2019 22nd International Conference on Control Systems and Computer Science (CSCS)

Making Pepper Understand and Respond in Romanian

Dan Tufiș
Research Institute for Artificial Intelligence “M. Drăgănescu”
Romanian Academy
Bucharest, Romania
tufis@racai.ro

Verginica Barbu Mititelu
Research Institute for Artificial Intelligence “M. Drăgănescu”
Romanian Academy
Bucharest, Romania
vergi@racai.ro

Elena Irimia
Research Institute for Artificial Intelligence “M. Drăgănescu”
Romanian Academy
Bucharest, Romania
elena@racai.ro

Maria Mitrofan
Research Institute for Artificial Intelligence “M. Drăgănescu”
Romanian Academy
Bucharest, Romania
maria@racai.ro

Radu Ion
Research Institute for Artificial Intelligence “M. Drăgănescu”
Romanian Academy
Bucharest, Romania
radu@racai.ro

George Cioroiu
Research Institute for Artificial Intelligence “M. Drăgănescu”
Romanian Academy
Bucharest, Romania
gcioroiu@racai.ro

***Abstract*—Within the larger project ROBIN, focused on the development of software and services for human-robot interaction, we present here a set of activities focused on the creation and enhancement of language resources necessary for making dialogue possible between humans and the robot Pepper. More precisely, we describe the preparatory activities for turning the robot Pepper into a dialog partner for a human by using the Romanian language. The language resources that have been created are a lexicon, a language model, and an acoustic language model. They have been enhanced by using other language resources, such as the Romanian wordnet and word embeddings extracted from the large Romanian reference corpus CoRoLa. They will ensure oral communication within some envisaged microworlds, thus they are specific to these microworlds. These resources have been carefully curated and evaluated.**

Keywords—Romanian language resources, human-robot situated dialog, microworlds, Pepper robot.

I. Introduction

The three-year project ROBIN, coordinated by University Politehnica of Bucharest, Faculty of Computer Science (http://aimas.cs.pub.ro/robin/en/), was launched in 2018. It is a user-centred project, whose aim is to develop software and services for human interaction with robots within a digital interconnected society. The ROBIN consortium includes experts in Natural Language Processing from the Research Institute for Artificial Intelligence Mihai Drăgănescu” of the Romanian Academy (RACAI) and University Politehnica of Bucharest, in robotics from the University Politehnica of Bucharest and the Institute of Mathematics of the Romanian Academy, well known specialists in the Internet of Things (IoT) technology from the University of Bucharest and Technical University of Cluj-Napoca, and experts in designing autonomous vehicles from University “Dunărea de Jos” of Galați. The project is focused on several types of robots: assistive ones - targeting users with special needs (people with some medical problems or aged people), robots for interaction with clients and software robots that can be installed on vehicles with the aim of (semi)autonomous driving. This complex project is made up of five subcomponents:

-ROBIN-Social (aimas.cs.pub.ro/robin/en/robin-social/),
-ROBIN-Car (aimas.cs.pub.ro/robin/en/robin-car/),
-ROBIN-Context (aimas.cs.pub.ro/robin/en/robin-context/),
-ROBIN-Dialog (aimas.cs.pub.ro/robin/en/robin-dialog/)
and
-ROBIN-Cloud (aimas.cs.pub.ro/robin/en/robin-cloud/).

The component projects have their own objectives and combine advanced techniques and technologies from Artificial Intelligence (AI), human-robot interaction, pervasive and Cloud computing, each of them being coordinated by different partners of the complex project.

The focus of this paper is the ROBIN-Dialog component project, whose main challenge is the language interaction between humans and the robot. This project is coordinated by RACAI. The specific objectives are the definition of several scenarios for some microworlds and the creation of the necessary Romanian language resources and processing tools for making a robot able to communicate with users in tasks defined within these microworlds. Successful communication implies either a verbal reaction to people’s verbal command or accomplishing a task that is formulated by the human user in spoken Romanian. The challenge is considerable, especially that this would be the first case of having a robot speak Romanian, as far as we are aware, although the field is evolving rapidly.

