The RACAI Text-to-Speech Synthesis System

Tiberiu Boroș, Radu Ion, Ștefan Daniel Dumitrescu

Research Institute for Artificial Intelligence “Mihai Drăgănescu”, Romanian Academy (RACAI)

tibi@racai.ro, radu@racai.ro, sdumitrescu@racai.ro

Abstract

This paper describes the RACAI Text-to-Speech (TTS) entry for the Blizzard Challenge 2013. The development of the RACAI TTS started during the Metanet4U project and the system is currently part of the METASHARE platform. This paper describes the work carried out for preparing the RACAI entry during the Blizzard Challenge 2013 and provides a detailed description of our system and future development directions.

Index Terms: speech synthesis, unit selection, concatenative

1. Introduction

Text-to-speech (TTS) synthesis is a complex process that addresses the task of converting *arbitrary* text into voice. The random text requirement is what actually complicates the process and in order to produce the desired results, a TTS synthesis system is required to chain (1) the difficult task of processing the input text into an intermediary coding that can be further exploited by the speech synthesizer, with (2) the challenging signal processing steps involved in the synthesis process itself. Thus, the overall quality of the synthesized voice is highly dependent on having both (1) a strong natural language processing (NLP) framework and (2) a voice synthesizer that can cope with various inconsistencies present in speech databases and can adapt its working set to the prosodic requirements indicated by the NLP framework. The NLP processing methods and techniques that are currently used in TTS synthesis offer accurate results for English, with most of them being easily adaptable to other languages. There are however particular languages which have special requirements when it comes to performing standard TTS text processing steps such as part-of-speech (POS) tagging, letter-to-sound (LTS) conversion or syllabification. The challenging languages usually share the common attributes of being highly inflectional and morphologically rich, which makes their processing complicated mostly because of the data-sparseness effect. As we will later show, the issue of highly inflectional languages is thoroughly addressed by our NLP framework and every sub-system of the text processing component is designed to allow easy tweaking of the feature sets it uses, which also directly makes the system adaptable to other languages.

The RACAI TTS is a very young system in comparison with most other TTSs present in the Blizzard Challenge and, naturally, some components still require work. However, we invested a lot of effort in the development of every individual sub-system and we focused on delivering state-of-the-art results and high flexibility, keeping in mind the more and more rising demand for multilingual systems. The development of RACAI TTS started during the METANET4U project and the entire system is now part of the META-SHARE platform[1]. Initially the system was designed as a set of standalone Natural Language Processing (NLP) Tools aimed at enabling text-to-speech (TTS) synthesis for less-resourced languages. A good example is the case of Romanian, a language which poses a lot of challenges for TTS synthesis mainly because of its rich morphology and its reduced support in terms of freely available resources. Initially all the tools were standalone, but their design allowed their integration into a single package and by adding a unit selection speech synthesis module based on the Pitch Synchronous Overlap-Add (PSOLA) algorithm we were able to create a fully independent text-to-speech synthesis system that we refer to as RACAI TTS.

A considerable progress has been made since the start of the project, but our system still requires development and although considered premature, our participation in the Blizzard Challenge 2013 has enabled us to locate and fix a number of faults in our system and has also helped us to understand what are the tasks we should focus on, in order to improve the performance of the TTS.

Most of the development work was carried out around the NLP framework and we can safely say that this component has reached its maturity, being able to provide accurate results with support to easily adapt the system to other languages by having full control over each module’s individual feature sets used during training. The weakest part of the system is probably the voice synthesis component which was only recently added without being fully tested and finished. However, its design is also intended to provide a high level of external control, allowing the user to tweak various parameters in the scoring functions with an option to apply changes without restarting the system (see section 2.3). This nice behavior enables to quickly observe how different parameter sets influence the result of the system, which is very useful in the absence of a more principled way of determining their values.

This paper describes the work carried out for preparing the RACAI entry during the Blizzard Challenge 2013 and provides a detailed description of our system and the future development directions.

2. System overview

In this section we will address the architecture of the system (Figure 1) and we will briefly describe the methods and techniques that are implemented within the RACAI TTS. The system is designed using the classical two-fold architecture, being composed of the NLP module responsible for text pre-processing and the DSP module responsible for converting the output of the NLP component into speech.

