



D3.18. Diseminare

Aceste rezultate au fost obținute prin finanțare în cadrul Programului PN-III Proiecte complexe realizate în consorții CDI, derulat cu sprijinul MEN – UEFISCDI,
Cod: PN-III-P1-1.2-PCCDI-2017-0818, Contract Nr. 73 PCCDI/2018:

“SINTERO: Tehnologii de realizare a interfețelor om-mașină pentru sinteza text-vorbire cu expresivitate”

© 2018-2020 – SINTERO

Acest document este proprietatea organizațiilor participante în proiect și nu poate fi reprodus, distribuit sau diseminat către terți, fără acordul prealabil al autorilor.

Denumirea organizației participante în proiect	Acronim organizație	Tip organizație	Rolul organizației în proiect (Coordonator/partener)
Institutul de Cercetări Pentru Inteligență Artificială “Mihai Drăgănescu”	ICIA	UNI	CO
Universitatea Tehnică din Cluj-Napoca	UTCN	UNI	P1
Universitatea Politehnică din București	UPB	UNI	P2
Universitatea "Alexandru Ioan Cuza" din Iași	UAIC	UNI	P3

Date de identificare proiect

Număr contract:	PN-III-P1-1.2-PCCDI-2017-0818, Nr. 73 PCCDI/2018
Acronim / titlu:	„SINTERO: Tehnologii de realizare a interfețelor om-mașină pentru sinteza text-vorbire cu expresivitate”
Titlu livrabil:	D3.18. Diseminare
Termen:	Noiembrie 2020
Editor:	Mircea Giurgiu (Universitatea Tehnică din Cluj-Napoca)
Adresa de eMail editor:	Mircea.Giurgiu@com.utcluj.ro
Autori, în ordine alfabetică:	Mircea Giurgiu, Beata Lorincz, Maria Nuțu, Adriana Stan
Ofițer de proiect:	Cristian STROE

Rezumat:

În acest livrabil se prezintă articolele științifice susținute și publicate de către partenerul UTCN la conferințe internaționale în anul 2020, paginile web cu demonstratoare online privind adaptarea sistemului de sinteză text-vorbire la stilul și expresivitatea unui nou vorbitor, precum și modul în care au fost implicați studenții în tematica proiectului pe durata elaborării proiectelor de diplomă sau participării la stagii de practică. Inițial au fost identificate conferințele de interes din domeniul aferent cercetărilor. Vom include lista articolelor și un rezumat cu principalele rezultate. Accesul la publicații (cf. 30.11.2020) este asigurat pe pagina web a proiectului în secțiunea dedicată. De asemenea, există o serie de pagini web cu demonstrații online de voci sintetice cu vorbitorii din noul corpus Swara 2.0, din sistemele de adaptare la noi vorbitori, din sistemele generate cu date atipice, respectiv din interfața interactivă online.

Cuprins

1. Introducere	4
2. Identificarea posibilităților de publicare pe anul 2020 și realizări.....	4
3. Publicații științifice în anul 2020.....	4
4. Lucrări de licență în legătura cu tematica proiectului	6
5. Stagii de practică pentru studenți	7
6. Pagini web ale proiectului SINTERO	7
7. Pagini Wiki interne grupului de cercetare cu mostre audio generate de diferite versiuni ale sistemelor de sinteză text – vorbire implementare în etapa a III-a.	8
8. Concluzii	10

1. Introducere

Acest livrabil prezintă o sinteză a articolelor publicate în anul 2020, pagina web a proiectului și o serie de demonstratoare online pentru adaptarea vocii sintetice la stilul și expresivitatea unui nou vorbitor, respectiv adaptarea rapidă a vocii sintetice folosind date audio atipice. Aceste rezultate demonstrează că prin utilizarea arhitecturilor de rețele neuronale profunde de tip Tacotron și DC-TTS se obține o calitate înaltă a vocii sintetice pentru vorbitori în limba română, calitate comparabilă cu rezultate de dată foarte recentă raportate la conferința internațională Interspeech 2020 și la competiția internațională Voice Conversion Challenge 2020. De asemenea, se prezintă referințe către paginile web ale demonstratoarelor online și tematici de cercetare în care au fost implicați și studenții pe durata stagiului de practică sau a elaborării proiectelor de diplomă.

2. Identificarea posibilităților de publicare pe anul 2020 și realizări

Conform cu strategia de diseminare inclusă în formularul de aplicație, pentru fiecare an calendaristic s-au identificat posibilitățile de diseminare și de publicare de articole la conferințe științifice sau în jurnale.

Pentru anul 2020 s-au identificat următoarele posibilități de publicare la conferințe internaționale:

- 2020 ISCA Interspeech, 25-29 Octombrie, Shanghai, China (<https://interspeech2019.org/>) - online
- 2020 IEEE 16th International Conference on Intelligent Computer Communication and Processing, Cluj-Napoca, 3-5 Sept 2020, Cluj-Napoca, Romania (www.iccp.ro)
- The 15th Edition of the International Conference on Linguistic Resources and Tools for Processing Romanian Language, 14-16 Decembrie, Bucuresti, Romania (<https://profs.info.uaic.ro/~consilr/>) – online
- IEEE International Conference – Intelligent Systems, 28-30 August 2020, Varna, Bulgaria (<https://www.ieee-is.org/>) - online
- 24th International Conference on Knowledge-Based and Intelligent Information and Engineering Systems, 16-18 Septembrie 2020, Venetia, Italia (<http://kes2020.kesinternational.org/>) - online
- 11th Nordic Conference on Human-Computer Interaction, Octombrie 2020, online.

3. Publicații științifice în anul 2020

Autori	Beata Lorincz, Maria Nuțu, Adriana Stan, Mircea Giurgiu
Titlu	„An Evaluation of Postfiltering for Deep Learning Based Speech Synthesis with limited data” (https://ieeexplore.ieee.org/abstract/document/9199932)
Ref.	IEEE 10th International Conference on Intelligent Systems (IS), 28-30 August 2020 (online)

Rezumat	<i>Recently, deep neural network (DNN) based speech synthesis achieved close to human speech quality and became the state-of-the art in the field of text-to-speech (TTS) synthesis systems. However, a major part of its efficiency comes from the use of large quantity of high-quality speech recordings. When this data is not available, other approaches are still preferred. This paper evaluates the DNN-based postfiltering of the synthesised speech as a means to increase the quality of DNN based TTS systems trained on very limited speech resources. 20 different systems are compared objectively using the Mel Cepstral Distortion measure. The systems differ in terms of: training data, network architecture, and training method. Out of the 20 initial systems, 7 are evaluated subjectively in listening tests performed for two different speakers. Results show that even when starting from as little as 5 minutes of speech recordings, the postfiltering process improves the quality of the synthetic speech output. So it can, therefore, be used as a training strategy for TTS systems where sufficient high-quality data is not available.</i>
---------	--