The paper is organized as follows: we present some related work about the robot we use in the project in section II and then we give an overview of our project (section III). The concept of microworld is defined, described and exemplified in section IV. The first language resource necessary for teaching the robot to participate in dialogs in Romanian, namely the lexicon, is presented in section V, while the next section contains a presentation of the language model and the processing platform. The acoustic model and its training for Romanian are described in sections VII and VIII, respectively, and are followed by conclusions.

2379-0482/19/$31.00 ©2019 IEEE
DOI 10.1109/CSCS.2019.00122

II. Related work

Created by SoftBank Robotics, the robot involved in the project ROBIN-Dialog, Pepper, was initially developed to ease the workload of a store staff and attract more customers. It was designed, from the start, to enter the B2C market that was expecting such a robot [1]. There were many experimental robots at the time (e.g. ASIMO, Baxter, COMAN, etc.) but they "did not target the same goal in terms of design and application as the Pepper robot, i.e., a robust, general-purpose, socially interactive humanoid robot" [1].

A first example of Pepper's interaction capabilities is given in [2]. This example is typical of a microworld interaction where Pepper was tasked with quizzing the people about Belgian chocolates, standing next to a chocolate shop in the Brussels airport. Pepper was able to ask the questions either using its tablet or using its speech interface. The authors conclude that "adding speech does not seem to provide significant value" because of the noisy airport environment affecting the speech recognition module. This experiment reveals the first real-world impediment in programming Pepper (or any other robot) to understand a language: speech recognition in a noisy environment. By default, Pepper does not support such an advanced ASR module and, in order to make it work, Pepper has to be upgraded with advanced hardware such as unidirectional microphones.

Garcia et al. evaluated Pepper as a social companion for the elderly people at a hospital [3]. Their goal was to develop metrics for an objective evaluation of Pepper's social skills but one of the clear-cut conclusions is that Pepper needs better adaptation to the user, especially when the user is an elderly person. In this respect, the robot "has to adapt its vocabulary and its behavior according to the user's age. It also has to change its speech velocity and its way to detect emotions (according to the speed of the user speaking)" [3]. This is an important observation for the ROBIN-Dialog project as well, given the fact that one of our envisaged microworlds concerns elderly people.

Perhaps one of the best examples of tailoring Pepper's capabilities to a microworld scenario is given by Tanaka et al. in [4]. They describe an educational application that teaches pre-school children to speak English, inserting Pepper in the learning loop. Pepper would ask the children to e.g. "Pick up a COLORED object." and then "talk" to the English teacher on its chest tablet, asking her "How to say COLOR in English?" The teacher would answer, e.g. "Red" and then both Pepper and the children would have the answer (later on, to reinforce the learned color, Pepper would suddenly ask "Give me a red object"). Another path in the "learning English" scenario involves children watching a short video on Pepper's tablet, e.g. an airplane flying. Then, the teacher-in-the-tablet would say in English and exemplify "flying" by extending her arms; Pepper would afterwards ask the children "Do you want to try it?" and extend its arms performing an indoor "flight" to the children's amusement.

All these examples rely on context-independent, unrelated tasks that can be performed by Pepper if they are very well defined (e.g. say this sentence, wait for a response, play a video, extend your arms, stop). The downside of this approach is that *the code that Pepper executes for each task component is not reusable for other tasks*. In the ROBIN-Dialog project, we aim to define a framework for a natural language understanding (NLU) pipeline, tailored to a specific microworld and scenario, that is able *to formally describe a task that Pepper executes in a microworld according to a prescribed scenario*. The language understanding pipeline is meant to be the means by which a natural language input (written or spoken) is mapped onto a sequence of actions compliant with the robot's abilities and coherent with the scenario of the respective microworld. This description implies that the NLU should be able to do simple inferences, solve language extragrammaticalities (such as references to previously mentioned objects or actions, elliptic sentences/requests) and this requires maintaining internal variables and a history of the dialog. For instance, a sequence "Raise the right arm" followed by "Now, the left one" poses no understanding problem for a human, but the robot should be explicitly told "Now, **raise** the left **arm**". From this point of view, our approach is based on a 21st century-updated version of GUS [5], a frame-driven dialog system that was designed for dialogs with mixed initiative and indirect or fragmented answers. The update includes using our state-of-the-art NLP processing flow TEPROLIN and using Machine Learning approaches to mine for the user's intent from an utterance, in a given scenario and microworld.

