresses the problem of mapping an acoustic signal to a sequence of words. When the input acoustic signal contains speech uttered by different speakers, the ASR task can be regarded as a two-step process: speaker diarization (who spoke when?) and speech-to-text transcription (what did he say?). Automatic speech recognition 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].

Large vocabulary continuous speech recognition (LVCSR) is a subclass of ASR, which aims at transcribing speech possibly containing most 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 four LVCSR systems developed for the Romanian language. In 2011 we published the first LVCSR results for Romanian (Cucu, 2011a; Cucu, 2011b), in August 2012 Google launched their online speech recognizer[2] for Android and Chrome. THINKTech Research Center[3] published a paper (Tarjan, 2012) on broadcast news recognition for Romanian in December 2012 and, finally, Vocapia Research[4] reported on a Romanian LVCSR system they developed recently in 2014 (Vasilescu, 2014).

The output of an Automatic Speech Recognition (ASR) system consists of raw text, often in lowercase format and without any punctuation information. The transcript is intended to be as close as possible to the speech content of the audio file (Buzo, 2014). This may be useful for a wide range of applications, such as database indexing and classification, where a machine uses this information in search related algorithms. For other tasks, where humans need to easily read and understand the text (e.g. subtitling, dictation and broadcast news transcription), capitalization, punctuation restoration and numbers formatting greatly improves the readability of automatic speech transcripts.

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 (Cucu, 2014). We also proposed recently a capitalization and punctuation restoration system for the Romanian language based on textual information only (Caranica, 2015). This paper builds upon previous work and presents the post-processing framework (based on words statistics and prosody), which formats the transcriptions generated by the ASR system, with the purpose of increasing their intelligibility. The emphasis will be on: (i) paragraph separation, (ii) formatting numbers and dates (words to

[1] Speech and Dialogue Research Group: http://speed.pub.ro
[2] Google ASR System: http://officialandroid.blogspot.ro/2012/08
[3] THINKTech Research Center: http://thinktech.hu
[4] Vocapia Research: http://www.vocapia.com

digits conversion), (iii) restoration of punctuation marks and (iv) capitalization of named entities. The proof-of-concept system discussed in this paper is available online[5].

2. AUTOMATIC SPEECH RECOGNITION

2.1 Applications and challenges

Automatic speech recognition has a wide range of applicability. The most important domain seems to be that of hands-free and eyes-free interfaces to computers or other devices. There are many applications in which the users need to use their hands and eyes for something else and speech remains their only alternative to being efficient. Moreover, as emphasized in the previous section, speech is the most natural mean of communication for human beings. Other major application areas are spoken dialogue systems for call centers and speech-to-speech translation systems. Speech-to-speech translation is at this moment a very hot topic in many academic and industrial research centers. Finally, ASR is applied to dictation: transcription of an extended monologue by a single specific speaker. Dictation is common in several fields, such as law, where many trials or official meetings need to be transcribed for further reference. Each of these applications is typically more restrictive than the general problem which requires the automatic transcription of naturally spoken continuous speech, by an unknown speaker, in any environment. The various sources of speech variability, which will be discussed further on, make the general task a very challenging one. Nevertheless, in many practical situations, the variability is restricted. For example, there may be a single, known speaker, or the speech to be recognized may be carefully dictated text rather than a spontaneous conversation, or the recording environment may be quiet and non-reverberant. In speech-to-text transcription, a distinction is made between parts addressing acoustic variability (acoustic modeling), and parts addressing linguistic uncertainty (language modeling).

One of the most important factors which influence the difficulty of the speech transcription process is the specific *speech recognition task*. This includes the language, the size of the vocabulary to be recognized and the linguistic uncertainty of the domain. Different spoken languages present different challenges for a speech recognizer. For a large number of languages there are very few speech and text resources available. These so-called *low-resourced languages* are spoken by a large number of people, but no prior work of collecting and organizing speech and/or text resources has been done. Consequently the task of designing an ASR system has to include resource collection also. There are other languages and dialects which are mostly spoken and have practically no written resources for language modeling. In this case the situation is even worse, because there is no way of acquiring the language resources and, in general, the linguistic rules are very loose.

Other languages "suffer" from a complex morphology. For example *rich-morphology languages* such as French and Romanian have larger vocabularies than poor-morphological languages such as English. In Romanian the present tense of the verb *to go* has five morphologically different forms: "merg", "mergi", "merge", "mergem", "mergeţi", "merg", while in French it has six: "vais", "vas", "va", "allons", "allez", "vont". In English, the same verb has only two morphologically different forms: "go", "goes". German and Turkish are some of the so-called *agglutinative languages*. In these languages a large number of new words can be formed by concatenation of morphemes. This also leads to larger vocabularies and consequently makes automatic speech recognition a more challenging task.

The *size of the vocabulary* is an important factor because it is obvious that a digits recognition task (with a ten words vocabulary) is much simpler than a spontaneous telephone speech recognition task (with a 64k words vocabulary). Nevertheless, larger vocabularies do not always mean a more difficult ASR task. The *linguistic uncertainty* of the possible speech utterances also plays a significant role. For example, a tourism-specific ASR task with a 64k words vocabulary which mostly contains proper names (places, restaurants, hotels, etc.) is not as difficult as a spontaneous telephone speech recognition task with an equal-size vocabulary. The low linguistic uncertainty (perplexity) of the first task makes it less difficult.