Autori	Beata Lorincz
Titlu	„Concurrent Phonetic Transcription, lexical stress assignment and syllabification with deep neural networks” (https://www.sciencedirect.com/science/article/pii/S1877050920318366)
Ref.	Proceedings of the 24th International Conference on Knowledge-Based and Intelligent Information and Engineering Systems, KES 2020, 16-18 Septembrie 2020, online.
Rezumat	<i>This paper evaluates four different sequence-to-sequence deep neural network architectures aimed to jointly solve the tasks of: phonetic transcription, lexical stress assignment and syllabification. These text processing tasks are considered essential components for high quality text-to-speech or automatic speech recognition systems, with the phonetic transcription being the most frequently used in these types of applications. Although each of the tasks has been individually and extensively analyzed in the scientific literature, there are few studies which target a concurrent solution for them. In general, the lexical stress assignment and syllabification are used as augmenting input features to the phonetic transcription model and not considered as target features. The proposed network architectures include recurrent, convolution and attention neural layers and were evaluated on hand-checked English and Romanian datasets. The accuracy of the models was evaluated in terms of accuracy for the concurrent prediction of all three tasks, as well as by discarding the syllabification or lexical stress predictions. The best results were obtained with a combination of convolution and attention layers, where the accuracy of the joint prediction for the three tasks was of 58.96% for English and 86.64% for Romanian. The same model for English obtains an accuracy of 59.70% when syllables are discarded and 64% when the prediction of lexical stress is ignored. With the same best performing model for Romanian an accuracy of 88.83% without syllables and 93.84% without lexical stress is obtained.</i>

Autori	Adriana Stan
Titlu	„RECOApy – Data Recording, Pre-processing and Phonetic Transcription for End-To-End Speech-Based Applications” (https://www.isca-speech.org/archive/Interspeech_2020/pdfs/1184.pdf)
Ref.	ISCA International Conference Interspeech 2020, 25-29 Octombrie 2020, online.

Rezumat	<i>Deep learning enables the development of efficient end-to-end speech processing applications while bypassing the need for expert linguistic and signal processing features. Yet, recent studies show that good quality speech resources and phonetic transcription of the training data can enhance the results of these applications. In this paper, the RECOApy tool is introduced. RECOApy streamlines the steps of data recording and pre-processing required in end-to-end speech-based applications. The tool implements an easy-to-use interface for prompted speech recording, spectrogram and waveform analysis, utterance-level normalisation and silence trimming, as well grapheme-to-phoneme conversion of the prompts in eight languages: Czech, English, French, German, Italian, Polish, Romanian and Spanish. The grapheme-to-phoneme (G2P) converters are deep neural network (DNN) based architectures trained on lexicons extracted from the Wiktionary online collaborative resource. With the different degree of orthographic transparency, as well as the varying amount of phonetic entries across the languages, the DNN's hyperparameters are optimised with an evolution strategy. The phoneme and word error rates of the resulting G2P converters are presented and discussed. The tool, the processed phonetic lexicons and trained G2P models are made freely available.</i>
---------	---

Autori	Kristen M Scott, Simone Ashby, Adriana Stan
Titlu	"Designing a Synthesized Content Feed System for Community Radio" (https://dl.acm.org/doi/abs/10.1145/3419249.3420177)
Ref.	Proceedings of the 11th Nordic Conference on Human-Computer Interaction, October 2020
Rezumat	<i>The use of text-to-speech to generate radio content is largely unexplored, despite the importance of radio in remote parts of the world, where TTS offers a robust means of transforming data into media for low-literate audiences and those without regular internet access. How suitable are TTS voices for meeting the expectations of radio listeners and what type of content are these voices best suited to deliver? We present an application for generating automated daily synthesized weather forecasts for selected locations and language varieties, based on the provision of a regularly updated weather data service. We present results from a pilot listener study aimed at exploring people's reactions to this and other synthesized audio content, as we begin to explore best practices around the design of a synthesized content feed system for community radio.</i>

Articole acceptate spre publicare - Dan Oneață, Alexandru Caranica, Adriana Stan, Horia Cucu, "An Evaluation Of Word-level Confidence Estimation For End-to-end Automatic Speech Recognition", IEEE Spoken Language Technology Workshop, 19-22 ianuarie 2021, Virtual.

Articole în jurnale ISI (în proces de finalizare)

- Beáta Lőrincz, Elena Irimia, Adriana Stan, "RoLEX: An extended Romanian lexical dataset and its evaluation for predicting concurrent lexical information", va fi trimis către IEEE Signal Processing Letters în perioada imediat următoare.
- Beáta Lőrincz, Adriana Stan, Mircea Giurgiu, "RoNNA: Romanian neural network API", va fi trimis către IEEE Signal Processing Letters în perioada imediat următoare.

4. Lucrări de licență în legătura cu tematica proiectului

- Ștefana Cîmpean - "Recunoașterea emoțiilor din vorbire folosind învățarea automată", lucrare de diplomă, iulie 2020.
- Andreea Sarca - "Automatic speech recognition system for Romanian using Deep Speech", lucrare de diplomă, iulie 2020.
- Roxana Marcu - "Automatic language identification from text", lucrare de diplomă, iulie 2020.
- Florin Ciotlăuș - "Music analysis using BLSTM and CNNs", lucrare de diplomă, iulie 2020.
- Cătălin Avram - "Automatic speaker recognition from SWARA corpus", lucrare de diplomă, iulie 2020.

5. Stagii de practică pentru studenți

- Georgiana Săracu - „Detecția stărilor depresive pe baza analizei semnalului vocal”, iulie-august 2020.
- Vlad Crehul - „Implementation of a Tacotron-based text to speech synthesis system”, iulie-august 2020.
- Vlad Crehul - „Testing experiments with Deep Speech automatic speech recognition for Romanian”, iulie-august 2020.
- Bogdan Oros - „Linear regression applied for speech classification”, iulie-august 2020.
- Ana Gheorghiu - „Analysis of prosodic events for music classification”, iulie-august 2020.

6. Pagini web ale proiectului SINTERO

← → ↺ <https://speech.utcluj.ro/sintero/> 🔍 ☆

SINTERO: Tehnologii de realizare a interfețelor om-mașină pentru sinteza text-vorbire cu expresivitate

Proiect finanțat de Ministerul Cercetării și Inovării, Program PN-III-P1-1.2-PCCDI, nr. 73/2018, durata: 2018-2020

