Advanced Engineering Forum *Online: 2013-06-27*
ISSN: 2234-991X, Vols. 8-9, pp 29-36
doi:10.4028/www.scientific.net/AEF.8-9.29

Romanian Language Voice Browsing for Web Applications Using Grapheme Level Acoustic Modeling

József Domokos[1, a], László Sándor[2, b], Ovidiu Buza[1,c] and Gavril Toderean[1,d]

[1]Communications Department, Technical University of Cluj-Napoca, Bariţiu 26-28, Cluj-Napoca, Romania

Electrical Engineering Department, Sapientia University, Sighişoarei 1/C, Corunca, Romania

[a]domi@ms.sapientia.ro, [b]guitonchaque@yahoo.com, [c]ovidiu.buza@com.utcluj.ro, [d]toderean@pro3soft.ro (József Domokos)

Keywords: Romanian language continuous speech recognition, multimodal interface, voice commands, voice navigation, grapheme based acoustic model, HMM, Viterbi decoding, Java Web application.

Abstract. The aim of this article is to present a demonstrative Web application with Romanian language continuous speech recognition based multimodal interface. The scope of the paper also includes the presentation and testing of the capabilities of a context dependent grapheme based acoustic model for the Romanian language. The article describes the system architecture, the Web application development and the speech database used for the acoustic feature vector construction and acoustic model training. Further the task grammar is presented. At the end recognition results are presented in both offline and online operating mode. The used speech corpora together with the transcriptions are freely available for academic use on the NaviRo project website: http://users.utcluj.ro/~jdomokos/naviro/.

Introduction

Speech is the most natural way of communication between human beings. This can explain the commercial success of modern large vocabulary continuous speech recognition (LVCSR) systems and their applications such as dictation systems, telephone speech transcription and call center applications, speaker independent automatic broadcast news transcription and indexing, lectures transcription, meetings transcription, voice command applications, voice search and so on [1].

In the past years there has been a significant growth in multimodal human computer interface (HCI) research including speech recognition based HCIs [2]. But speech recognition systems are mainly developed for English language although the used techniques are mostly language independent. The lack of freely usable linguistic resources is the main problem in development of LVCSR systems for other languages.

Romanian language is a resource scarce language [3][4]. There are just a few freely usable language resources such as transcribed and annotated speech corpora, phonetic transcription dictionaries. In this way only an insignificant number of speech recognition systems exist to deal with the Romanian language. Excepting the recently introduced Google's Voice Search, there is no other fully functional free to use LVCSR system for Romanian although there are reported several Romanian language ASR systems [5][6]. This is the main reason which calls for further research in this domain.

The components of a LVCSR system are presented in Fig. 1.

Front-End Processing is a language independent task and the role of this subsystem is to extract the acoustic feature vector from the time windows o f the speech signal. The most widely used feature vector set for LVCSR systems consist of 13 Mel Frequency Cepstral Coefficients (MFCCs), calculated within a 25 ms Hamming window, with a 50% overlap between the windows. In addition to this static cepstra the speed and acceleration coefficients (also called delta and delta-delta or double delta) are also used to form the 39 coefficients of the final feature vector. Often the first MFCC is replaced by the energy in the feature vector [1][7][8].

To increase the robustness of the feature vector, cepstral mean subtraction (CMS) and speaker based cepstral coefficients mean and variance normalization is commonly performed [1].

Acoustic Modeling subsystem is dominated by Hidden Markov Models (HMMs) over the past decades. This model shows the best performances in LVCSR systems which handle large vocabularies of words, making impossible to collect enough representation for each word to train word HMMs [9]. Instead LVCSR systems use subword units like phones, syllables or graphemes that occurs more often in reasonably size speech corpora.

The most widely used subword units are the phonemes and context dependent phonemes (i.e. triphonemes or quintphonemes). The latest has the advantage that it can handle the effects of coarticulation. In a typical language there are tens of thousands triphone units [9]. This makes difficult to train speaker independent triphone HMMs for under resourced languages.