There are two pre-requisites for building new voices:

1) The NLP sub-modules must be trained using corpora specific to the target language/domain in order to assure the symbolic pre-processing required by TTS synthesis: tagging, syllabification, grapheme-to-phoneme conversion etc.;

[1] http://ws.racai.ro:9191

2) The speech database must be already segmented at phoneme level.

Once the pre-requisites are met, adding a new voice is straight forward: the result of the NLP framework is aligned with the segmented speech database and the prosody prediction module is trained to model the pitch and duration parameters based on the output of the NLP component. To simplify the process the extraction of the required parameters are directly performed by the system, without requiring any external tools.

2.1. Natural language processing (NLP) component

Careful and accurate processing of the input text is a requirement in the design and implementation of any TTS system. Also, the rising demand for multilingual TTS systems and the fact that different languages have different underlying rules for the phonetically oriented tasks makes it generally better to design data-driven modules that allow changes in their feature sets without requiring any additional coding.

We invested a lot of effort in providing compatibility with morphologically rich languages and we succeeded in achieving state-of-the art results in all our components. However, one of the main drawbacks of our approach is that we were unable to provide a better method for prosody prediction and currently, our system as well as many others, relies only on surface clues extracted from the local context. However, we will continuously work on this problem since it is one of the most important features that a high performance NLP framework for TTS synthesis must provide.

Most of the core components of the NLP framework have already been presented in our previous work, thus in what follows we will only provide a brief overview of the issues they are intended to solve and the methods they are based on.

POS tagging is a technique used in many NLP applications such as Information Retrieval, Machine Translation, Word-Sense Disambiguation, Parsing and it also plays an important role in TTS synthesis for tasks such as homograph disambiguation and prosody prediction. Most POS taggers commonly employ Hidden Markov Models (HMMs) but there are many other approaches based on popular classifiers such as Maximum Entropy [1], [2], Bayesian Networks [3], Neural Networks [4], Conditional Random Fields (CRF) [5], etc. Although these are all well-established POS tagging methods that have been successfully applied to English and other languages, their usage on morphologically rich languages is usually troublesome mostly because of the data-sparseness effect. For example, the Romanian language uses a number of ~1200 tags that can be reduced to 614 by exploiting the language specific syncretism; Czech uses an even larger number of about 2000-3000 tags. Different methodologies have been proposed for reducing this unwanted effect (e.g. Tiered Tagging [9]) and they attempt to overcome the lack of statistical evidence by performing tagging in multiple passes and adding layers of information. The RACAI TTS uses a Neural Network MSD Tagger, which was introduced in [7]. The idea behind this method is to generate an encoding for every tag inside the tagset, in which every possible morphological attribute value receives a unique ID. Every tag is converted into a real-valued vector based on the IDs of its morphological attributes. A neural network is than trained to learn local agreements between attribute values (e.g. gender or case) based on a window of 5 words/tags (2 previously assigned tags and the following 3 probable tags computed using Maximum Likelihood Estimates (MLE) for each individual attribute value).

This tagging method performed very well on Romanian (98% accuracy) and other highly inflectional languages. The main benefits of this technique are that (1) by controlling the network topology one can easily determine a trade-off between speed and accuracy and (2) it allows attribute masking without having to redesign the tagset, thus enabling the system to be retrained for various NLP tasks which do not require the prediction of all morphological attributes.