Another important factor which influences the difficulty of the speech process is the speaking style. The speaking style refers to how fluent, natural or conversational the speech is. Obviously, *isolated words speech* recognition, in which each word is surrounded by some sort of pause, is much easier than recognizing *continuous speech* in which words run into each other and have to be segmented. In fact, in the early days of automatic speech recognition, systems solved the problem of where to locate word boundaries by requiring the speaker to leave pauses between words: the pioneering dictation product Dragon Dictate (Baker, 1989) is a good example of a large-vocabulary isolated words recognition system.

[5] SpeeD rich speech transcription system: http://speed.pub.ro/live-transcriber

Continuous speech tasks themselves vary greatly in difficulty. For example, the task of recognizing *read speech* is much easier than the task of recognizing more natural styles of speech such as *conversational* or *spontaneous speech*. The greater acoustic variability makes the latter task more challenging. This difference in difficulty between continuous speech tasks is reflected in the increased word error rates for spontaneous speech recognition compared with the recognition of read speech.

The difficulty and consequently the accuracy of the speech recognition process is also influenced by the *acoustic environment* in which the speech is recorded, along with any *transmission channel*. Outside of quiet offices and laboratories, there are usually multiple acoustic sources including other talkers, environmental noise and electrical or mechanical devices. In many cases, it is a significant problem to separate the different acoustic signals found in an environment. The microphone used for recording also has a significant impact on the speech recognition accuracy. Commercial dictation systems and most of the laboratory research in speech recognition are done with high-quality, head-mounted microphones. Other types of microphones come with different disadvantages which contribute to the quality of the ASR system. Variations in transmission channel occur due to movements of the talker's head relative to the microphone and transmission across a telephone network or the internet. Probably the largest disparity between the accuracy of automatic speech recognition compared with human speech recognition occurs in situations with high *additive noise*, *multiple acoustic sources*, or *reverberant environments*. Noisy speech with a low signal-to-noise ratio can cause the word error rates to go up by 2 to 4 times compared to clean speech.

Finally, the speaker characteristics have also a significant impact on the accuracy of a speech recognizer. The variability in speaker characteristics resides in the speaker accent, the language/dialect he uses, whether he is a native or a non-native speaker, the speech rate, the speaker age and of course the differences in the speech production anatomy and physiology. Moreover, different speakers exhibit different degrees of intrinsic variability based on the emotional state, temporary health problems, etc. The inter-speaker variability can be dealt with by designing *speaker-dependent* ASR systems. The drawback here is that a new acoustic model has to be created for every new speaker. This leads to a more complex system, but also raises several trainability issues (insufficient training data for every speaker and others). On the other hand, *speaker-independent* ASR systems are simpler and more flexible (they can be used to recognize the speech of any speaker). Nevertheless, a speaker-independent system is less accurate for a given speaker when compared to a speaker-dependent system for that particular speaker (if sufficient training data is available for the speaker). Although speaker adaptation algorithms have made great progress over the past 15 years, it is still the case that the adaptability and robustness to different speakers exhibited by automatic speech recognition systems is very limited compared with human performance.

The speaker characteristics variability is evident and very annoying in native versus non-native speech. Although human beings can understand quite well non-native speech, the automatic speech recognition systems exhibit very limited robustness when they are required to recognize this type of speech. Several studies reported huge differences in performance for native versus non-native speech on the same ASR task. For example, the word error rate on Vietnamese-accented French and Chinese-accented French has been reported to be about 5 times higher than for native speakers on the same task (Tan, 2008). Similarly, the word error rate on Korean-accented English has been reported to be about 9 times higher than for native speakers (Oh, 2007). Obviously, the differences also depend on the speaking level for the non-natives and on the relationship between the two languages. For example, (Wang, 2003) reports that the word error rate on German-accented English is only 3 times higher than for native speakers on the same task. Nevertheless, non-native speech recognition is still an open issue and a high number of studies (among which we mention (Tan, 2007; Oh, 2007; Tan, 2008; Sam, 2010)) have been published in the past few years on this subject.

The state-of-the-art paradigm for large vocabulary continuous speech recognition is the hidden Markov model (HMM). For LVCSR in particular, the HMM-based acoustic model is used in conjunction with an n-gram model which is responsible for the language modeling part. Statistical language models (n-grams) have become the state-of-the-art solution for language modeling since the tremendous expansion of the Internet, which provided enough data to suitably train these systems.

2.2 The achitecture of an asr system

The automatic speech recognition (ASR) process addresses the problem of mapping an acoustic signal to a sequence of words. This task is also called speech-to-text transcription. ASR is one of the first fields in which data-driven, machine learning, statistical modeling approach became standard. The basic statistical framework was created and developed during almost two decades by Baker (Baker, 1975), a team at IBM (Jelinek, 1976;

Bahl, 1983) and a team at AT&T (Levinson, 1983; Rabiner, 1989). The speech-to-text task can be formulated in a probabilistic manner as follows:
What is the most likely sequence of words W in the language L, given the speech utterance X?*

The formal representation uses the argmax function, which selects the argument that maximizes the word sequence probability:

$$W^* = \arg\max_{W} p(W \mid X) \tag{2.1}$$

Equation 2.1 specifies the most probable word sequence as the one with the highest posterior probability, given the speech utterance. Bayes rule is used to compute this posterior probability and the most probable word sequence becomes:

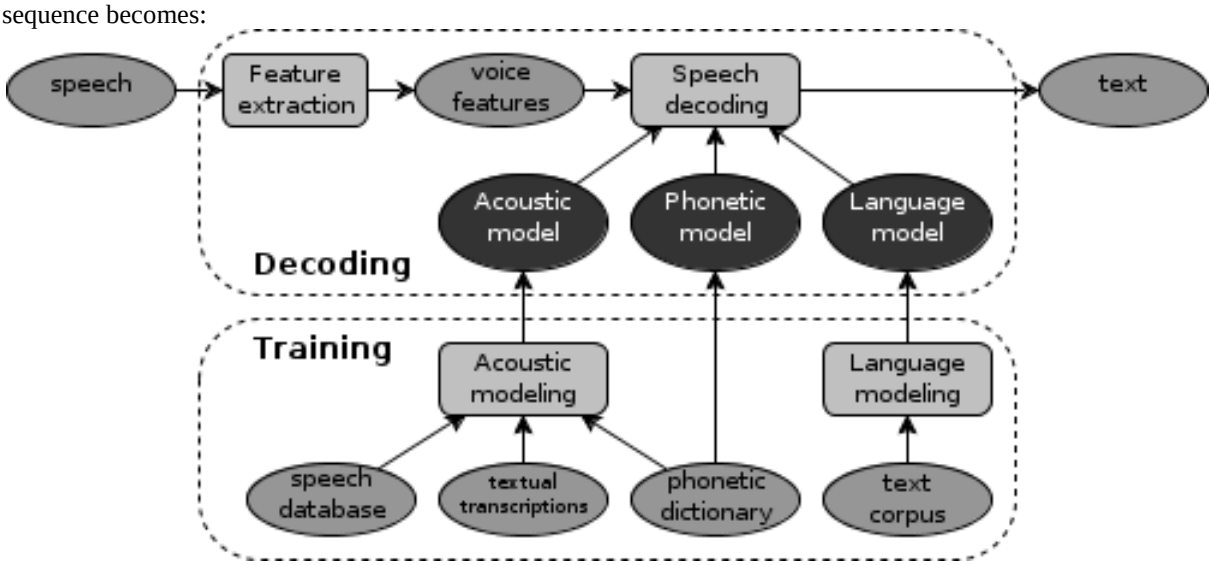

Figure 1 ASR system architecture

$$W^* = \arg\max_{W} \frac{p(X \mid W)\,p(W)}{p(X)} \tag{2.2}$$

p(X), the probability of the speech utterance is independent of the sequence of words *W*, thus it can be ignored. Consequently, Equation 2.2 becomes:

$$W^* = \arg\max_{W} p(X \mid W)\,p(W) \tag{2.3}$$

Equation 2.3 exhibits two interesting factors which can be estimated directly. The initial problem (of estimating the word sequence given the speech utterance) has now been split into two simpler problems: a) estimating the prior probability of the word sequence *p(W)* and b) estimating the likelihood of the acoustic data given the word sequence *p(X|W)*. The probability of the word sequence can be estimated using solely a *language model*, while the likelihood of the acoustic data given the words sequence can be computed based on an *acoustic model*. The two models can be constructed independently as shown in Figure 1, but will be used together to decode a speech utterance as specified in Equation 2.3. Figure 1 presents the architecture of an ASR system and also shows the methods and type of data required in the development phase.

The acoustic model has the role of estimating the likelihood of the spoken message, given the sequence of words. In state-of-the-art LVCSR systems, the acoustic model does not model words as basic speech units because (i) every speech recognition task comes with its own vocabulary of words for which there are not already trained acoustic models available and (ii) the number of words in any language is too large.

Consequently, instead of words, LVCSR systems model sub-lexical acoustic units, called phonemes, or, more recently, even sub-phonetic acoustic units, called senones. The acoustic model comprises a set of models for phonemes or senones, which can be connected during the speech recognition process to form models for words and then sequences of words. These are eventually used to estimate the likelihood that the spoken message contains a specific sequence of words.

The language model 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 $W = w_1, w_2, \ldots, w_n$, 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 favor of the first phrase.

The phonetic or pronunciation model is used to link the acoustic model (which estimates the acoustic likelihood of phonemes) to the language model (which estimates the probability of word sequences). A phonetic model is usually a pronunciation dictionary that maps all the words in the vocabulary to one or several sequence of phonemes representing the pronunciation of those words. The phonetic dictionary can be regarded as an interface between the acoustic model, which models phonemes and the language model which models words.

Figure 1 illustrates the processes involved in the development of an ASR system and the resources needed to create the acoustic, language and pronunciation models. The acoustic model is trained on a corpus of audio clips comprising speech and their corresponding transcriptions. In the case of LVCSR systems, which use statistical language models, large corpora of text are needed to create the language models. Small-vocabulary ASR systems usually use rule-grammars, which do not require any additional text or acoustic resources. Figure 1 also shows that the ASR system does not model speech directly (at the waveform level). A feature extraction block is employed to extract specific acoustic features which are further used to create the acoustic model. Consequently, the same feature extraction block is also needed and used in the decoding process.