Proiect component al proiectului complex

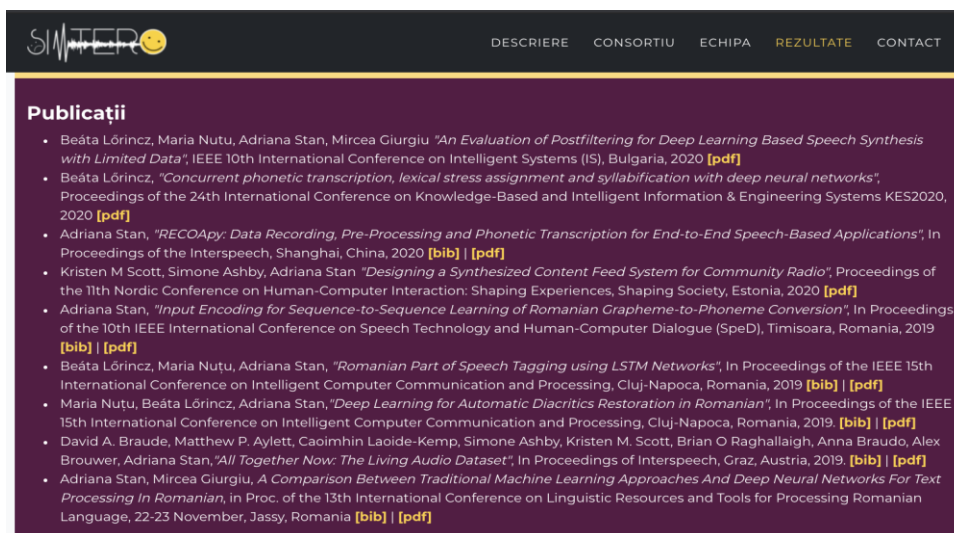
RETEROM

Proiecte paralele

COBILIRO **TEPROLIN** **TADARAV**

Raport științific

- **Raport științific etapa 1 - sinteză (2018) [pdf]**
 - **D1.15.** Identificare pattern-uri prozodice [pdf]
 - **D1.16.** Metode de clasificare a stilului de exprimare din text [pdf]
 - **D1.17.** Analiza metodelor de control și adaptare automată a expresivității [pdf]
 - **D1.18.** Implementarea modulului de control automat al prozodiei [pdf]
 - **D1.19.** Diseminare [pdf]
- **Raport științific etapa 2 - sinteză (2019) [pdf]**
 - **D2.15.** Implementarea modulului de identificare a stilului de vorbire și nivelului de expresivitate din analiza textului [pdf]
 - **D2.16.** Implementarea unui modul de adaptare la un nou vorbitor a sistemului de sinteză [pdf]
 - **D2.17.** Implementarea unui modul de transplantare a prozodiei unui vorbitor în sistemului de sinteză [pdf]
 - **D2.18.** Îmbunătățirea componentei de modelare și control al prozodiei; activități de testare, validare / demonstrare module software [pdf]
 - **D2.19.** Diseminare [pdf]
- **Raport științific etapa 3 - sinteză (2020) [pdf]**
 - **D3.15.** Dezvoltarea unei noi tehnologii pentru adaptarea vocii sintetice la stilul și expresivitatea unui nou vorbitor [pdf]
 - **D3.16.** Dezvoltarea unei noi metode de adaptare rapidă a vocii sintetice folosind date audio atipice [pdf]
 - **D3.17.** Integrare tehnologie nouă și demonstrarea în realizarea interfețelor om-mașină pentru sinteza text – vorbire. [pdf]
 - **D3.18.** Diseminare [pdf]



Publicații

- Beăta Lőrincz, Maria Nutu, Adriana Stan, Mircea Giurgiu "An Evaluation of Postfiltering for Deep Learning Based Speech Synthesis with Limited Data", IEEE 10th International Conference on Intelligent Systems (IS), Bulgaria, 2020 [\[pdf\]](#)
- Beăta Lőrincz, "Concurrent phonetic transcription, lexical stress assignment and syllabification with deep neural networks", Proceedings of the 24th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems KES2020, 2020 [\[pdf\]](#)
- Adriana Stan, "RECOApy: Data Recording, Pre-Processing and Phonetic Transcription for End-to-End Speech-Based Applications", In Proceedings of the Interspeech, Shanghai, China, 2020 [\[bib\]](#) | [\[pdf\]](#)
- Kristen M Scott, Simone Ashby, Adriana Stan "Designing a Synthesized Content Feed System for Community Radio", Proceedings of the 11th Nordic Conference on Human-Computer Interaction: Shaping Experiences, Shaping Society, Estonia, 2020 [\[pdf\]](#)
- Adriana Stan, "Input Encoding for Sequence-to-Sequence Learning of Romanian Grapheme-to-Phoneme Conversion", In Proceedings of the 10th IEEE International Conference on Speech Technology and Human-Computer Dialogue (SpeD), Timisoara, Romania, 2019 [\[bib\]](#) | [\[pdf\]](#)
- Beăta Lőrincz, Maria Nutu, Adriana Stan, "Romanian Part of Speech Tagging using LSTM Networks", In Proceedings of the IEEE 15th International Conference on Intelligent Computer Communication and Processing, Cluj-Napoca, Romania, 2019 [\[bib\]](#) | [\[pdf\]](#)
- Maria Nutu, Beăta Lőrincz, Adriana Stan, "Deep Learning for Automatic Diacritics Restoration in Romanian", In Proceedings of the IEEE 15th International Conference on Intelligent Computer Communication and Processing, Cluj-Napoca, Romania, 2019. [\[bib\]](#) | [\[pdf\]](#)
- David A. Braude, Matthew P. Aylett, Caoimhin Laoide-Kemp, Simone Ashby, Kristen M. Scott, Brian O Raghallaigh, Anna Braudo, Alex Brouwer, Adriana Stan, "All Together Now: The Living Audio Dataset", In Proceedings of Interspeech, Graz, Austria, 2019. [\[bib\]](#) | [\[pdf\]](#)
- Adriana Stan, Mircea Giurgiu, A Comparison Between Traditional Machine Learning Approaches And Deep Neural Networks For Text Processing In Romanian, in Proc. of the 13th International Conference on Linguistic Resources and Tools for Processing Romanian Language, 22-23 November, Jassy, Romania [\[bib\]](#) | [\[pdf\]](#)



Prezentări video ale articolelor diseminate la conferințe cu desfășurare virtuală:

- Beăta Lőrincz, Maria Nutu, Adriana Stan, Mircea Giurgiu "An Evaluation of Postfiltering for Deep Learning Based Speech Synthesis with Limited Data", IEEE 10th International Conference on Intelligent Systems (IS), Bulgaria, 2020
- Beăta Lőrincz, "Concurrent phonetic transcription, lexical stress assignment and syllabification with deep neural networks", Proceedings of the 24th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems KES2020
- Adriana Stan, "RECOApy: Data Recording, Pre-Processing and Phonetic Transcription for End-to-End Speech-Based Applications", In Proceedings of the Interspeech, Shanghai, China, 2020

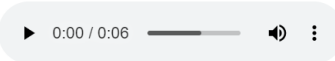
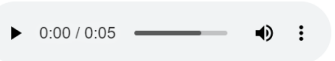
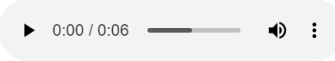
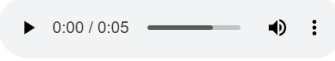
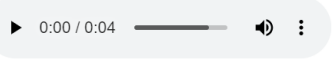
7. Pagini Wiki interne grupului de cercetare cu mostre audio generate de diferite versiuni ale sistemelor de sinteză text – vorbire implementare în etapa a III-a.

Mostre audio aferente noilor voci din SWARA 2.0 sunt disponibile la adresa:

<https://gitlab.utcluj.ro/speech/tts-samples/-/wikis/home>.

<https://gitlab.utcluj.ro/speech/tts-samples/-/wikis/Sisteme%20Tacotron2%20voci%20noi>

Sisteme Tacotron2 voci noi

Voice ID	Sample 1 - "Tezele ar putea fi eliminate în acest an școlar, iar Ministerul Educației analizează mai multe variante."	Sample 2 - "Astfel încât să fie aplicate aceleași reguli și pentru copiii care învață în scenariul verde,"
DLL		
DOL		
EME		

<https://gitlab.utcluj.ro/speech/tts-samples/-/wikis/Date%20atipice>

- Sistem Tacotron2 antrenat cu date preluate din segmentarea și transcrierea automată a unei resurse audio colectate de P1-CoBiLiRo, realizată în P3-TADARAV

Voice ID	Natural	Sample 1 - "Tezele ar putea fi eliminate în acest an școlar, iar Ministerul Educației analizează mai multe variante."	Sample 2 - "Tezele ar putea fi eliminate în acest an școlar, iar Ministerul Educației analizează mai multe variante."
COB			

- Sistem Tacotron2 antrenat cu date din audiobook-ul Mara folosind doar transcrierea ortografică (Mara-letters) sau transcrierea fonetică și accentul (Mara-phoneAcc) rezultate din rețele de predicție concurentă a informației lexicale

Voice ID	Natural	Sample 1 - "Tezele ar putea fi eliminate în acest an școlar, iar Ministerul Educației analizează mai multe variante."	Sample 2 - "Tezele ar putea fi eliminate în acest an școlar, iar Ministerul Educației analizează mai multe variante."
MARA-letters			

<https://gitlab.utcluj.ro/speech/tts-samples/-/wikis/Sistem%20DC-TTS%20Date%20expresive%20neexpresive>

Model	Epoci	Sample 1	Sample 2
Forma ortografică & Neexpresiv	3000		
Forma ortografică & Neexpresiv & Expresiv	4500		
Forma ortografică & Neexpresiv & Expresiv	6000		

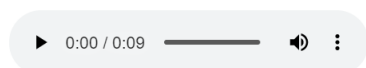
<https://gitlab.utcluj.ro/speech/tts-samples/-/wikis/Sistem%20DC-TTS%20Date%20lexicale>

Text sintetizat:

Sample 1: Ce se va întâmpla cu aceasta după ce vom ieși din starea de urgență.