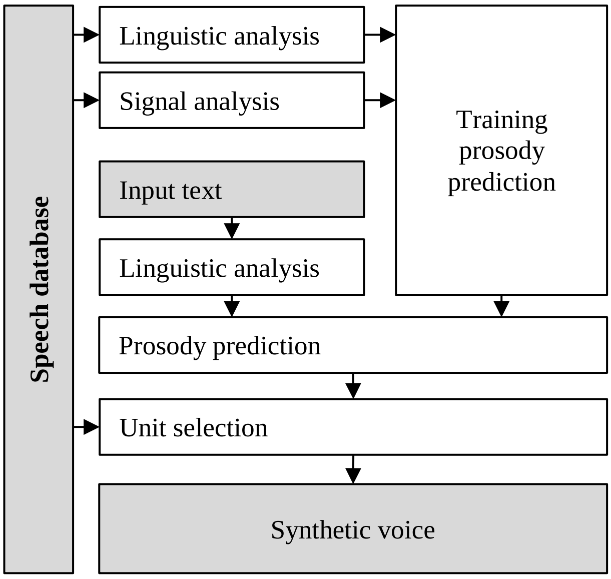

Figure 1 - *RACAI TTS architecture*

The other typical set of processing steps involved in speech synthesis is performed using an unified Margin Infused Relaxed Algorithm (MIRA) [8] framework in which all subtasks were cast as sequence labeling problems. The framework was thoroughly presented and tested in [9]. In our approach we used the numbered onset-nucleus-coda (ONC) method proposed in [10], a similar strategy as the one introduced in [11] for grapheme-to-phoneme (G2P) conversion and other custom tagging strategies for the other related tasks.

2.2. Speech synthesis component

The speech synthesis component is fairly simple. It uses a Viterbi algorithm for selecting the optimal speech units and the speech synthesis is carried out using PSOLA.

The Pitch marking is performed using a Yin Pitch estimator [12] and the system uses Mel-Frequency Cepstral Coefficients (MFCC) for the spectral representation of the speech signal.

The Viterbi algorithm uses a weighted function (equation 1), which embeds the target (equation 2) and concatenation costs (equation 3). The concatenation cost is measured using a linear function of the spectral discontinuity combined with an exponential function of the F0 frequency mismatch measured as an average over a larger number of signal frames, because our informal listening tests showed that artifacts created by joining units with incompatible pitch values are more likely to be noticeable.

$$S = \alpha T_{cost} + \beta C_{cost} \quad (1)$$

$$T_{cost} = |t_f - f| w_f + |t_d - d| w_d \quad (2)$$

$$C_{cost} = \sum_i \Delta MFCC_i + (\Delta F_0)^2 \quad (3)$$

where

S	The unit cost
T_{cost}	The target cost
C_{cost}	The concatenation cost
t_f	Target F0
f	F0 value of the unit measured as an average of 6 frames
t_d	Target duration
d	Actual duration
$\Delta MFCC_i$	Difference of the i-th component of between the MFCC feature vectors of two adjoined units
ΔF_0	Difference between the average F0 of two adjoined units measured over a 6 frames
α, β, w_f, w_d	Externally defined weights

To avoid the unwanted effect of DSP processing of the speech signal, a decision threshold is used to select if the units will be modified by the PSOLA algorithm to match the desired values for pitch and duration. If the difference between the target pitch and duration and the actual values for these parameters are smaller than that threshold, the units are left unchanged.

2.3. Technical specifications

In order to assure the system's portability to various platforms we chose Java as the development language, ensuring that the RACAI TTS does not have any external dependencies at runtime.

Currently the speech synthesizer only implements the concatenative unit-selection method, but future development plans include adding support for statistical parametric speech synthesis.

The training procedure and the possibility to tweak the system for various tasks are both defining features for any TTS system, regarding which we can highlight the following capabilities of the RACAI platform:

- It allows full control over the feature sets used by each individual module involved in text processing;
- It offers good support for highly inflectional languages;
- Every parameter concerning the unit selection process can be controlled directly at runtime, without requiring a system re-initialization, which makes it very easy to test how these parameters influence the outcome;
- Creating new voices is a straightforward process and it only requires a time aligned speech corpus for which one can employ the services of the Hidden Markov Model Toolkit (HTK) [13].

3. Participation in the Blizzard Challenge

The participation in the Blizzard Challenge 2013 was a last minute decision. In the first two weeks of the competition our system was still under development and we did not plan to attend this year. However, when the first tests on Romanian were completed successfully and after careful consideration we decided that participating in this challenge will enable us to test how our TTS compares to other state-of-the art systems.

This was a beneficial decision which helped us further improve our system and easily pinpoint weak spots in our framework. Unfortunately we did not have enough time to prepare the data before entering the competition, no manual editing was performed on the prompts and no fine tuning on the automatically determined boundaries was possible. Also, we were unable to enter any other tracks except EH2, although participating in all tracks would have created a more realistic view on the current state of the system.

3.1. Blizzard Challenge preparation

In order to thoroughly describe our entry in the competition we will start by introducing the external resources used in the English adaptation of the models.