3. FORMATTING ASR TRANSCRIPTIONS: PROPOSED ALGORITHM AND RESULTS

3.1 Speed's rich speech transcription system

The rich speech transcription system developed by the Speech and Dialogue Research Group is presented in Figure 2. Apart from the automatic speech recognition core, this system comprises a speech pre-processing frontend, which is responsible with voice activity detection and speaker diarization, and a transcription post-processing framework, which is responsible with transcription formatting. The pre-processing frontend was discussed in (Buzo, 2014).

Voice activity detection is needed in order to split the raw audio signal into segments comprising speech and segments comprising music, noise, silence, etc. Obviously, only the speech segments will be further processed. The voice activity detection block associates the output speech segments with timestamps relative to the initial speech signal. This timing information can be used in the end to associate speech transcriptions with the various parts of initial speech signal.

Speaker diarization is the process of segmenting a speech signal based on the speakers that uttered the corresponding signals. Speaker diarization practically answers the questions "who spoke when?" by generating speech segments associated with speaker information (speaker ids). This information is used in the post-processing framework to associate speech transcriptions with the corresponding speakers. The speaker diarization block also preserves the timing information associated with the speech segments.

The transcription post-processing framework uses the speaker information and the timestamps associated with the raw, unformatted transcriptions to organize them into paragraphs, insert punctuation marks and capitalize the text. The restoration of punctuation marks and capitalization are performed using statistical linguistic information (Caranica, 2015) and acoustic-related information. Moreover, the post-processing framework formats numbers and dates (converts numbers written with words into numbers written with digits) creating a more intelligible transcription. Two examples of the impact of transcription post-processing on intelligibility are presented in Figure 3 and Figure 4.

3.2 Transcription post-processing framework

The transcription post-processing framework is presented in Figure 5. The unformatted transcriptions, associated with timestamps and speaker information, are formatted sequentially by four processing blocks. First, numeric entities are identified and formatted. Second, the transcription is segmented into paragraphs. Finally, the punctuation marks are restored and the text is capitalized in a two-step process: a data-driven approach based on statistical linguistic information and a knowledge-based approach using acoustic-related information.

It is worth mentioning that the post-processing framework was designed to preserve the word-level timestamps and the segment-level speaker information available in the unformatted transcription. The design of the post-processing framework also took into account the fact that the processing has to be performed online, i.e. each unformatted transcription part generated by the ASR system has to be processed right away, before the entire transcription of the whole audio file is available. This poses additional problems, because formatting a transcription part depends on the last words in the previous transcription part, on potential speaker changes between parts, on the time difference between parts, etc.

Numbers formatting is the first transcription post-processing operation and it is performed in a knowledge-based manner using text-to-digits conversion rules. Although these rules are specific for the Romanian language many of them can be considered general, because Romanian cardinal numbers are formed similarly to English cardinal numbers:

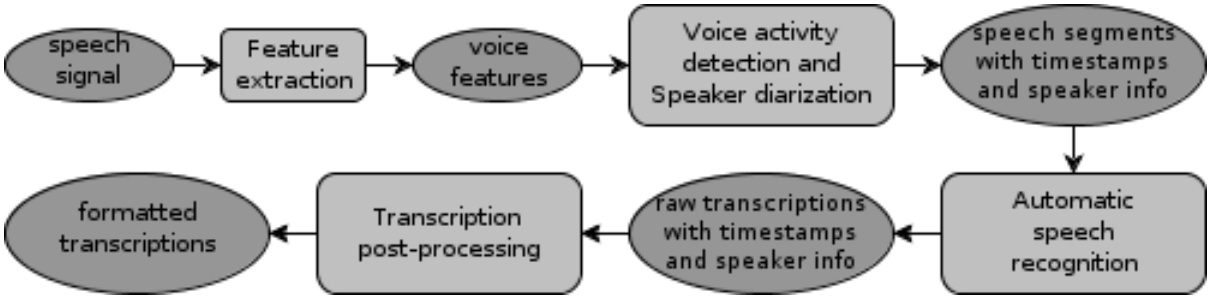

Figure 2 Rich speech transcription system

actorul jean constantin a fost transferat astăzi la spitalul din constanţa institutul de boli cardiovasculare cc iliescu din bucureşti pe care l-a îngrijit până acum spun că pacientul şi-a dorit să fi adus în capitală pentru investigaţii suplimentare cu detalii rămân surioara sa regală şi spune să aranjăm constantin ajuns aici la institutul ce ce iliescu din capitală în această dimineaţă cu o ambulanţă alături de el se află soţia sa şi medicii l-au internat la secţia de cardiologie unde le monitorizează permanent să-ţi specialiştii spun că starea lui este stabilă acum dar nu vor să intre în detalii pentru că familia pacientului a cerut că discreţia sa constantin are optzeci de ani şi a fost săptămâna trecută internat la spitalul judeţean din constanţa şi pentru că avea dureri în piept şi respira cu dificultate şi de îndată acesta sunt informaţiile la revedere

a) Speech transcription before post-processing

Speaker #1: Actorul Jean Constantin a fost transferat astăzi la Spitalul din Constanţa, Institutul de Boli Cardiovasculare cc. Iliescu din Bucureşti pe. Care l-a îngrijit. Până acum spun că pacientul şi-a dorit să fi adus în Capitală pentru investigaţii suplimentare. Cu detalii. Rămân surioara. Sa. Regală.