III. Overview of the ROBIN-Dialog project

A speech dialog with a robot assumes at least four components, as suggested by the diagram in Figure 1: a module for automatic speech recognition (ASR), a natural language processing unit (NLP) with two subcomponents - one for analysis (NLU) and one for generation (NLG), a dialog manager (DM) and automatic speech synthesis from text (TTS). The ASR component is a trained module, language dependent, for turning the stream of vocal signals into a text. Our ASR system is composed of:

- an acoustic model (that represents the relationship between acoustic data and phonemes),
- a lexicon enhanced with phonetic transcriptions (that represents the relationship between strings of phonemes and words; those are the words that the system can recognize) and
- a language model (representing a distribution probability of words into sequences).

The lexicon and the language model are dependent on the context (microworld) the robot should be operating in. The NLP module is also trained for dealing with sequences of words as produced by the ASR component. It analyses the input text linguistically, that is, sentences and then tokens are identified, words are morphologically analysed and lemmatized, and generates an actionable representation making sense within the current microworld and the ongoing dialog. The dialog manager is practically a script matching the input representation with the collection of action representations that the robot can and may perform, as well as the representation of the appropriate response to be conveyed to the human partner. If the input requests an action to be performed by the robot, the Dialog-Manager (DM) sends the request to the actional platform of the robot (mapping and navigation, animation and moving around, etc.). This module is currently emulated by means of an API to the wit.ai platform (https://wit.ai/). The dialog manager should be also in charge of maintaining a dialog history so as to be able to cope with the references to previous actions (anaphoras and ellipses). The language processing unit has to generate a fully instantiated sentence to the wit.ai platform, that is any input containing an ellipsis should be completed with the missing part, any pronominal anaphora should be replaced by the lexical items identifying the referred entities.

Currently, this requirement is not yet implemented. The NLP module will receive the robot's encoded reply (acknowledgement, question answer, clarification request if necessary) and turn it into a sequence of words (a sentence) meaningful for the human partner. This sequence of words represents the input for the TTS module, which will turn it into a spoken reply.

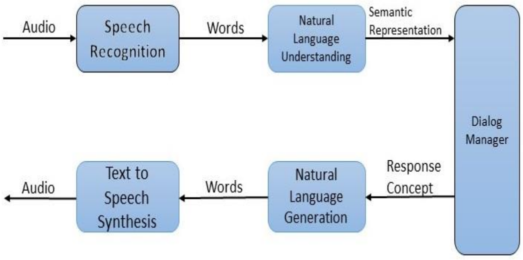

Fig.1. A standard architecture for the language processing flow.

Out of the mentioned modules, ASR, NLP and TTS need training for the language to be used in the man-robot interaction, in our case Romanian. In order to train a language processing module (be it spoken or written), appropriate language resources have to be constructed. The DM needs to be adapted to the envisaged microworlds and robot's capabilities.

Our goal in this paper is to present partial results from the project ROBIN-Dialog: creating the data models and the appropriate processing tools, indispensable for making Pepper able to engage in a meaningful dialog in Romanian. As far as we are aware, this would be the first experiment of teaching a robot (like Pepper) to interact with humans in the Romanian language and this represents the main novelty of the project.

IV. MICROWORLDS

A microworld is to be understood as representing a limited space, the objects, people, and the robot in it, as well as verbal exchanges between the people and the robot on topics connected to these elements that define the microworld. Thus, the human-robot dialogs are situated ones (microworlds and scenarios based) and this feature differentiates them from the dialogs carried on chatbots.

One example is the microworld of the notebooks department in an electronics store. This microworld is made up of the physical space occupied by this department, by the notebooks that are commercialized by that store, their characteristics on the basis of which customers decide what products they buy, their availability, the provisional date for their becoming available, the robot and the customers who interact with the robot for finding the notebook they want to purchase or for finding the right configuration for their needs. A microworld is also anchored in time. Within it, one can imagine various dialogues, but they must be specific to this space and activity.

These components of the microworld require the following on the robot's part: awareness of the respective space topology, the ability to move in that space, ability to understand human language and to react to it either by giving an oral response or by executing an action within the space limits, etc.

Three microworlds were imagined for the robot Pepper in this project. They all imply the assistive role of the robot: it can assist (i) elderly people at home; (ii) visitors in a research center; (iii) customers in the notebooks department of an electronics store. In what follows we offer a brief description of the microworlds.