Sample 2: Elevii vor putea intra în școală doar cu declarația pe propria răspundere completată de către părinți.

Referință (audio natural):



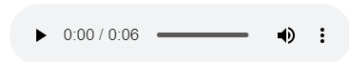
Date text	Sample 1	Sample 2
Forma ortografică		
Forma transcrisă fonetic		

<https://gitlab.utcluj.ro/speech/tts-samples/-/wikis/Sistem%20DC-TTS%20Englez%C4%83%20Singur%20Vorbitor%20Date%20lexicale>

Sistem DC TTS Engleză Singur Vorbitor Date lexicale

LJSpeech Voce

Referință (audio natural):



Text sintetizat:

Sample 1: The demands on the President in the execution of His responsibilities in today's world are so varied and complex

Sample 2: and the traditions of the office in a democracy such as ours are so deep-seated as to preclude absolute security.

Date text	Sample 1	Sample 2
Forma ortografică		
Forma transcrisă fonetic		
Forma transcrisă fonetic cu accent		

<https://gitlab.utcluj.ro/speech/tts-samples/-/wikis/Sistem%20DC-TTS%20Englez%C4%83%20Date%20lexicale>

Sistem DC TTS Engleză Date lexicale

LibriTTS voci

Text sintetizat:

Sample 1: Jimmie Dale slipped his mask into his pocket, and, with the parcel under his arm, stepped to the door and unlocked it.

Sample 2: He paused for an instant on the threshold for a single, quick, comprehensive glance around the room-then passed on out into the street.

Epoci	Sample 1	Sample 2
1000		
3000		

Voci masculine

Date text	Epoci	Sample 1	Sample 2
Forma ortografică	1000		

8. Concluzii

Accesul la publicațiile elaborate în anul 2020 este asigurat la adresa <http://speech.utcluj.ro/sintero/rezultate/>. Pagina web are un conținut dinamic, adaptat cu realizările din proiect, astfel că pentru această raportare se pot accesa și mostre cu semnal sintetic generat de modulul de adaptare la un noi vorbitori, așa cum este prezentat în paginile web cu demonstratoarele.

An Evaluation of Postfiltering for Deep Learning Based Speech Synthesis with Limited Data

Beáta Lőrincz,^{1,2} Maria Nuțu,^{1,2} Adriana Stan,¹ Mircea Giurgiu¹

¹ Communications Department, Technical University of Cluj-Napoca, Romania

² Department of Computer Science, "Babeș-Bolyai" University, Cluj-Napoca, Romania

Email: {beata.lorincz, maria.nutu, adriana.stan, mircea.giurgiu}@com.utcluj.ro

Abstract—Recently, deep neural network (DNN) based speech synthesis achieved close to human speech quality and became the state-of-the art in the field of text-to-speech (TTS) synthesis systems. However, a major part of its efficiency comes from the use of large quantity of high-quality speech recordings. When this data is not available, other approaches are still preferred.

This paper evaluates the DNN-based postfiltering of the synthesised speech as a means to increase the quality of DNN-based TTS systems trained on very limited speech resources. 20 different systems are compared objectively using the Mel Cepstral Distortion measure. The systems differ in terms of: training data, network architecture, and training method. Out of the 20 initial systems, 7 are evaluated subjectively in listening tests performed for two different speakers. Results show that even when starting from as little as 5 minutes of speech recordings, the postfiltering process improves the quality of the synthetic speech output. So it can, therefore, be used as a training strategy for TTS systems where sufficient high-quality data is not available.

Keywords—statistical-parametric synthesis; limited data; deep neural networks; postfiltering; text-to-speech synthesis; Romanian

I. INTRODUCTION

Recently, Tacotron 2 [1] a text-to-speech (TTS) synthesis system based on deep neural networks (DNN), obtained a Mean Opinion Score (MOS) rating equal to 4.53. This rating is very similar to the MOS score for natural speech (4.58 as reported by the authors of [1]). This result, in conjunction with multiple other studies of DNN-based speech synthesis [2, 3, 4, 5, 6, 7, 8, 9], made this approach the new state-of-the-art paradigm for TTS systems. However, all these systems require large amounts of high quality speech recordings for training—over 20 hours of data from a single speaker for most of the previously cited works. So there is still the issue of obtaining good TTS systems for languages or speakers where data is limited. In this case, there are several approaches, such as that of Lee et al. [10] which grades and filters the available data to maximize the quality of the output. Another interesting study for this scenario is that of Sone et al. [11], which uses a deep relational model to estimate a neural network's parameters from the joint distribution of acoustic and linguistic features.

Yet the most common approach is to fine-tune or adapt a pre-trained model's parameters using data from the target speaker or language [12, 13, 14]. Or to append

speaker/language embeddings to the linguistic/acoustic features, so that the model can jointly learn common and discriminative features from the training set [6, 13, 15, 16, 17].

Although not aimed at solving the data limitation problem, the postfilter presented in [18] could be an alternative solution. This postfilter is trained to map the synthetic speech generated by a Hidden Markov Model (HMM) based system into natural samples by using two DNNs, one operating in the Mel cepstral domain, and the other in the spectral domain. Other studies related to this topic are those of Coto-Jimenez and Close [19] and Muthukumar and Black [20]. Coto-Jimenez and Close append a deep neural network with long-short term memory cells as a postfiltering step for HMM-based speech synthesis. Muthukumar and Black also use a recurrent neural network to enhance the output of the ClusterGen statistical-parametric synthesiser. [21] presents a speaker-adaptive postfiltering method for statistical parametric speech synthesis using pre-trained models adapted with limited data to new speakers. A postfilter implemented with Generative Adversarial Network (GAN) is proposed by [22] that is used to learn how to discriminate between synthesised and natural speech. If multi-speaker pre-trained models are available, with few shot methods good quality speech can be obtained for the newly added speaker [23, 24]. To the best of our knowledge, there are no methods which postfilter the DNN-based speech synthesis output without adapting existing models to newly added speakers.

Starting from this overview, we address the problem of developing DNN-based speech synthesis systems with limited speech data by employing a post-synthesis neural network trained to learn the mapping between the synthesised acoustic features and the natural speech features. The method builds upon previously published studies, and focuses on an extensive evaluation of several training strategies and network architectures. 20 different systems are trained and analysed objectively. Out of the 20 systems, 7 were selected for a subjective listening test incorporating two different voices. Both the objective and subjective results illustrate that the postfiltering method can be successfully applied for building TTS systems when large quantities of data are not readily available.

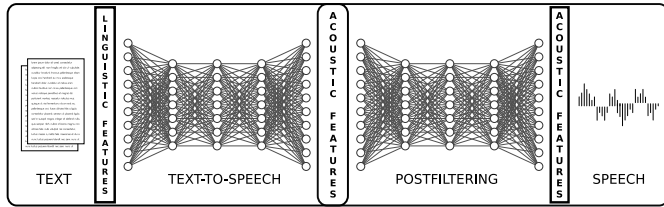


Fig. 1. The postfiltering process.