Speaker #2: Şi spune să aranjăm Constantin ajuns aici la Institutul ce. Ce Iliescu din Capitală. În această dimineaţă cu o ambulanţă. Alături de el se află soţia sa şi. Medicii l-au internat la secţia de cardiologie unde le monitorizează permanent să-ţi Specialiştii spun că starea lui este. Stabilă . Acum, dar nu vor să intre în detalii, pentru că familia pacientului a cerut că discreţia. Sa Constantin are 80 de ani şi a fost săptămâna trecută, internat la Spitalul Judeţean din Constanţa. Şi pentru că avea dureri în piept şi respira cu dificultate şi de îndată acesta sunt informaţiile. La revedere.

b) Speech transcription after post-processing

Andreea Esca: Actorul Jean Constantin a fost transferat astăzi de la spitalul din Constanţa la Institutul de Boli Cardiovasculare C.C. Iliescu din Bucureşti. Medicii care l-au îngrijit până acum spun că pacientul şi-a dorit să fie adus în capitală pentru investigaţii suplimentare. Cu detalii despre starea maestrului vine Ioana Şanta. Bună seara Ioana.

Ioana Şanta: Bună seara. Jean Constantin a ajuns aici la Institutul C.C. Iliescu din capitală în această dimineaţă cu o ambulanţă. Alături de el se află soţia sa. Medicii l-au internat la secţia de cardiologie unde îl monitorizează permanenţă. Specialiştii spun că starea lui este stabilă acum, dar nu pot să intre în detalii pentru că familia pacientului a cerut discreţie. Jean Constantin are 81 de ani şi a fost săptămâna trecută internat la Spitalul Judeţean din Constanţa pentru că avea dureri în piept şi respira cu dificultate. Deocamdată acestea sunt informaţiile. La revedere.

c) Ideal speech transcription

Figure 3 The importance of ASR transcription post-processing. Example #1.

pe douăzeci aprilie două mii treisprezece la palatul parlamentului din bucureşti a avut loc o conferinţă de presă la conferinţă au participat peste optzeci de persoane din marile oraşe ale ţării timişoara cluj-napoca iaşi şi altele premierul victor ponta şi preşedintele României traian băsescu au prezentat un plan comun de rezolvare a problemelor ţării printre altele s-a discutat despre restituirea unei tranşe de cinci virgulă douăzeci şi şapte la sută din datoria externă a României adică suma de cinci milioane o sută de mii de euro

a) Speech transcription before post-processing

Speaker #1: Pe 20 aprilie 2013. La Palatul Parlamentului din Bucureşti. A avut loc o conferinţă de presă. La conferinţă au participat peste 80 de persoane din marile oraşe ale ţării, Timişoara cluj-napoca, Iaşi şi altele.

Speaker #1: Premierul Victor Ponta şi preşedintele României, Traian Băsescu. Au prezentat un plan comun de rezolvare a problemelor ţării . Printre altele s-a discutat despre restituirea unei tranşe de 5,27%. Din datoria externă a României, adică suma de 5.100.000 de euro.

b) Speech transcription after post-processing

Horia: Pe 20 aprilie 2013, la Palatul Parlamentului din Bucureşti, a avut loc conferinţă de presă. La conferinţă au participat peste 80 de persoane din marile oraşe ale ţării: Timişoara, Cluj-Napoca, Iaşi şi altele.

Horia: Premierul Victor Ponta şi preşedintele României, Traian Băsescu, au prezentat un plan comun de rezolvare a problemelor ţării. Printre altele s-a discutat despre restituirea unei tranşe de 5,27% din datoria externă a României, adică suma de 5.800.000 de euro.

c) Ideal speech transcription

Figure 4 The importance of ASR transcription post-processing. Example #2.