(i) When assisting elderly people at home, the robot needs to know about the person's possible health problems and due to visual observation to be able to react in a helpful way (an elderly person falling should be immediately reported, alerting, for instance, the ambulance). The assisted people may need being reminded when to take some medication, what medication to take and/or even where it is and the robot is supposed to be ready to help.

(ii) Within a research center, the robot will need to meet people, recognize them, show them the way around the center (always making sure that the guest follows it). The robot should also be able to communicate with the people in the center by delivering a verbal message from one person to another, without mistaking the people. Moreover, the robot must be aware all the time of the activity it is involved in and this is proven by making it be able to answer questions such as "What are you doing (now)?".

(iii) As a shop assistant, the robot must know the products the department commercializes, their availability, their characteristics, as well as the types of jobs they are adequate for (e.g. notebooks for gaming, for design, programming, etc.)

In order to teach the robot communicate in Romanian in the scope of the described microworlds, we developed screenplays, that, like film scenarios, involve action and dialog. The action is quite minimal, whereas for the dialog the actors are identified and their possible lines are created. The actors are humans and the robot. The dialog is adequate to the actions the robot needs to perform. An exhaustive list of possible actions was imagined and the lines for both humans and the robot were written.

In order to play its assistive role, the robot must be able to cope with different ways of saying the same thing. As a consequence, the original form of the possible verbal interactions between a human and the robot must be conceived of at a conceptual, language independent level. This level is further lexicalized: pairs of verbal interactions are given as many lexicalized forms as possible, so that to ensure the robot's widest exposure to a large vocabulary and numerous syntactic structures. For example, the following ways of asking the robot if it knows the person called "Romeo" were identified in Romanian: "Îl știi pe Romeo?" / "Știi cine e/este Romeo?" / "Îl cunoști pe Romeo?".

The enhancement of the vocabulary is further ensured by means of word embeddings and expansion with the help of wordnet semantic relations, as described below.

V. THE LEXICON

Based on the screenplays/dialogs created for the microworlds of interest, we developed a lexicon that will be essential to the speech processing applications designed for Pepper. The lexicon is a collection of words and information about their dictionary form (lemma), morpho-syntactic label codifying their morpho-syntactic description, the stress (marked by an apostrophe), the syllabification (marked by a dot) and the phonetic SAMPA transcription. An entry in the lexicon is a line containing all this information separated by tabs, as in Table I.

TABLE I. STRUCTURE AND CONTENT OF A LEXICON ENTRY

word	lemma	MSD	syllables	accent	ph_transcript
afișând	afișa	Vmg	a.fi.șând	afiș'ând	a f i sh 1 n d

The screenplays were processed – lemmatized and part-of-speech (POS) tagged – (using the TTL module [6], part of the TEPROLIN [7] processing flow http://89.38.230.23/teprolin/ developed at RACAI), and a preliminary lexicon was generated, with only the first three positions shown in Table I (word, lemma and morpho-syntactic description - MSD) being completed. Since we wanted a dialog system capable of dealing with the human tendency to opt for different ways of expressing the same semantic content, we aimed for a lexicon as comprehensive as possible in the scope of the imagined microworlds. Therefore, the initial set of words was extended in two stages:

Stage1. For each lemma in this preliminary lexicon (containing 257 lemmas of content words), we generated 10 other lemmas that have a similar distributional semantics, using word-embeddings generated from the Romanian reference corpus CoRoLa [8], [9]. This corpus is large, counting almost 1 billion tokens, and its texts belong to over seventy domains, which makes the lexical diversity wide: besides the words frequent in the general language and in all domains of activity, the corpus also contains words specific to all these domains. The new lemmas added to the lexicon using CoRoLa were manually validated so as to ensure their semantic closeness to the microworlds of interest;

Stage 2. After that, we used the Romanian WordNet (RoWN) as another resource for mining new vocabulary, in order to further extend the existing lexicon of the microworlds. RoWN [10] is an electronic "lexical database", containing only open class words (nouns, verbs, adjectives and adverbs). Sets of synonymous terms, or synsets, constitute its basic organization elements. The current version integrates 60,000 synsets organized in separate hierarchies for nouns, verbs, adjectives, and adverbs. Several types of relations between synsets are recorded, including hyponymy (specific-generic) and meronymy (part-whole) among nouns. In addition, each synset has a definition (or gloss) that explains the meaning of the literals in the synset.