The POS tagger was trained using the morpho-syntactic-descriptions (MSD) tagset which is fully described in the MULTEXT EAST specifications [14]. For training our models ee used Orwell's "1984" novel, a MSD tagged corpus with translations available for multiple languages.

The G2P model was trained using the CMUDict [15] lexicon from which we manually edited a few entries and we filtered out all non-English words using Princeton Word Net (PWN) [16].

The syllabification model was trained using Websters's Pocket Dictionary [17] which was automatically pre-processed according to the numbered ONC procedure.

The biggest challenge in this year's competition was the speech corpora preparation, which according to the organizers contained a number of imperfect prompts that created problems in the automatic segmentation process. Unfortunately we did not have enough time to manually check and correct the corpus and, instead, we used a simple statistical method, which, based on the duration of individual phonemes and their spectral characteristics determined by HTK, attempted to remove incorrect segments. However, the process was not precise and we were forced to use a very low rejection threshold which caused us to waste ~30% of the corpus. Also, some errors still made it through this filtering process and though we detected them while we were synthesizing the sentences for the EH2 track, we did not correct these errors because it would have been unfair for the evaluation process.

HTK was the only external tool used in our voice creation process upon which we depended to:

- Align speech spans to phonemes for each utterance of the corpus;
- Insert short pauses ('sp') in the utterances that will further refine speech to phoneme alignment;
- Filter out utterances for which the speech to phoneme aligner could not find a probable-enough alignment, mainly due to the fact that the utterance and its

prompt were slightly different. Some examples are given in Table 1;

Table 1 - *Example of erroneous prompts and recordings;*

Problem type	**Prompt IDs**
Prompt contained more words than the actual recording	CA-BB-09-15, CA-BB-20-10, CA-MP1-10-088
The prompt was totally different from the recording	CA-BB-09-23, CA-BB-09-25, CA-MP1-09-042, CA-MP1-09-043, CA-MP1-09-047, CA-MP1-09-055, CA-MP1-09-058, CA-MP1-09-062, CA-MP1-09-068, CA-MP1-09-069, CA-MP1-09-071

In order to obtain the refined alignments, we performed the following steps:

- Generate the phonetic transcription dictionary (with ‘HDMan’) for all words of the corpus using an enriched version of the CMU Dictionary. The OOV words found in the speech corpus were automatically transcribed using three different G2P algorithms: the previously described MIRA-based G2P, a MaxEnt classifier and a custom designed algorithm called DLOPS [18]. All alternative conversions generated were added to the lexicon and we later relied on HVite to choose the most probable one;
- Generate an initial phonetic transcription of the speech corpus (with ‘HLEd’) using the first available pronunciation from the dictionary for every word of the corpus;
- Scanning the phonetically-transcribed corpus from the previous step, generate initial 3-state, left-right with no skips HMM models (‘monophone HMMs’ in HTK terminology) for all phonemes in the inventory (not including the short pause ‘sp’ “phoneme”) (with ‘HERest’) and re-estimated the initial models 4 times. The pruning thresholds specified with the ‘-t’ switch of HERest were ‘250.0 150.0 1000.0’;
- Added the short pause ‘sp’ HMM model (initially copied from the silence ‘sil’ model at the start/end of corpus utterances) and re-estimated all HMM models another 4 times using the same parameters of HERest;
- Re-generate the phonetic transcription (including generation of short pauses) of the speech corpus (with ‘HVite’) using the best HMM models from the previous step in order to obtain the pronunciations that best match the acoustic data (in case that a word has multiple pronunciations in the dictionary);
- Finally, re-estimating (4 times) the monophone HMMs including short pause with the new corpus transcription and generate the alignments with HVite giving it the option to output the start/end times for every monophone.

3.2. Results

The objective of Blizzard Challenge is to enable speech synthesis researches to evaluate their methods, techniques, algorithms and their processing methodologies starting from a common resource base. Apart from providing the necessary resources, the organizers also ensure a unitary evaluation system that enables a fair comparison between the submissions regarding the similarity to the original speaker, the naturalness and the intelligibility of the synthetic voices.

Before discussing the results we have to mention that one of the main issues of the RACAI submission was generated by an error in the unit selection algorithm which caused the target and unit costs to accumulate over multiple sentences during the synthesis of paragraphs. We were able to locate and fix this fault only after the challenge had already ended and, as a consequence, the paragraphs in our submission contained badly chosen units that affected the Mean Opinion Score (MOS) obtained on them.