Also phonetic transcription is a time consuming task for resource scarce languages with no phonetic transcription dictionaries. Graphemes can successfully substitute phonemes in case of phonetic languages. As Romanian is a language with strong phonetic nature, a good choice in this case seems to be to use context dependent trigraphemes as subword units and to build HMMs for these trigraphemes. In this way it is possible to skip the grapheme-to-phoneme conversion task. There is some recent research reported in the literature about using grapheme models instead of phoneme models for speech segmentation [10].

Language Modeling. N-gram statistical models are the most widely used language models to represent the apriori probability of a word sequence in a LVCSR system [1][7]. Smoothing the probabilities of a language model is essential to deal with the unseen words from the training corpus. Kneser-Ney smoothing algorithm is the state of the art in this domain [1]. It outperforms all other smoothing algorithms in LVCSR applications.

Decoding. The role of decoder is to search for the optimal sequence of words given the sequence of observed acoustic feature vector. Decoder use information from the acoustic model and the language model to perform search. The most widely used search methods are the Viterbi decoding and language model rescoring.

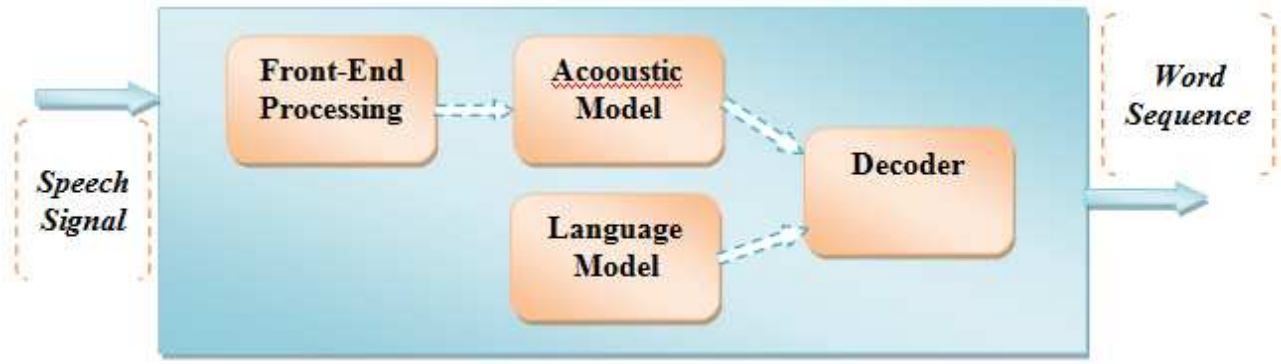

Fig. 1. - LVCSR sytem components

The aim of this article is to present a Romanian language voice driven Internet browsing demo application and to show the capabilities of a context dependent trigraphemes based acoustic model trained on a small continuous speech corpus. The presented demo application shows how can be added speech recognition based multimodal interface to any Web application.

The remaining part of this paper is organized as follows: Firstly the system architecture with a detailed deployment diagram of the demo application is presented. The following section describes the speech database used for the acoustic feature vector construction and acoustic model training. Further the task grammar is presented and the Web application is described. Finally the recognition results are presented in both offline and online operating mode with some discussions on it. Conclusions and references are presented at the end of the paper.

System Architecture

Whole system architecture is depicted in Fig. 2 as a deployment diagram. The client side components and the server side components are separately presented.

At the client there is no need to perform any installation or configuration. The client machine must be equipped with sound card and a microphone, and must have installed an Internet browser. This configuration can be found on every standard PC or laptop computer. The client side application runs in the client browser as a simple HTML page and contains a light-weight Flash application to capture audio through the client Web browser, a JavaScript for transmitting recorded audio to the server trough the Internet using a HTTP Post request. Before starting voice command we must grant the access to the microphone and start recording from the Flash interface. The recorded speech signal is then streamed to the server. The audio recording and transmission is realized using the WAMI (Web-Accessible Multimodal Applications) Toolkit [11][12][13].