II. POSTFILTERING SETUP

The scope of our study is to determine a DNN-based postfiltering method for the DNN-based synthesis, such that the final speech output of the system is enhanced even when only limited training data is available. Thus, we employed a two-step procedure: first, a DNN-based TTS system is trained with various amounts of data; and second, the output of the synthesis system is used as input for a postfiltering neural network. An overview of the process is shown in Figure 1.

In DNN-based TTS systems the general trend, nowadays, is to use end-to-end architectures which learn to map raw text-sequences into acoustic representations or waveforms [9]. However, although this training scheme yields very high quality speech output, it is not well suited for the case of limited training data, or real-time synthesis. Therefore, in this study, the synthetic voices are built using the statistical parametric approach. The text is first converted into a set of discrete lexical features, including phonetic transcription, lexical stress assignment, syllabification and part-of-speech tagging, as well as a set of contextual features, such as left and right phonemes, number of syllables and words in a sentence, etc. The complete list of lexical features is based on the common HTS label format [25]. The lexical features are then paired at frame-level to an acoustic parametrization. Phone-level time alignments between text and speech are required, and can be obtained with forced alignment procedures [26].

For the postfiltering step, the entire training dataset prompts are synthesised with the respective TTS system, and the output features are retained. Dynamic Time Warping (DTW) [27] is then applied to align the synthetic and natural feature vectors. The resulting aligned pairs of acoustic features represent the input data for the training of the postfiltering network.

III. EVALUATION

A. Data

The training data consists of the RND1 subset of the SWARA Romanian speech corpus [28].¹ Out of the 17 speakers, we chose 8 female ones: *BAS*, *CAU*, *EME*, *DCS*, *DDM*, *HTM*, *PMM* and *SAM*. As the corpus data does not contain purposely built test sets, two other female speakers: *BEA* and *MAR* were additionally recorded and added to the training set. The prompts were the same as for SWARA speakers, and the recordings took place in similar studio conditions. None of the speakers in the combined dataset are professional speakers.

¹Available online: speech.utcluj.ro/swarasc/

The data is sampled at 48kHz with 16bps, and it was manually segmented at utterance-level. Phoneme state-level alignments were obtained from an iteratively trained HMM-based forced aligner, similar to the first step from the ALISA tool [26]. The aligner used 100 utterances from each speaker. No evaluation of the alignment accuracy was carried out.

B. Synthesis systems

The DNN-based TTS systems followed the Blizzard Challenge 2017 Merlin baseline system setup [29, 30]. Linguistic features were derived with an updated version of the Romanian TTS front-end described in [31].² Acoustic features were extracted with the WORLD vocoder [32], and comprised 59 plus the 0th Mel generalised coefficients, 5 band aperiodicity coefficients and a fundamental frequency (F_0). The acoustic features were augmented with their delta and delta-delta values. The network architecture consists of 6 layers with 1024 nodes each. The system is trained using the *tanh* activation function and the stochastic gradient descent optimizer. A separate network with similar architecture is trained to predict the duration of the phoneme states. The postfiltering uses the same set of acoustic features extracted with WORLD, and a baseline network architecture as the one described in [33].

For the evaluation to provide a correct overview over the effectiveness of the postfiltering, we trained 20 different synthesis systems using the *BEA* data. The systems use different training strategies, quantities of training data, and types of postfiltering network architectures. Their details are presented next.

The *training strategy* analyses: simple DNN-based TTS systems trained on linguistic-to-acoustic pairs of features (ID:M);³ TTS system plus DNN postfiltering trained on synthesised-to-natural acoustic feature pairs (ID:M***PF**);⁴ and DNN speaker adaptation, where an initial eigen voice is trained from the data of all the speakers, and then the network weights are fine tuned for a target speaker (ID:SPKA).

The amount of *training data* for the TTS system was set to: 50 utterances (approx. 5 minutes), 100 utterances (approx. 10 minutes), and 500 utterances (approx. 50 minutes). In the postfiltering step we also selected 50, 100 or 500 utterances. The postfiltering utterances were the same as those used to train the correspondent TTS system. The utterances are random newspaper sentences, and they are not phonetically or acoustically balanced or filtered. To overcome the lack of data, we also used an artificial data enhancement method, in which the original speech samples were added twice to the training set, thus doubling the training data (ID:Db). This method was applied either for just the postfiltering network, or for both the TTS system and the postfiltering (ID:M*Db_P*Db).

In this study, for the postfiltering, only a feed-forward *network architecture* was considered. However, the number of layers (4, 5 and 6), the activation function (*tanh* and

²Online demo: www.romaniantts.com

³The ID refers to the system ID used in Table I.

⁴The asterisk (*) marks a variable value.

ReLU), and the number of neurons per layer (256, 1024, and layer halving or bottleneck: 1025-512-256-512-1024) were examined.

For *speaker adaptation*, different volumes of data from each speaker were used to train the eigen voice (ID:SPKA*_E*): 100 utterances which translates into 1000 total utterances for training, and 500 utterances from each speaker, 5000 in total. The network's adaptation was then performed with either 100 or 500 utterances from the target speaker. For postfiltering a network trained on the same 100 utterances from each of the 10 speakers (ID:*MSPK) was evaluated. This system is similar to the speaker adaptation strategy, in the sense that the network is trained on multi-speaker data. However, no aposteriori weight tuning was performed, and no speaker embeddings were used as features.

Table I summarises the systems' description and the IDs selected for the objective and the subjective evaluations.⁵ Audio samples from all the systems are available at: speech.utcluj.ro/pf_is2020/.

C. Listening test setup

Although many studies have been conducted on the objective analysis of the synthesised speech quality [34], there are still no measures which truly correlate to the perceptual evaluation of the synthesised speech. Hence, subjective listening tests are required. In this evaluation, as the number of initial systems is quite large for a listening test, the 7 most relevant systems were selected and tested with two different voices. The systems and their listening test identifiers are shown in Table I.

The lower bound of our setup is M050 (A)–the TTS system trained on 50 utterances (approx. 5 mins). The upper bounds are M500 (G) trained on 500 utterances (approx. 50 mins) and the natural (H) samples. System M100 (B) is our baseline for the postfiltering process. Out of the various postfilter network architectures, the 6 *tanh* layers of 1024 nodes each (M100_PF100_6TANH1024) exhibited the best objective score for both speakers (see Section IV), and they were included for the evaluation of the postfiltering effect alone (C). Artificially doubling the data in both voice training and postfiltering also showed an increase of the objective score, so that system (M100Db_PF100Db) was selected, as well. As the multi-speaker network could be viewed as an eigen postfilter, systems M100_PF_MSPK and SPKA100_E100 were included for the multi-speaker setup comparison.

The listening test comprised 4 sections: a) *Naturalness*–evaluated using a Mean Opinion Score (MOS) scale consisting of 5 points [1-Unnatural, 5-Natural]; b) *Speaker similarity*–evaluated on a 5-point MOS scale [1-Not similar at all, 5-Very similar]; c) *Intelligibility*–evaluated using a Word Error Rate (WER) measure; and d) *ABX naturalness*–each system was randomly paired with all other systems and listeners had to mark which sample sounds more natural.

⁵Different IDs are used in the objective evaluation as it is easier to follow the multiple setups.