For all the lemmas obtained in Stage 1 we extracted from RoWN the words that are in the same synsets with the respective lemmas and the direct hypernyms of those lemmas. The motivation behind this is offered by the rephrasing strategies in any language, which make use of synonyms or more general words (hypernyms). The limitation to direct hypernyms was made to maintain the microworlds' specificity (on upper levels we could get to too general concepts) and also to avoid aggregating words emanating from distinct senses of the words than the ones intended (because of the polysemy phenomenon). All the generated candidates were manually filtered very carefully and all the lemmas which did not have a similar meaning to the original lemma were removed. The benefits of imposing this specificity to microworlds' lexicon translates into faster lookup time at decoding (since there are fewer words) and better WER (Word Error Rate).

After extending the list of lemmas associated to the microworlds to 1,741 entries, we enriched the lexicon with all their morphological variants by looking-up in an extensive Romanian lexicon developed at RACAI (comprising over 1 million hand-validated entries), as well as with functional words. These forms are necessary for ensuring syntactic variation in the robot's linguistic knowledge. Thus, ROBIN-Dialog word-form lexicon reached 34,199 entries. In a further processing step, we completed the lexicon with the information about stress, syllabification and phonetic translation using other modules of the processing flow TEPROLIN. All the annotations were validated, either by looking-up in existing corrected resources, or by manual checkout. This lexicon is public and may be freely downloaded from the address http://www.racai.ro/p/robin/rapoarte/lexiconrobin.rar

VI. THE LANGUAGE MODEL AND THE NLP PLATFORM

The tri-gram language model is built using the SRILM utility, based on the transcriptions of the available recordings (most extracted from the oral part of CoRoLa corpus). The texts are processed as described in the previous section by means of the processing flow TEPROLIN accessible on the dashboard at http://89.38.230.23/teprolin/. The TEPROLIN processing flow allows for various output formats (JSON, CoNLL, XML or dependency-Tree). Text processing is meant to serve both ASR and TTS processes. For instance, given the input text message:
"*Nu știu exact ce fel de laptop vreau să cumpăr.*" (en. "*I don't know exactly what type of laptop I want to buy.*")
the output of the TEPROLIN (in CoNLL format) showing the word form, its lemma, its MSD tag, the simplified tag (CTAG), the phonetic transcription and the accent position and the syllabification is:

```
Nu       nu        Qz          QZ       n.u              nu
știu     ști       Vmip1s      V1       S.t.i.w          știu
exact    exact     Rgp         R        e.gz.a.k.t       e.x'act
ce       ce        Dw3--r---e  RELR     tS.e             ce
fel      fel       Ncms-n      NSN      f.e.l            fel
de       de        Spsa        S        d.e              de
laptop   laptop    Ncms-n      NSN      l e p.t.o.p      l'e.top
vreau    vrea      Vmip1s      V1       v.r.e_X.a.w
         vreau
să       să        Qs          QS       s.@              să
cumpăr   cumpăra   Vmsp1s      V1       k.u.m.p.@.r      c'um.păr
.        .         PERIOD      PERIOD
```

VII. THE ACOUSTIC MODEL

We have investigated the performance of various tools for training an ASR system and got to the conclusion that the most useful ones are:

- CMUSphinx - it contains a lot of open-source tools that are useful for the development of speech recognition applications. It is written in C and implements a continuous detection, speaker independent, using HMM (Hidden Markov Model) for the acoustic model and statistical n-gram models for language model. It is widely used on low resource systems, having even an embedded version called Pocketsphinx.
- Kaldi - while it is similar with CMUSphinx, it is written in C++ and implements more algorithms, including neural networks. It can be trained both for speaker dependent and independent models, and it can decode both online and offline.

We experimented with both packages and on our training data Kaldi provided better results (lower WER), so we decided to opt for this solution.

However, regardless of the tool, there are some files that are required in order to obtain an ASR system. Those are:

- Audio files and their transcript;
- Language model;
- Phonetic dictionary;
- A list of phonemes.