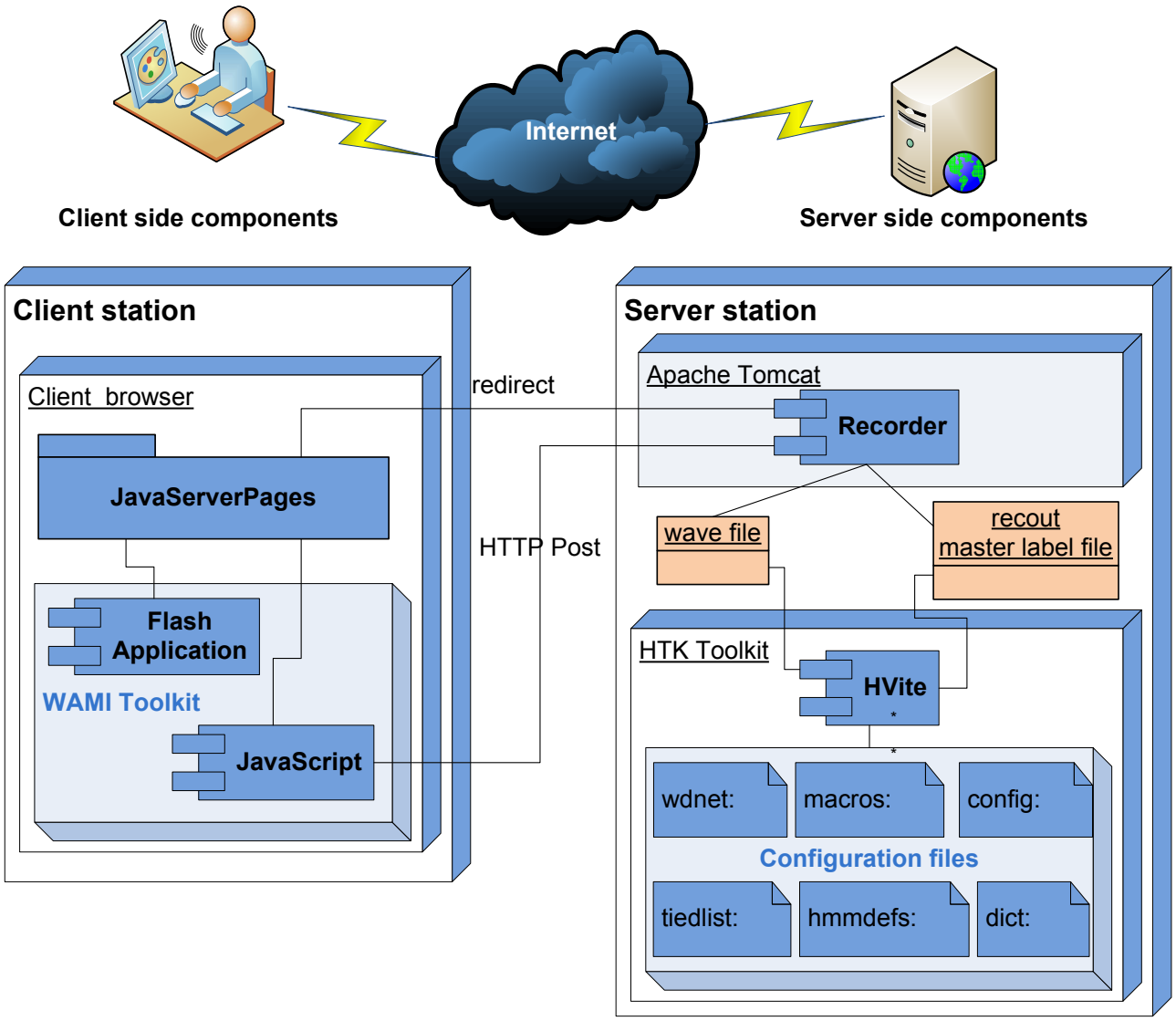

Fig. 2. – Deployment diagramm showing system architecture

The server side Web application runs on an Apache Tomcat JSP container as a recorder JavaServer Page which is waiting for HTTP Post request to download the audio stream. Once the audio stream is downloaded, it will be stored as a wave file on the server machine's file system.