IV. RESULTS

A. Objective measures

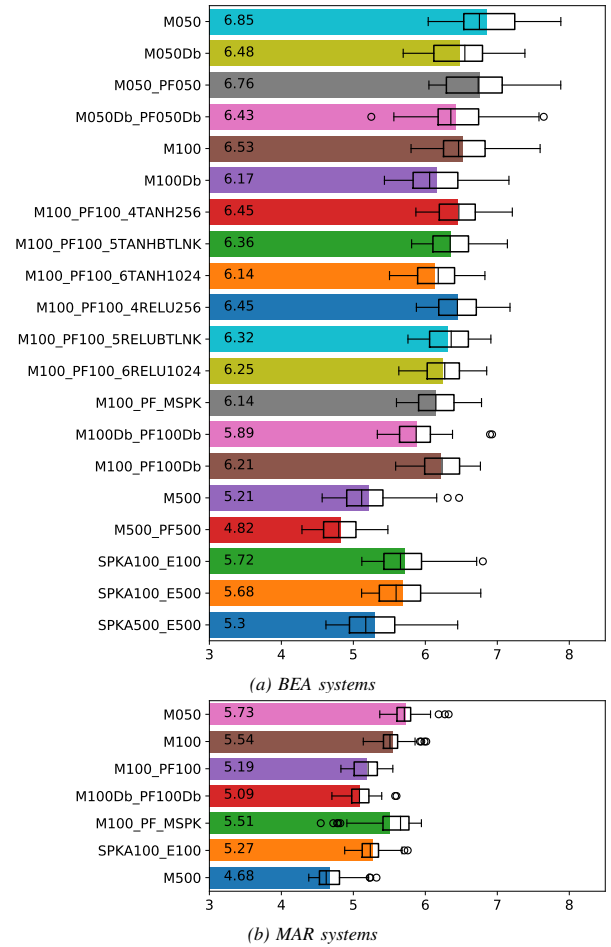


Fig. 2. Average Mel Cepstral Distortion for the (a) BEA and (b) MAR systems. Horizontal bars represent the mean MCD values, and are overlapped with boxplots.

The systems' performance is objectively measured with Mel Cepstral Distortion (MCD) [35]. Because the accuracy of the state-level alignment is unknown, the MCD value was obtained over the best path in DTW, and it does not take into account the 0th coefficient. 50 utterances not contained in the training dataset were synthesised and used to compute the average MCD for speakers BEA and MAR. MAR speaker's distortion included only the listening test systems. Figure 2 shows these results.

As expected, out of the baseline TTS systems, M500 performed the best and M050 the worst. M100's scores are quite low, but artificially doubling the data increases the quality of the synthesis (M050Db, M100Db). The postfiltering also decreases the cepstral distortion relative to the correspondent TTS (M050_PF050, M100_PF100, M100Db_PF100Db, M500_PF500). The average decrease in MCD is 5%, with a maximum of 7.5% for M500_PF500. Postfiltering plus data doubling has the most effect (M050Db_PF050Db,

TABLE I
SYNTHESIS SYSTEMS' DESCRIPTION

No.	System ID	Listening test ID	No. utts voice training	No. utts postfiltering	Postfiltering architecture
1	NAT	H	Natural	N/A	N/A
2	M050	A	50	N/A	N/A
3	M050Db	-	50x2	N/A	N/A
4	M100	B	100	N/A	N/A
5	M100Db	-	100x2	N/A	N/A
6	M500	G	500	N/A	N/A
7	M050_PF050	-	50	50	6 TANH x 1024
8	M050Db_PF050Db	-	50x2	50x2	6 TANH x 1024
9	M100_PF100_4TANH256	-	100	100	4 TANH x 256
10	M100_PF100_5TANHBTLNK	-	100	100	5 TANH (1024-512-256-512-1024)
11	M100_PF100_6TANH1024	C	100	100	6 TANH x 1024
12	M100_PF100_4RELU256	-	100	100	4 RELU x 256
13	M100_PF100_5RELUBTLNK	-	100	100	5 RELU (1024-512-256-512-1024)
14	M100_PF100_6RELU1024	-	100	100	6 RELU x 1024
15	M100_PF_MSPK	E	100	10x100 Multi-speaker	6 TANH x 1024
16	M100Db_PF100Db	D	100x2	100x2	6 TANH x 1024
17	M100_PF100Db	-	100	100x2	6 TANH x 1024
18	M500_PF500	-	500	500	6 TANH x 1024
			No. utts for eigen voice	No. utts for target speaker	
19	SPKA100_E100	F	10x100	100	
20	SPKA100_E500	-	10x500	100	
21	SPKA500_E500	-	10x500	500	

M100Db_PF100Db), with a 10% decrease in MCD for M100Db_PF100Db. Doubling the data for the postfiltering alone (M100_PF100Db) only marginally decreases the MCD. With respect to the postfilter network architecture, the 6 *tanh* layers with 1024 nodes per layer (M100_PF100_6TANH1024) had the best performance. All other network architectures have higher MCD scores, yet not significantly higher. When multiple speakers are available, speaker adaptation is indeed a solution: systems SPKA100_E100, SPKA100_E100, SPKA_E500 have some of the lowest MCD scores. However, the multi-speaker postfilter (M100_PF_MSPK) is comparable only to the speaker-dependent filter.

B. Listening tests

The 7 selected systems, along with natural speech samples were included in two separate listening tests: one for speaker *BEA*, and one for speaker *MAR*. Each voice was evaluated by 20 native Romanian listeners. A couple of listeners misread the MOS scale, and their results were discarded.

Figure 3 shows the results. The best performing system (G) is considered the baseline synthesis system as it uses the most amount of data (approx. 50 minutes). The other systems analysed are of higher interest in the evaluation as they use approximately 10 minutes or less data. It can be observed that the naturalness and the speaker similarity are improved by the postfiltering (C) for both speakers. Artificially doubling the data (D) enhances the output speech's naturalness, but not the speaker similarity. However, the intelligibility is affected by the postfiltering in all setups, and slightly improved by the data doubling. The multi-speaker postfiltering network (E) has similar effects as the speaker dependent postfiltering in terms of naturalness. But it is interesting to notice that when it comes to the speaker similarity section, the network trained with multi-speaker data performs better than the speaker dependent

one. In the ABX section, the preference over each systems is incremental, with a minor exception for *MAR*'s system D.

C. Discussions

Both the objective and the subjective results showed that postfiltering and artificial data doubling have beneficial effects over the quality of the synthesised output, and can be jointly used in scenarios where the training speech data is insufficient. The effect of the postfilter can be interpreted as a result of the fact that as opposed to the TTS network, it only needs to learn a mapping of vectors which are sampled from similar feature spaces. So it actually learns where the TTS system failed with respect to the natural sample, and not to the lexical input. Artificially doubling the data is useful especially in the DNN setup. Here, the training is done at batch-level, and a global overview of the entire dataset is not available to the learning mechanism at each step. As the batches are not selected sequentially, having more samples to learn from can improve the output. Similar to the data doubling, the high sampling frequency (48kHz) also provides more data points. This was also useful for HMM-based synthesis [31]. The fact that the eigen-postfilter was rated higher in the speaker similarity test, could be a result of a better modelling of the speech characteristics in general, and not of the target speaker in particular.

By listening to the samples, there are some interesting observations to be made. Many of the voiced/unvoiced decision errors of the TTS system were corrected by the postfilter. Also, the buzziness of the TTS speech is noticeably reduced. However, the postfilter makes the speech more metallic-sounding, and it could translate into the drop in intelligibility. The decrease of intelligibility is not at all desired, especially in the case of limited training data, and it is already the focus of our next studies.