For training the language model (LM) of our ASR module, based on Kaldi, we used the SRI-LM toolkit [11] and trained 3- and 4-gram models with and without pruning. Pruned models are smaller and they avoid the overfitting of the training data by removing n-grams that do not contribute to a significant increase of the model's perplexity on the training set. We set the value to 10^{-8}, a value that was recommended by the toolkit. Another parameter tuning with respect to the LM training was the choice of the discount method: we used Chen and Goodman's modified Kneser-Ney discounting for n-grams of order n [12], experimentally proven to be the most effective discount method in the cited paper. Finally, all of our models were interpolated, a fact that is mentioned to improve the performance of the LM with respect to the chosen discount method.

The training text contained approx. 592 million words, chosen from the CoRoLa corpus, out of the texts that contained Romanian diacritics. Sentence boundaries were added explicitly, for each extracted sentence.

The phonetic dictionary was automatically extracted from the unigrams of all pruned/unpruned language models. Each word of the dictionary was phonetically transcribed with the TEPROLIN platform, which uses the SAMPA phonemes for Romanian[1].

We tried to evaluate the performances of the ASR module, but the preliminary results are rather modest: 25% WER (Word Error Rate) and 75% SER (Sentence Error Rate). The evaluation was done mainly on radio records of free speech. However, when we evaluated noiseless speech records, the recognition was almost perfect (two slightly misrecognized words out of 49). As the proper vocal interaction with the robot is very likely to be affected by noise, we think that the current ASR version needs serious improvements.

The TTS part of ROBIN-DIALOG will exploit our previous results [13] obtained for embedded devices, results which are supposed to be further improved within another running project (ReTeRom, http://www.racai.ro/p/reterom/) aiming at producing a large speech corpus and state of the art general purpose speech processing tools, including TTS systems.

However, we finalized a first version of the TTS module for ROBIN, based on the TTS-Cube due to Tiberiu Boroş[2]. The system may be tested with both male and synthesized voices. The quality is quite good, although we still have to improve the response time. The interested user may test an on-line demo version of the current version of the TTS module at: http://www.racai.ro/tools/speech-tools/tts-robin/

The processing control is shown in Figure 2 below. First, the text goes through a phonetic embedding step. The output of this step will be an array of integers, each number representing the ID of a phone. Note that explicit phonetic transcription it's not needed if one considers each letter to be a phone one its own, hence TTS-Cube can realise end-to-end synthesis. The phonetic encoder model depends on the language of the text.

Next, the embedding array and voice info are being processed by an encoder and transformed into mel-log spectrogram. The encoder model also depends on the language. This encoder is similar to those proposed in Tacotron[14] and Char2Wav[15], but has a lightweight architecture with just a two-layer bidirectional LSTM encoder and a two-layer LSTM decoder.

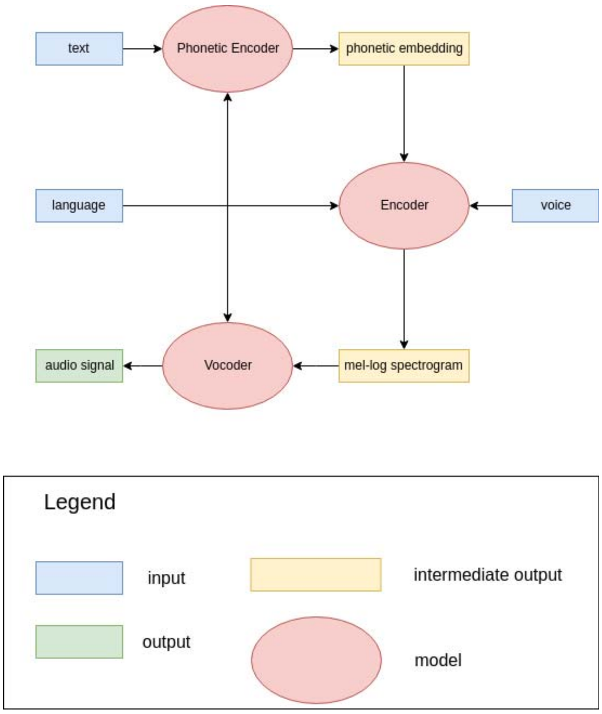

Fig.2. Processing flow in the TTS-Cube

The mel-log spectrogram obtained from the previous step is passed to a vocoder that finally produces the audio signal. This model also depends on the language of the text. In order to achieve real-time synthesis, the vocoder model is based on ClariNet[16].