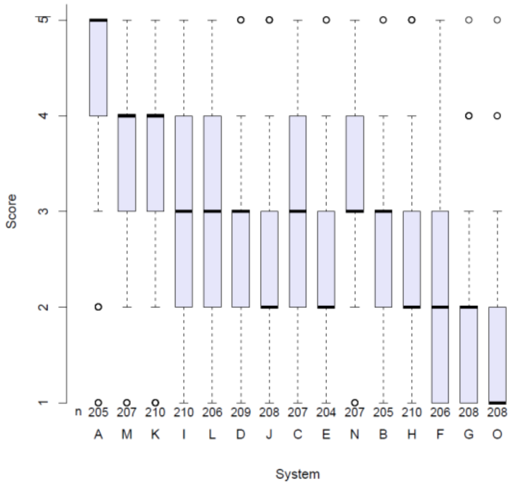

Figure 2 - *RACAI TTS (system J) results (similarity to the original speaker - all listeners/all data)*

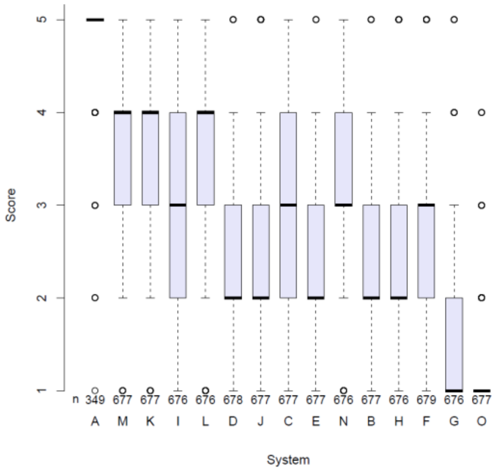

Figure 3 - *RACAI TTS (system J) results (naturalness - all listeners/novel)*

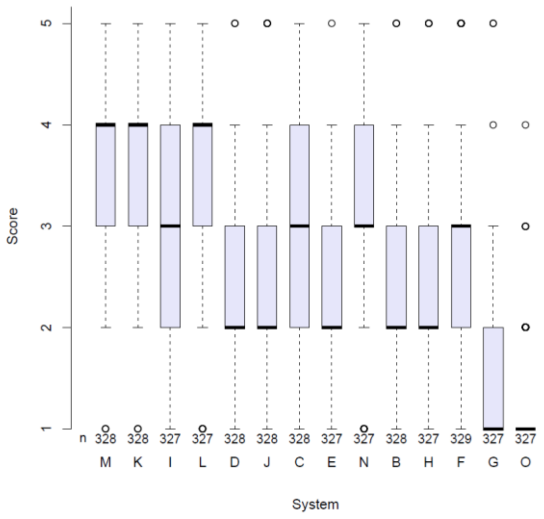

Figure 4 - *RACAI TTS (system J) results (naturalness - all listeners/news)*

The parameter set used in the unit selection process was chosen so that continuous segments would be preferred over those that met the prosody requirements. Tweaking the parameter set was a subjective process and none of the members of the RACAI Team are native English speakers. Figure 2 shows the MOS score for the similarity with the original speaker (actual value 2.4) calculated using all listeners and all data. Figures 3 and 4 contain the naturalness results for the news (2.5) and novel (2.3) sections calculated using the scores from all the users.

Our system obtained an extremely high Word Error Rate (WER) of 46%, which was not an unexpected result since in our parameter tweaking process we favored the naturalness and similarity with the original speaker tests. As mentioned earlier, the system supports a decision threshold for the prosodic modification of the selected units. Lowering this parameter increased the synthesis quality for the Semantically Unpredictable Sentences (SUS), but it resulted in an unnatural sounding voice, which was an undesired effect for the other tests in the challenge.

Finally, another drawback in our approach was the high number of rejected units generated by the corpus filtering procedure, which in some cases resulted in the removal of all candidates for certain diphones (e.g. “SH pau”).

4. Conclusions

In this paper we offered a detailed description of the RACAI TTS synthesis system and the work carried out during the preparation of our Blizzard Challenge 2013 submission.