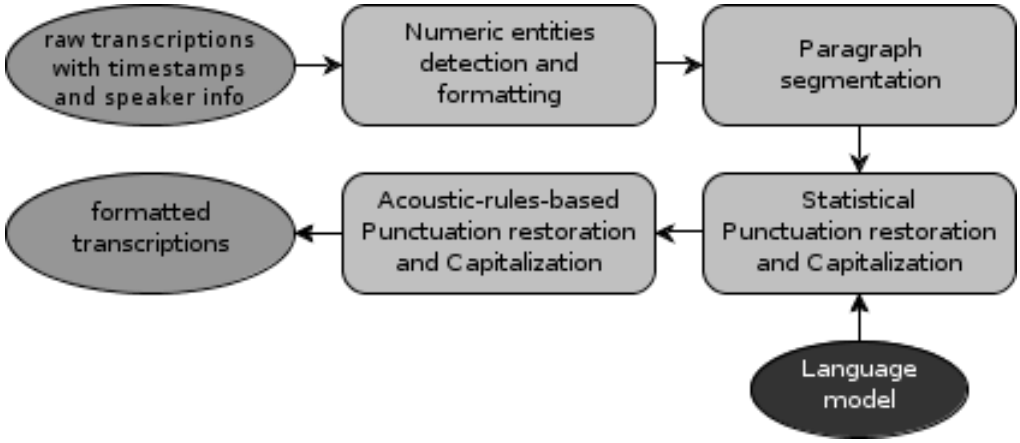

Figure 5 Transcription post-processing framework

- There are special words denoting digits: “a”/”an”/”one” (“o”, “un”, “unu”, “una” in Romanian), “two” (“doi”, “două” in Romanian), …, “nine” (“nouă” in Romanian).
- Two-digit numbers between 10 and 19 are compound words formed by concatenating the unit words: 1, 2, ..., 9 (”unu”, ”doi”, ..., ”nouă” in Romanian), with the preposition ”to” (”spre” in Romanian) and with the word ”ten” (”zece” in Romanian). For example, 15 is “cincisprezece”. 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 exception “fif” instead of “five” in “fifteen”).
- The numbers between 21 and 99 are written as sequences of words obtained 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. For example, 36 is he number 36 is written “treizeci şi şase”. There are no exceptions to this composition rule. However, as described in (Cucu, 2015), in order to be able to model colloquial pronunciations of these numbers, our ASR system models them as artificial compound words by merging the words with an underscore (e.g. ”treizeci şi şase” => ”treizeci_şi_şase”)
- Numbers with more than three digits are formed based on two-digit numbers and other special words denoting hundreds (“sută”, “sute” in Romanian), thousands (“mie”, “mii” in Romanian), etc. For example, “thirty two thousand six hundred fifty seven” is “treizeci şi două de mii şase sute cincizeci şi şapte” in Romanian.

Romanian ordinal numbers are formed by joining the corresponding cardinal number with a feminine (“a”) or a masculine (“lea”) suffix. For example, “the twenty-seventh boy” would be “al douăzeci şi şapte*lea* băiat” in Romanian and “the twenty-seventh girl” would be “a douăzeci şi şapte*a* fată” in Romanian.

The various numeric entities that were taken into account to be formatted by the post-processing framework are positive and negative rational cardinal numbers, ordinal numbers, dates and times. The algorithm designed and applied to format these numeric entities consists of five passes through the list of words in the transcription, as follows:

- Pass #1.
  - 1.1 Single-word cardinal numbers and artificial compound words are converted from words to integer tokens (e.g. "trei" => 3; "douăzeci_şi_cinci" => 25; "doisprezece" => 12).
  - 1.2 Single-word ordinal numbers are split into an integer and a string suffix ("treilea" => 3 "-lea"; "trezeci şi şaptelea" => 37 "-lea", etc.)
  - 1.3 Multipliers are replaced with the corresponding integer numbers ("sută"/"sute" => 100; "mie"/"mii" => 1000; "milion"/"milioane" => 1000000). Exceptions:
    - "de" before a multiplier is deleted
    - "o"/"un" before multiplier is replaced with 1
    - "la sută" is replaced with "%"
    - a non-numeric entity followed by "mie" is left unchanged because "mie" has several meanings: "thousand" or "me"
- Pass #2. The transcription is iterated from left to right and the integer tokens 100, 1000 and 1000000 are merged with their left neighbor (LN) and their right neighbor (RN) to form a single integer token with the value 100 | 1000 | 1000000 * LN + RN.
- Pass #3. The word "comma" ("virgulă" in Romanian) representing the decimal separator (as opposed to English, which uses "point") is merged with its left neighbor (LN) and its right neighbor (RN) to form a single token with the value (LN + 0.RN).

Table 1 Examples of numbers formatting output after each algorithm pass

Input	şapte sute optzeci_şi_trei de milioane trei sute optzeci_şi_nouă de mii şaptezeci_şi_nouă virgulă trei sute
Pass #1	7 100 83 1000000 3 100 89 1000 79 virgulă 3 100 euro
Pass #2	783389079 virgulă 300 euro
Pass #3	783389079.3 euro
Pass #4	783389079.3 euro
Pass #5	783.389.079,3 euro

Input	pe data de trei ianuarie o mie trei sute optzeci_şi_nouă s-a intamplat ceva
Pass #1	pe data de 3 ianuarie 1 1000 3 100 89 s-a intamplat ceva
Pass #2	pe data de 3 ianuarie 1389 s-a intamplat ceva
Pass #3	pe data de 3 ianuarie 1389 s-a intamplat ceva
Pass #4	pe data de 3 ianuarie 1389 s-a intamplat ceva
Pass #5	pe data de 3 ianuarie 1389 s-a intamplat ceva

Input	o scădere de minus zero virgulă şaptesprezece la sută în sondaje
Pass #1	o scădere de minus 0 virgulă 17 % în sondaje
Pass #2	o scădere de minus 0 virgulă 17 % în sondaje
Pass #3	o scădere de minus 0.17 % în sondaje
Pass #4	o scădere de -0.17% în sondaje
Pass #5	o scădere de -0,17% în sondaje

Input	a obtinut locul al douăzeci_şi_cincilea
Pass #1	a obtinut locul al 25 -lea
Pass #2	a obtinut locul al 25 -lea
Pass #3	a obtinut locul al 25 -lea
Pass #4	a obtinut locul al 25 -lea
Pass #5	a obtinut locul al 25-lea

- Pass #4. The word "minus" (same in Romanian) is merged with its right neighbor (RN) to form a single token with the value -RN.
- Pass #5. Cardinal numbers with more than 3 digits are separated using the thousands separator ("." in Romanian), with one exception: numbers representing years are not separated. Cardinal numbers followed by ordinal prefixes ("-a" or "-lea") from which they were separated in pass #1 are now remerged to form a single token (e.g. "27-lea").