VIII. TRAINING AN ACOUSTIC MODEL FOR ROMANIAN LANGUAGE

The database used for training the acoustic model contains over 100 hours of quality speech along with their transcribed text. They are the oral component of the CoRoLa corpus. Based on all the sentences from the database a statistical language model was created using tri-gram distribution. Also, every word was added in the training vocabulary with its phonetic transcription, resulting in a phonetic dictionary of about 46,000 entries. Since the Romanian language is composed of 31 phonemes, we believe the 46,000 words in our database are enough to cover any combination of 2 consecutive phonemes, thus the acoustic model will be able to recognise all the words in the ROBIN-Dialog lexicon. This size will certainly be enlarged as is makes the aim of the running ReTeRom project.

Since the original audio files were collected from various sources, they had to be brought to a unique form. As such, every audio file in the database was changed to be mono and have a sampling frequency of 16 KHz. The files were also subjected to a VAD (voice activity detection) operation in order to remove the silence from the beginning or end.

[1] https://www.phon.ucl.ac.uk/home/sampa/rom-uni.htm

[2] https://github.com/tiberiu44/TTS-Cube

The resulting acoustic model is based on the hybrid DNN-HMM model, the architecture of which is shown in Figure 3.

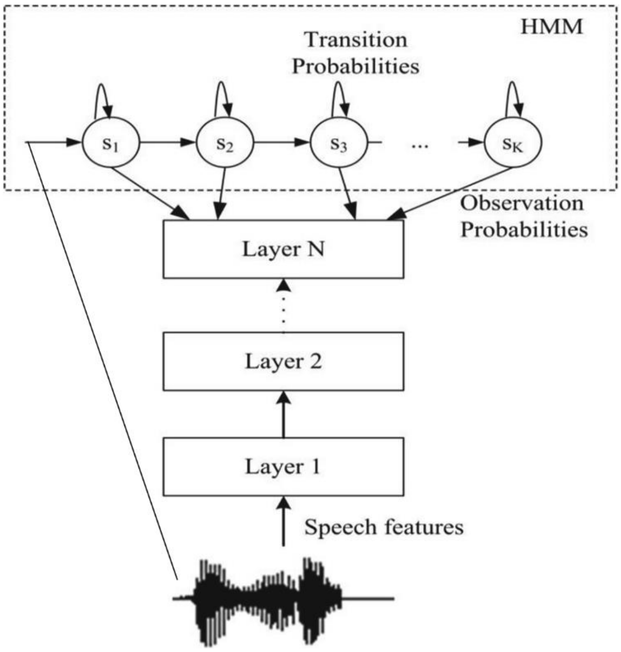

Fig.3. Hybrid model architecture.

Obtaining such a model requires the following steps to be made on training data:

1. Compute MFCC (Mel-frequency cepstral coefficients) [17]
2. Compute Delta features
3. Compute Delta-Delta features [18]
4. Applying LDA (Linear Discriminative Analysis) to reduce dimension
5. Applying MLLT (Maximum Likelihood Linear Transform) for a better decorrelation
6. Applying fMLLR (feature space Maximum Likelihood Linear Regression) to normalize multi-speaker variability [19]

Those features extracted from the speech signal will be the input for the neural network, while the output will be the observation sequence of the HMM model.

CONCLUSIONS AND FURTHER WORK

The robots are nowadays given routine tasks that involve a lot of repetitive actions that are well delineated. Pepper is a humanoid robot with assistive facilities. That is why we have opted for it in the scenarios we have imagined in this project. We described here the results obtained in the first year of the project ROBIN-Dialog. The emphasis was placed on developing the basic language resources and speech models (language and acoustic) necessary for making the robot Pepper learn to participate in verbal communication with users speaking Romanian. The NLP processing chain is operational at the moment and the next phase will be devoted to integrating a core module for Dialog Management, formalizing the microworlds as well as further improving language resources and models.