The participation in the Blizzard Challenge 2013 offered the chance to test our system’s capabilities and to obtain real-world feedback on its performance. Being able to see how different speech synthesis methods and techniques compare to one another is only attainable through the use of the same corpora. One of Blizzard Challenge’s important scientific contributions is the provision of such a unitary testing framework.

During this year’s competition we were able to use one of the largest speech databases we ever worked with. This enabled us to validate our system and to adopt our future research directions.

Adapting the system to English proved challenging, but the adjustments made to the RACAI TTS and the assistive tools we specially developed support the flexibility of the system and enable rapid creation of new voices based on different speech databases and languages.

The previously mentioned adjustments refer to (1) exposing the feature sets used by the NLP components by allowing their modification according to the application’s needs and (2) making the weights of the cost function editable at run-time.

Future development plans include:

- Tweaking some of the algorithms in order improve the speech synthesis speed;
- Increase the naturalness of the synthetic voice by improving the concatenative speech synthesis module;
- Increasing the performance of the corpus filtering method;
- Developing a better prosody prediction system;
- Adding support for parametric speech synthesis.

5. Acknowledgements

The work reported here was funded by the project METANET4U by the European Commission under the Grant Agreement No 270893

6. References

[1] Berger, A. L., Pietra, V. J. D., & Pietra, S. A. D. (1996). A maximum entropy approach to natural language processing. Computational linguistics, 22(1), 39-71.

[2] Ratnaparkhi, A. (1997). A simple introduction to maximum entropy models for natural language processing. IRCS Technical Reports Series, 81.

[3] Samuelsson, C. (1993, June). Morphological tagging based entirely on Bayesian inference. In 9th Nordic Conference on Computational Linguistics.

[4] Schmid, H. (1994, August). Part-of-speech tagging with neural networks. In Proceedings of the 15th conference on Computational linguistics-Volume 1 (pp. 172-176). Association for Computational Linguistics.

[5] Lafferty, J., McCallum, A., & Pereira, F. C. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

[6] Tufiş, D. (1999, January). Tiered tagging and combined language models classifiers. In Text, Speech and Dialogue (pp. 28-33). Springer Berlin Heidelberg.

[7] Boroş, T., Radu, I. and Tufiş, D. (2013). Large tagset labeling with Feed Forward Neural Networks. Case study on Romanian Language. In Proceedings of ACL 2013.

[8] Crammer, K., & Singer, Y. (2003). Ultraconservative online algorithms for multiclass problems. The Journal of Machine Learning Research, 3, 951-991.

[9] Boroş, T. (2013). A unified lexical processing framework based on the Margin Infused Relaxed Algorithm. A case study on the Romanian Language. Accepted in Proceedings of RANLP 2013. 7-13 September Hissar, Bulgaria

[10] Bartlett, S., Kondrak, G., & Cherry, C. (2008, June). Automatic Syllabification with Structured SVMs for Letter-to-Phoneme Conversion. In ACL (pp. 568-576).

[11] Jiampojamarn, S., Cherry, C., & Kondrak, G. (2008, June). Joint Processing and Discriminative Training for Letter-to-Phoneme Conversion. In ACL (pp. 905-913).

[12] De Cheveigné, A., & Kawahara, H. (2002). YIN, a fundamental frequency estimator for speech and music. The Journal of the Acoustical Society of America, 111, 1917.

[13] Young, S., Evermann, G., Kershaw, D., Moore, G., Odell, J., Ollason, D., ... & Woodland, P. (2002). The HTK book (for HTK version 3.2). Cambridge university engineering department.

[14] Erjavec, T. (2004, May). MULTEXT-East Version 3: Multilingual Morphosyntactic Specifications, Lexicons and Corpora. In LREC.

[15] Weide, R. (2005). The Carnegie mellon pronouncing dictionary [cmudict. 0.6].

[16] Fellbaum, C. (2010). WordNet (pp. 231-243). Springer Netherlands.

[17] Amsler, R. A. (1980). The structure of the Merriam-Webster pocket dictionary.

[18] Boroș, T., Ștefănescu, D., & Ion, R. (2012). Bermuda, a data-driven tool for phonetic transcription of words. In Natural Language Processing for Improving Textual Accessibility (NLP4ITA) Workshop Programme (p. 35).