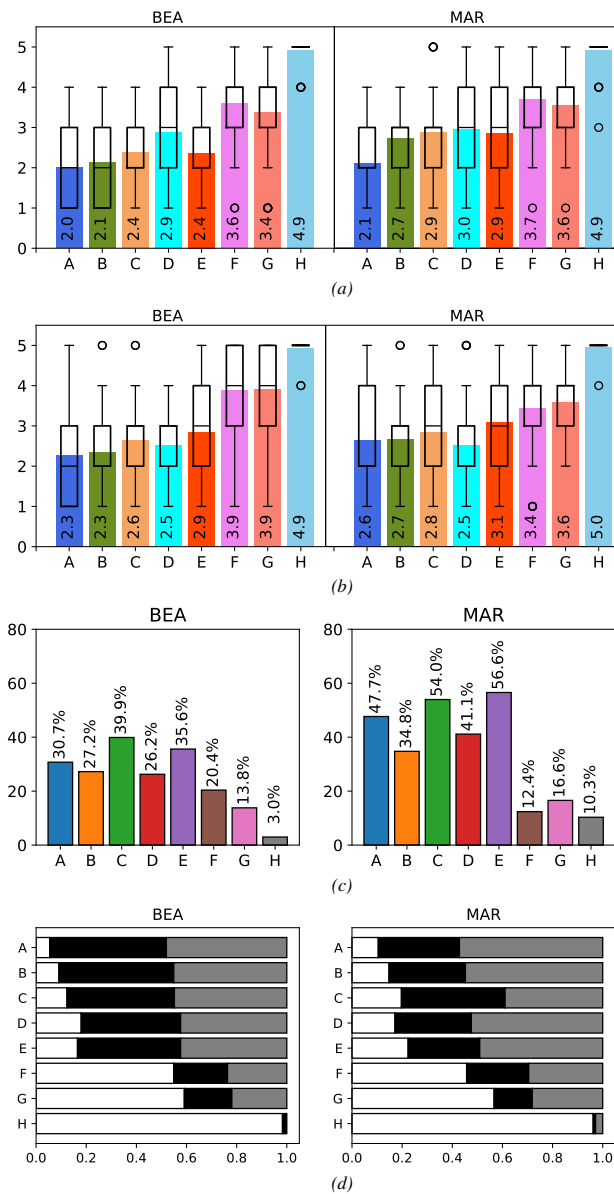


Fig. 3. Listening test results for speakers **BEA** and **MAR**: (a) Naturalness MOS scores, (b) Speaker similarity MOS scores, (c) Intelligibility WER, and (d) ABX preference. In (a) and (b) is fedbars represent the mean value with boxplots overlapped. In (c) bars represent the average WER. In (d) the horizontal bars represent the preference for one system against all others, no preference, or preference for any of the other systems.

V. CONCLUSIONS

This paper described an evaluation of a DNN-based post-filtering method for DNN generated speech using limited resources. The postfilter is trained on pairs of synthetic-to-natural acoustic features, and used to enhance the output of DNN-based TTS system trained on the same data. Starting from as little as 10 minutes of speech from one speaker, this processing chain improves the output synthetic speech as evaluated objectively with MCD, and subjectively in listening tests. A downside of this process at this point is the drop in intelligibility, which can be caused by the metallic speech

characteristic introduced by the postfilter, and it needs to be investigated further.

For future work, we still need to study other network topologies, as well using other vocoders, or adding additional features to the postfilter, such as lexical or speaker embeddings. In the multi-speaker postfilter, we also need to analyse the weight tuning for the target speaker. Using the correct state-level alignments also needs to be considered. This is important for a direct mapping of synthetic-to-natural features. Also, male speaker voices were not evaluated, and might exhibit a different behaviour.

ACKNOWLEDGMENTS

This work was funded through a grant from the Romanian Ministry of Research and Innovation, PCCDI UEFIS-CDI, project number PN-III-P1-1.2-PCCDI-2017-0818/73. We would also like to thank our listening test volunteers.

REFERENCES

- [1] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. J. Skerry-Ryan, R. A. Saurous, Y. Agiomyriannakis, and Y. Wu, "Natural TTS synthesis by conditioning wavenet on mel spectrogram predictions," *CoRR*, vol. abs/1712.05884, 2017. [Online]. Available: <http://arxiv.org/abs/1712.05884>
- [2] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "WaveNet: A Generative Model for Raw Audio," in *arXiv*, 2016. [Online]. Available: <https://arxiv.org/abs/1609.03499>
- [3] A. van den Oord, Y. Li, I. Babuschkin, K. Simonyan, O. Vinyals, K. Kavukcuoglu, G. van den Driessche, E. Lockhart, L. C. C. Rus, F. Stimberg, N. Casagrande, D. Grewe, S. Noury, S. Dieleman, E. Elsen, N. Kalchbrenner, H. Zen, A. Graves, H. King, T. Walters, D. Belov, and D. Hassabis, "Parallel WaveNet: Fast High-Fidelity Speech Synthesis," Google Deepmind, Tech. Rep., 2017. [Online]. Available: <https://arxiv.org/abs/1711.10433>
- [4] Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio, Q. Le, Y. Agiomyriannakis, R. Clark, and R. A. Saurous, "Tacotron: Towards End-to-End Speech Synthesis," in *Proc. Interspeech*, 2017. [Online]. Available: <https://arxiv.org/abs/1703.10135>
- [5] S. Ö. Arik, M. Chrzanowski, A. Coates, G. Diamos, A. Gibiansky, Y. Kang, X. Li, J. Miller, J. Raiman, S. Sengupta, and M. Shoenybi, "Deep Voice: Real-time Neural Text-to-Speech," *CoRR*, vol. abs/1702.07825, 2017. [Online]. Available: <http://arxiv.org/abs/1702.07825>
- [6] S. Ö. Arik, G. F. Diamos, A. Gibiansky, J. Miller, K. Peng, W. Ping, J. Raiman, and Y. Zhou, "Deep Voice 2: Multi-Speaker Neural Text-to-Speech," *CoRR*, vol. abs/1705.08947, 2017. [Online]. Available: <http://arxiv.org/abs/1705.08947>
- [7] W. Ping, A. Peng, A. Gibiansky, S. Ö. Arik, A. Kannan, S. Narang, J. Raiman, and J. Miller, "Deep Voice 3: 2000-Speaker Neural Text-to-Speech," *CoRR*, vol. abs/1710.07654, 2017. [Online]. Available: <http://arxiv.org/abs/1710.07654>
- [8] S. Mehri, K. Kumar, I. Gulrajani, R. Kumar, S. Jain, J. Sotelo, A. C. Courville, and Y. Bengio, "SampleRNN: An Unconditional End-to-End Neural Audio Generation Model," *CoRR*, vol. abs/1612.07837, 2016. [Online]. Available: <http://arxiv.org/abs/1612.07837>
- [9] J. Sotelo, S. Mehri, K. Kumar, J. F. Santos, K. Kastner, A. Courville, and Y. Bengio, "Char2Wav: End-to-end speech synthesis," in *International Conference on Learning Representations (Workshop Track)*, April 2017.
- [10] K.-Z. Lee, E. Cooper, and J. Hirschberg, "A comparison of speaker-based and utterance-based data selection for text-to-speech synthesis," in *Proc. Interspeech 2018*, 2018, pp. 2873–2877. [Online]. Available: <http://dx.doi.org/10.21437/Interspeech.2018-1313>