Speech recognition module runs also on the server side. The recorder starts HVite HTK component [8] for speech decoding. Speech recognition module has the already trained acoustic

model and word network for task grammar as well as all the necessary configuration files prepared. It performs speech decoding and writes the recognition result in the recout.mlf master label file. The Web application takes the voice command and redirects client browser to the desired Web page.

The Database

The demo application task grammar is a simple HTK gram file using regular expressions. This is enough to describe simple navigation commands on every usual Web page. We have three types of elements in an utterance, namely the:

$command = Navighează | Deschide | Încarcă | Porneşte
$website = Sapientia | UTCluj | Autovit | Prosport | GSP | TVR | Gazeta [Sporturilor] etc.
$termination = [punct] ro | [punct] com | [punct] org | [WEB] Mail

In the above examples | stands for "logical or" expression, and [ ] denotes an optional word. In this circumstance we can have utterances with the fallowing syntax:

(SENT-START ($command $website $termination | $website $termination | $website) SENT-END)

In the syntax SENT-START denotes the start of an utterance while SENT-END denotes the end of an utterance.

We can have just the name of the website, the name of the website with the termination, a command with the website name or all three elements together: command with website and with termination. The user is free to utter his command in any of the above described syntax. We have to deal with a predefined grammar but which is very similar to natural language. It does not introduce any hard constrain for the speaker.

The recorded database contains 600 speech utterances in Romanian language with continuous speech Internet browser navigation commands. Here are some example utterances containing popular Romanian sport magazine Web site addresses as well as Technical University of Cluj-Napoca and Sapientia University's home pages:

Deschide Gazeta Sporturilor (Open Gazata Sporturilor)
Navighează Autovit.ro (Browse autovit.ro)
Încarcă Sapientia Web mail (Load Sapientia Web Mail)
Porneşte Prosport.ro (Start prosport.ro)
UTCluj Mail (UTCluj Mail)

The database was recorded in quiet office condition, using a headset connected to a laptop computer, at a sampling frequency of 16 kHz, each sample coded on 16 bit. For quick and continuous recording we have developed a bash script to wait for the recorded wave file, increase a counter each time a new file is saved by HSLab [5] and rename it automatically according to the counter value in a predefined format as S0001.wav – S0600.wav. The utterances were than cropped using Audacity application to cut out the long silence regions at the beginning and at the end of speech.

We have than created the list of train speech files (S0001.wav – S0400.wav) and the list of test speech files (S0401.wav – S0600.wav). All the transcriptions of the utterances are stored in a master label files in orthographic transcription. We have generated word level transcriptions and grapheme level transcription to.

Then we extracted the dictionary from the transcriptions and the grapheme list from the dictionary. In this demo task we have to deal with 25 words and 27 graphemes including the sil (silence) and sp (short pause between the words).

The acoustic feature vectors and trigram models

Acoustic feature vectors were constructed with HCopy using energy and 12 MFCCs (resulting 13 coefficients). In addition to the static cepstra 13 delta and 13 acceleration coefficients were added to the feature vector. Thus we have 39 features extracted for each time window of 25 ms speech signal using Hamming window and with an overlap of 15 ms between the consecutive time windows. The normalization of speech signal was made using a preemphasis coefficient having a value of 0.97.

Acoustic modeling was made with HMMs. First we have created monographeme HMMs for each grapheme used. In addition we have introduced sil (for silence portion in speech) and sp (short pause between the words) models like in [8]. We used flat start monographemes with 5 states left–right topology and no skips between the states. Each model has an initial state and a final state used for model concatenation and three emitting states to describe the beginning, the middle and the end of the acoustic realization of a grapheme.