Table 1 presents some examples phrases comprising various types of numeric entities and the intermediate format of the phrases after each algorithm pass.

The paragraph segmentation block is designed to decide whether the current transcription part belongs to the previous paragraph (i.e. the paragraph to which the previous transcription part also belongs) or to a new paragraph. This processing block uses acoustic information only: silence fillers and speaker changes. The current transcription part is marked as belonging to a new paragraph (i) if it is the first transcription part of the audio file or (ii) if the previous transcription part belongs to a different speaker or (iii) if the silence duration between the previous transcription part and the current transcription part is longer than a predefined threshold. The information regarding paragraph segmentation is used by both the subsequent blocks.

As Figure 5 shows, punctuation restoration and capitalization is performed in two steps. The first step uses a data-driven approach based on statistical linguistic information (i.e. occurrence probabilities for punctuation marks and capitalized words in specific contexts). This approach was proposed and discussed in (Caranica, 2015). Practically, an n-gram language model, comprising the occurrence probabilities for punctuation marks and capitalized words, is used to compute (i) the likelihoods that a punctuation mark should or should not be inserted after a certain word, in a certain context, and (ii) the likelihoods that a word should or should not be to be capitalized in a certain context. In the actual implementation, this approach also uses information regarding whether the current transcription part is the first part in a new paragraph or not. If it is not, then the processing is performed on the current transcription part prefixed by the last few words in the previous transcription (constituting "the history" of the first word in the current transcription part).

The second step uses acoustic-related information (non-speech/silence fillers, time difference between consecutive words and speaker changes) to restore the punctuation and capitalize words in a knowledge-based manner, i.e. based on some human-designed rules. The rules we designed and used are the following:

- Consecutive non-speech/silence fillers are merged into a single silence filler. Its duration is computed as the sum of the durations of the composing fillers. These silence fillers will be replaced with punctuation marks based on their duration (see following rules).
- A period is inserted at the end of the previous transcription part if the current transcription part belongs to a new paragraph.
- A comma is inserted at the beginning of the current transcription part if the current transcription part belongs to the same paragraph as the previous transcription part and if the time difference between the two transcription parts is longer than a predefined threshold.
- A period is inserted instead of a silence filler if the time difference between the adjacent words is longer than a predefined threshold. If the time difference is smaller than the threshold, then a comma is inserted instead of that silence filler. If the time difference is very small, the silence filler is simply deleted. Exceptions:
    - No punctuation mark is inserted if the previous token is already a punctuation mark (previously inserted by the statistical restoration module).
    - No punctuation mark is inserted if the silence filler is the first token in the first transcription part of a new paragraph.
- Any time a period is inserted the next word is also capitalized.

Two examples of speech transcriptions before and after post-processing were presented in Figure 3 and Figure 4. Note that the most important process that contributed to the intelligibility of the post-processed transcription is the paragraph segmentation. Without paragraph segmentation, a long audio file would be transcribed into a non-intelligible sequence of words separated by spaces. Numeric entities formatting is also very important in improving the readability of the transcription. In the example in Figure 4 there are several numeric entities that were formatted. It is worth noting that the benefit obtained by formatting a simple number in an easy context: "eighty people" => "80 people" ("optzeci de persoane" => "80 de persoane", in Romanian) is minimal, but the benefit obtained by formatting large amounts of money or dates is truly significant. In the example, the phrase "on the twenty of april twenty thirteen" ("pe douăzeci aprilie două mii treisprezece", in Romanian) is formatted as "on 20 April 2013" ("pe 20 aprilie 2013", in Romanian) and the phrase "five million one hundred thousand Euro" ("cinci milioane o sută de mii de euro") is formatted as "5,100,000 Euro" ("5.100.000 de euro", in Romanian).

Punctuation and capitalization are also important in obtaining a readable transcription. The enumeration "timişoara cluj-napoca iaşi and others" ("timişoara cluj-napoca iaşi şi altele", in Romanian) representing some of the most important cities in Romania is eventually formatted as "Timişoara cluj-napoca, Iaşi and others" ("Timişoara cluj-napoca, Iaşi şi altele", in Romanian). Although not entirely correct (the second named entity, Cluj-Napoca, is not capitalized and not preceded by a comma), the formatted version of the phrase is clearly more readable. Sentence detection and the insertion of periods do not work very well: the punctuation mark is usually inserted when the speaker pauses and this rarely corresponds to the end of the sentence, especially in spontaneous speech. However, even if they are not sentence ends, speaker pauses marked with period increase the intelligibility of the transcription. For example, the formatted phrase "On 20 April 2013. At the Parliament Palace in Bucharest. A press conference took place." ("Pe 20 aprilie 2013. La Palatul Parlamentului din Bucureşti. A avut loc o conferinţă de presă.", in Romanian) is still intelligible although the first two periods should have been commas.

4. CONCLUSION

This paper presented the transcription post-processing framework we developed to increase the intelligibility of the ASR transcriptions generated by our LVCSR system for the Romanian language. The paper discussed many examples of ASR transcriptions which are practically unusable by a human operator due to their low intelligibility. The solutions proposed to (i) convert numeric entities written with words into digits, (ii) separate the text into paragraphs, (iii) insert punctuation marks and (iv) capitalize the text were discussed in detail and concrete examples were provided for each case.