ACKNOWLEDGMENT

This work was supported by a grant of the Romanian National Authority for Scientific Research and Innovation, CNCS – UEFISCDI, project number PN-III 72PCCDI/2018, ROBIN – “Roboții și Societatea: Sisteme Cognitive pentru Roboți Personali și Vehicule Autonome”.

REFERENCES

[1] A. K. Pandey, R. Gelin, “A Mass-Produced Sociable Humanoid Robot: Pepper: The First Machine of its Kind.” IEEE Robotics & Automation Magazine, pp. 40–48, September 2018.

[2] L. De Gauquier, H.-L. Cao, P. B. Esteban, A. De Beir, S. van de Sanden, K. Willems, M. Brengman, B. Vanderborght, “Humanoid Robot Pepper at a Belgian Chocolate Shop”, HRI ’18 Companion: 2018 ACM/IEEE International Conference on Human-Robot Interaction Companion, March 2018.

[3] M. Garcia, L. Béchade, G. Dubuisson-Duplessis, G. Pittaro, L. Devillers, “Towards metrics of Evaluation of Pepper robot as a Social Companion for Elderly People”, Proceedings of the 8th International Workshop on Spoken Dialog Systems, IWSDS 2017.

[4] F. Tanaka, K. Isshiki, F. Takahashi, M. Uekusa, R. Sei, K. Hayashi, “Pepper Learns Together with Children: Development of an Educational Application”, Proceedings of the 2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids), November 2015.

[5] D. G. Bobrow, R. M. Kaplan, M. Kay, D. A. Norman, H. Thompson, T. Winograd, “GUS, A Frame-Driven Dialog System,” Artificial Intelligence 8, pp.155–173, 1977.

[6] R. Ion, “Word Sense Disambiguation Methods Applied to English and Romanian” (in Romanian), PhD Thesis Romanian Academy, Bucharest, 2007.

[7] R. Ion, “TEPROLIN: An Extensible, Online Text Preprocessing Platform for Romanian”, Proceedings of the 13th International Conference “Linguistic Resources and Tools for Processing the Romanian Language”, November 2018.

[8] V. Barbu Mititelu, D. Tufiș, E. Irimia, “The Reference Corpus of the Contemporary Romanian Language (CoRoLa)”, Proceedings of LREC 2018, pp. 1178-1185.

[9] V. Păiș, Dan Tufiș, “Computing distributed representations of words using the CoRoLa corpus”, Proceedings of the Romanian Academy, series A, pp. 403-410, 2018.

[10] D. Tufiș, V. Barbu Mititelu, “The Lexical Ontology for Romanian”, in Language Production, Cognition, and the Lexicon. Text, Speech and Language Technology, vol 48, N. Gala, R. Rapp, G. Bel-Enguix, Eds. Springer, 2014, pp 491-504.

[11] A. Stolcke, “SRILM – An Extensible Language Modeling Toolkit”,. Proc. Intl. Conf. on Spoken Language Processing, vol. 2, pp. 901-904, Denver.

[12] S. Chen, J. Goodman, “An empirical study of smoothing techniques for language modeling”, Computer Speech and Language 13, pp. 359—394, 1999

[13] T. Boros, S. D. Dumitrescu, “Robust deep-learning models for text-to-speech synthesis support on embedded devices”, Proceedings of the 7th International Conference on Management of computational and collective intElligence in Digital EcoSystems, pp. 98-102, October 2015.

[14] Wang et al., ”Tacotron: Towards End-to-End Speech Synthesis”, arXiv:1703.10135.

[15] Jose Sotelo et al., “Char2Wav: End-to-End speech synthesis”, ICLR 2017.

[16] Wei Ping, Kainan Peng, Jitong Chen, “ClariNet: Parallel Wave Generation in End-to-End Text-to-Speech”, arXiv:1807.07281.

[17] Md. Sahidullah, G. Saha, "Design, analysis and experimental evaluation of block based transformation in MFCC computation for speaker recognition", Speech Communication 54 (4), pp. 543–565, 2011.

[18] K. Kumar, C. Kim, R. M. Stern, “Delta-spectral cepstral coefficients for robust speech recognition”, Proceedings of ICASSP, 2011, pp.4784–4787.

[19] S. P. Rath, D. Povey, K. Veselý, J. Černocký, "Improved feature processing for deep neural networks", INTERSPEECH-2013, pp. 109-113.