After several successive reestimation stages, we have built the monographeme models and we have realigned the training data. Realignment and parameter estimation were made using forced alignment with HTK Toolkit HVite component [8].

Given the set of trained monographeme models we have created trigrapheme transcription for the database and built context dependent tied states trigrapheme HMMs in two steps: first we have converted monographeme transcriptions into trigrapheme transcriptions and in the second stage we have tied similar acoustic states for robust estimation of all state distributions and we have reestimated the trigrapheme HMMs parameters.

The Web Application

The Web application development involves speech signal recording on the client side through a client Web browser. The application must be browser independent. For this purpose we have used an Adobe Flash application from the WAMI Toolkit. WAMI Toolkit gives a simple way to add speech recognition capabilities to any Web page. WAMI recorder was developed just for recording audio samples throug a browser.

To create a browser independent application, WAMI recorder uses a light-weight Flash application to collect audio sequences from the client Web browser and a JavaScript for sending it to a server via a HTTP POST request. We have take into account that speech recognition task is not trivial and needs some computational power, therfore it is not suitable to run this task in the client browser. That's why we make audio file transfer to the server side to perform speech recognition on the server machine. The entire interface for sound recording and transmission can be constructed using simple HTML code, JavaScript and the preconfigured Flash swf application.

Below are presented some screenshots about the Flash application when ask to enable the microphone and start recording or play recorded audio (Fig. 2).

Fig. 3. - Microphone enable and buttons to start recording and play recorded audio [13]

The server side was developed considering the Java SE platform and JavaServer Pages technology. Server application runs on an Apache Tomcat JSP server, listens for HTTP POST requests from the clients and opens an input stream to get the audio sequence. The recorded audio is then saved to the server's file system using FileOutputStream as a raw wave sound file.

The speech recognition module runs on the server machine, the recorder JSP call the HVite to perform recognition after the speech signal is saved on the server. On the server there are prepared the configuration files for the decoder and also the trained acoustic model is available. Once recognition is performed, the JavaServer Page reads the result and redirects client browser to the desired Web page.

Discussions on Recognition Results

To test the ability of the recognizer we have performed offline and online recognition tests. Table 1 summarizes recognition results for both experiments.

Table 1.
Summary of the recognition results

	Offline testing	**Online testing**
Number of test sentences	200	200
Insertion errors	0	11
Deletion errors	0	2
Substitution errors	8	2
Total number of sentence level errors	8	15
Sentence level error rate	4.00%	7.50%
Correct sentence transcriptions	192	185
Sentence level recognition rate	96.00%	92.50%
Number of test words	808	636
Insertion errors	0	13
Deletion errors	4	3
Substitution errors	8	2
Total number of word level errors	12	18
Word error rate (WER)	1.49%	2.83%
Correct word transcriptions	796	618
Word level recognition rate	98.51%	97.17%

Offline recognition. Offline recognition tests were performed using prerecorded wave files without using the client side components. We have used 200 test files with continuous speech command utterances. We have used HVite for decoding and the recognition results are shown below in terms of sentence level recognition results and word level recognition results.

At the sentence level we achieved 96% correct transcriptions. From a total number of 200 test sentences 192 were correctly recognized. There were 8 substitution errors. No insertion errors and no deletion errors were found.

At the word level recognition we achieved 98.51% correct transcriptions. This means only 1.49% word error rate (WER) with 4 deletion errors, 8 substitution errors and no insertion errors from a number of 808 test words.

Online recognition. Online recognition test was performed using a number of 200 from live continuous speech recorded through a client browser. At the sentence level we have 185 correct transcriptions that means 92.5% recognition rate. We have counted 13 substitution errors, no deletion errors and no insertion errors.

At the word level transcription we achieved a WER of 2.83% with 18 mistakes out of 636 test words. 13 insertion errors, 3 deletion errors and 2 substitution errors occurred during this test.