In the near future we plan to extend the numeric entities detection and formatting algorithm in order to be able to format more types of numeric entities. Moreover, we plan to extend the acoustic-rules-based punctuation restoration and capitalization module to take into account other prosody features such as the F1 contour. The statistical punctuation restoration and capitalization module can also be improved by extending the text corpus on which the n-gram language model was trained.

ACKNOWLEDGEMENTS

This work was supported in part by the Sectoral Operational Programme "Human Resources Development" 2007-2013 of the Ministry of European Funds through the Financial Agreements POSDRU /159/1.5/S/132395 and POSDRU /159/1.5/S/134398 and in part by the PN II Programme "Partnerships in priority areas" of MEN - UEFISCDI, through project no. 332/2014.

REFERENCES

Bahl, 1983 L. R. Bahl, F. Jelinek and R. L. Mercer, "A maximum likelihood approach to continuous speech recognition," IEEE *Trans. Pattern Anal. Machine Intell.*, Vol. PAMI-5, pp. 179-190, 1983.

Baker, 1975 Baker, J., "The DRAGON system – an overview," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 23, no. 1, pp. 24-29, 1975.

Baker, 1989 Baker, J., "DragonDictate – 30K: Natural language speech recognition with 30,000 words," *Eurospeech 1989*, pp. 161-163, 1989.

Buzo, 2014 A. Buzo, H. Cucu, L. Petrică and D. Burileanu, "An Automatic Speech Recognition Solution with Speaker Identification Support," in Proc. Int. Conf. Communications (COMM), Bucharest, Romania, 2014, pp. 119-122.

Caranica, 2015 A. Caranica, H. Cucu, A. Buzo, C. Burileanu, "Capitalization and Punctuation Restoration for the Romanian Language", *University "Politehnica" of Bucharest Scientific Bulletin*, Series C, (in press 2015).

Cucu, 2011a H. Cucu, "Towards a speaker-independent, large-vocabulary continuous speech recognition system for Romanian," PhD Thesis, University Politehnica of Bucharest, 2011.

Cucu, 2011b H. Cucu, L. Besacier, C. Burileanu, A. Buzo, "Enhancing Automatic Speech Recognition for Romanian by Using Machine Translated and Web-based Text Corpora," in Proc. Int. Conf. Speech and Computer (SPECOM), Kazan, Russia, 2011, pp. 81-88.

Cucu, 2014 H. Cucu, A. Buzo, L. Petrică, D. Burileanu and C. Burileanu, "Recent Improvements of the SpeeD Romanian LVCSR System," in Proc. Int. Conf. Communications (COMM), Bucharest, Romania, 2014, pp. 111-114.

Cucu, 2015 H. Cucu, A. Caranica, A. Buzo, C. Burileanu, "On transcribing informally-pronounced numbers in Romanian speech," in Proc. Int. Conf. Telecommunications and Signal Processing (TSP), Prague, Czech Republic, (in press, 2015).

Jelinek, 1976 F. Jelinek, "Continuous speech recognition by statistical methods," *Proceedings of the IEEE*, vol. 64, pp. 532-536, 1976.

Levinson, 1983 S.E. Levinson, L.R. Rabiner and M.M. Sondhi, "An introduction to the application of the theory of probabilistic functions of Markov process to automatic speech recognition," *Bell System Technical Journal*, Vol.62, No.4, pp. 1035-1074, 1983.

Oh, 2007 Oh, Y.R., Yoon, J.S., Kim, H.K., "Acoustic model adaptation based on pronunciation variability analysis for non-native speech recognition," Speech Communication, 49, pp.59-70, 2007.

Rabiner, 1989 L.R. Rabiner "A tutorial on Hidden Markov Models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, pp. 257–286, 1989.

Sam, 2010 Sam, S., Castelli, E., Besacier. L., "Unsupervised Acoustic Model Adaptation for Multi-Origin Non Native ASR," *Interspeech 2010*, pp. 254-257, Tokyo, Japan, 2010.

Tan, 2007 Tan, T.-P., Besacier. L., "Acoustic Model Interpolation for Non-Native Speech Recognition," *ICASSP 2007*, pp. 1009-1012, Honolulu, USA, 2007.

Tan, 2008 Tan, T.-P., *Automatic Speech Recognition for Non-Native Speakers*. PhD Thesis, University Joseph Fourier, Grenoble, France, 2008.

Tarjan, 2012 B. Tarjan, T. Mozsolics, A. Balog, D. Halmos, T. Fegyo, P. Mihajlik, "Broadcast news transcription in Central-East European languages," in Proc. Int. Conf. Cognitive Infocommunications (CogInfoCom), Kosice, Slovakia, 2012, pp. 59-64.

Vasilescu, 2014 Ioana Vasilescu, Bianca Vieru, and Lori Lamel. "Exploring Pronunciation Variants for Romanian Speech-to-Text Transcription," in Proc. Int. Workshop on Spoken Language Technologies for Under-resourced Languages (SLTU), pp. 161–168, 2014.

Wang, 2003 Wang, Z., Schultz, T., Waibel, Alex, "Comparison of acoustic model adaptation techniques on non-native speech," *ICASSP 2003*, vol.1, pp. 540-543, Hong Kong, 2003.