Conclusions

The achieved recognition rates encourage us to perform tests on a larger database. We also want to train a speaker independent trigrapheme based acoustic model for Romanian language and to perform online tests with the new system.

Context dependent grapheme based subword units seems to be a good choice for Romanian language modeling. The recognition performances of the system strengthen this. The context dependent trigraphemes have the power to deal with Romanian language pronunciation. In the future we will try to train also quintgrapheme models to improve system performance.

This demo application shows how speech recognition can be used to design and develop multimodal interfaces. Once we have on each Web page a navigation menu, this can be easily sent to the server side to build a task grammar, similar or in many cases simpler than the one used in the demo application, and to perform speech recognition to decode the user's choice. After this process the client browser can be redirected to the desired Web page.

The recorded speech corpora with the sentence level, word level, grapheme level and trigrapheme level transcriptions are freely available on the NaviRo project website: http://users.utcluj.ro/~jdomokos/naviro/.

Acknowledgment

This paper was supported by the project "Development and support of multidisciplinary postdoctoral programmes in major technical areas of national strategy of Research - Development - Innovation" 4D-POSTDOC, contract no. POSDRU/89/1.5/S/52603, project co-funded by the European Social Fund through Sectoral Operational Programme Human Resources Development 2007-2013.

References

[1] G. Saon, J.-T. Chien, Large-Vocabulary Continuous Speech Recognition Systems. A look and some recent advances, in IEEE Signal Processing Magazine, 29/6 (2012) 18-33.

[2] N. Sebe, Multimodal interfaces: Challenges and perspectives, in Journal of Ambient Intelligence and Smart Environments, 1 (2009) 19–26.

[3] C. Burileanu, V. Popescu, A. Buzo, C.S. Petrea, D. Ghelmez-Haneş, Spontaneous Speech Recognition for Romanian in Spoken Dialogue Systems, in Proceedings of the Romanian Academy, Vol. 11, Series A, 1 (2010) 83–91.

[4] A. Stan, J. Yamagishi, S. King and M. Aylett, The Romanian Speech Synthesis (RSS) corpus: building a high quality HMM-based speech synthesis system using a high sampling rate, in Speech Communication, Vol 53, Issue 3 (2010) 442-450.

[5] D.-P. Munteanu, C.-I. Vizitiu, Robust Romanian language automatic speech recognizer based on multistyle training, in WSEAS Transactions on Computer Research, Volume 3 Issue 2, (2008), 98-109.

[6] D. Militaru, I. Gavăt, O. Dumitru, T. Zaharia, S. Segarceanu, ProtoLOGOS, System for Romanian Language Automatic Speech Recognition and Understanding (ASRU), in Proceedings of the 5th Conference on Speech Technology and Human-Computer Dialogue (SpeD), (2009), 21-32.

[7] D. Jurafsky, J. H. Martin, Speech and language processing: an introduction to natural language processing, computational linguistics and speech recognition, 2nd edition, Pearson Prentice Hall 2008.

[8] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. (A.) Liu, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, P. Woodland, The HTK Book (For HTK version 3.4), Cambridge University Engineering Department, 2006.

[9] K. Livescu, E. Fosler-Lussier, F. Metze, Subword Modeling for Automatic Speech Recognition. Past, present and emerging approaches, in IEEE Signal Processing Magazine, 29/6 (2012) 44-57.

[10] A. Stan, P. Bell, S. King, A Grapheme-based Method for Automatic Alignment of Speech and Text Data, in Proceedings of the IEEE Workshop on Spoken Language Technology, 2012.

[11] I. McGraw, C. Lee, L. Hetherington, S. Seneff, J. Glass, Collecting voices from the cloud, in Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC), 2010.

[12] A. Gruenstein, I. McGraw, I. Badr, The WAMI Toolkit for Developing, Deploying, and Evaluating Web-Accessible Multimodal Interfaces, in Proceedings of the 10th International Conference on Multimodal Interfaces (ICMI), 2008.

[13] Information on the WAMI Toolkit Web site http://wami.csail.mit.edu/