- [11] K. Sone and T. Nakashika, "DNN-based Speech Synthesis for Small Data Sets Considering Bidirectional Speech-Text Conversion," in *Proc. Interspeech 2018*, 2018, pp. 2519–2523. [Online]. Available: <http://dx.doi.org/10.21437/Interspeech.2018-1460>
- [12] Y. Fan, Y. Qian, F. Soong, and L. He, "Speaker and language factorization in DNN-based TTS synthesis," in *Proc. of ICASSP*, 03 2016, pp. 5540–5544.
- [13] S. Ö. Arik, J. Chen, K. Peng, W. Ping, and Y. Zhou, "Neural Voice Cloning with a Few Samples," *CoRR*, vol. abs/1802.06006, 2018. [Online]. Available: <http://arxiv.org/abs/1802.06006>
- [14] Z. Huang, H. Lu, M. Lei, and Z. Yan, "Linear networks based speaker adaptation for speech synthesis," *arXiv e-prints*, p. arXiv:1803.02445, Mar 2018.
- [15] I. Demirsahin, M. Jansche, and A. Gutkin, "A Unified Phonological Representation of South Asian Languages for Multilingual Text-to-Speech," in *Proc. The 6th Intl. Workshop on Spoken Language Technologies for Under-Resourced Languages*, 2018, pp. 80–84.
- [16] B. Li and H. Zen, "Multi-Language Multi-Speaker Acoustic Modeling for LSTM-RNN based Statistical Parametric Speech Synthesis," in *Proc. of Interspeech*, 2016.
- [17] Y. Lee, T. Kim, and S. Lee, "Voice Imitating Text-to-Speech Neural Networks," *CoRR*, vol. abs/1806.00927, 2018. [Online]. Available: <http://arxiv.org/abs/1806.00927>
- [18] L.-H. Chen, T. Raitio, C. Valentini-Botinhao, Z.-H. Ling, and J. Yamagishi, "A deep generative architecture for postfiltering in statistical parametric speech synthesis," *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, vol. 23, no. 11, pp. 2003–2014, Nov. 2015. [Online]. Available: <http://dx.doi.org/10.1109/TASLP.2015.2461448>
- [19] M. Coto-Jiménez and J. G. Close, "LSTM deep neural networks postfiltering for improving the quality of synthetic voices," *CoRR*, vol. abs/1602.02656, 2016. [Online]. Available: <http://arxiv.org/abs/1602.02656>
- [20] P. K. Muthukumar and A. W. Black, "Recurrent neural network postfilters for statistical parametric speech synthesis," *CoRR*, vol. abs/1601.07215, 2016. [Online]. Available: <http://arxiv.org/abs/1601.07215>
- [21] M. G. ztrk, O. Ulusoy, and C. Demiroglu, "Dnn-based speaker-adaptive postfiltering with limited adaptation data for statistical speech synthesis systems," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2019, pp. 7030–7034.
- [22] T. Kaneko, H. Kameoka, N. Hojo, Y. Ijima, K. Hiramatsu, and K. Kashino, "Generative adversarial network-based postfilter for statistical parametric speech synthesis," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 4910–4914.
- [23] S. Arik, J. Chen, K. Peng, W. Ping, and Y. Zhou, "Neural voice cloning with a few samples," in *Advances in Neural Information Processing Systems*, 2018, pp. 10 019–10 029.
- [24] Y. Jia, Y. Zhang, R. Weiss, Q. Wang, J. Shen, F. Ren, P. Nguyen, R. Pang, I. L. Moreno, Y. Wu *et al.*, "Transfer learning from speaker verification to multispeaker text-to-speech synthesis," in *Advances in neural information processing systems*, 2018, pp. 4480–4490.
- [25] K. Tokuda, H. Zen, and A. Black, "An HMM-Based Speech Synthesis System Applied To English," in *Proc. of SSW*, 10 2002, pp. 227 – 230.
- [26] A. Stan, Y. Mamiya, J. Yamagishi, P. Bell, O. Watts, R. Clark, and S. King, "ALISA: An automatic lightly supervised speech segmentation and alignment tool," *Computer Speech and Language*, vol. 35, pp. 116–133, 2016. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0885230815000650>
- [27] S. Salvador and P. Chan, "Toward accurate dynamic time warping in linear time and space," *Intell. Data Anal.*, vol. 11, no. 5, pp. 561–580, Oct. 2007.
- [28] A. Stan, F. Dinescu, C. Tiple, S. Meza, B. Orza, M. Chirila, and M. Giurgiu, "The SWARA Speech Corpus: A Large Parallel Romanian Read Speech Dataset," in *Proceedings of the 9th Conference on Speech Technology and Human-Computer Dialogue (SpeD)*, Bucharest, Romania, July, 6-9 2017.
- [29] S. King, L. Wihlborg, and W. Guo, "The Blizzard Challenge 2017," in *Proc. Blizzard 2017*, Stockholm, Sweden, September 2017.
- [30] Z. Wu, O. Watts, and S. King, "Merlin: An Open Source Neural Network Speech Synthesis System," in *9th ISCA Speech Synthesis Workshop (2016)*, Sep. 2016, pp. 218–223.
- [31] 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," *Speech Communication*, vol. 53, no. 3, pp. 442–450, 2011.
- [32] M. Morise, F. Yokomori, and K. Ozawa, "WORLD: A Vocoder-Based High-Quality Speech Synthesis System for Real-Time Applications," *IEICE Transactions*, vol. 99-D, pp. 1877–1884, 2016.
- [33] T. Toda, L.-H. Chen, D. Saito, F. Villavicencio, M. Wester, Z. Wu, and J. Yamagishi, "The Voice Conversion Challenge 2016," in *Interspeech 2016*, 2016, pp. 1632–1636.
- [34] R. Ullmann, R. Rasipuram, M. Magimai.-Doss, and H. Bourlard, "Objective intelligibility assessment of text-to-speech systems through utterance verification," *Idiap, Idiap-RR Idiap-RR-06-2015*, 4 2015.
- [35] R. Kubichek, "Mel-cepstral distance measure for objective speech quality assessment," in *Proceedings of IEEE Pacific Rim Conference on Communications Computers and Signal Processing*, vol. 1, May 1993, pp. 125–128 vol.1.



RECOApy: Data recording, pre-processing and phonetic transcription for end-to-end speech-based applications

Adriana Stan

Communications Department
Technical University of Cluj-Napoca, Romania

adriana.stan@com.utcluj.ro

Abstract

Deep learning enables the development of efficient end-to-end speech processing applications while bypassing the need for expert linguistic and signal processing features. Yet, recent studies show that good quality speech resources and phonetic transcription of the training data can enhance the results of these applications. In this paper, the RECOApy tool is introduced. RECOApy streamlines the steps of data recording and pre-processing required in end-to-end speech-based applications. The tool implements an easy-to-use interface for prompted speech recording, spectrogram and waveform analysis, utterance-level normalisation and silence trimming, as well as grapheme-to-phoneme conversion of the prompts in eight languages: Czech, English, French, German, Italian, Polish, Romanian and Spanish.

The grapheme-to-phoneme (G2P) converters are deep neural network (DNN) based architectures trained on lexicons extracted from the Wiktionary online collaborative resource. With the different degree of orthographic transparency, as well as the varying amount of phonetic entries across the languages, the DNN's hyperparameters are optimised with an evolution strategy. The phoneme and word error rates of the resulting G2P converters are presented and discussed. The tool, the processed phonetic lexicons and trained G2P models are made freely available.

Index Terms: speech recording tool, multilingual, phonetic transcription, grapheme-to-phoneme, evolution strategy, sequence-to-sequence, convolutional networks, transformer networks.

1. Introduction

Nowadays, in the development of deep neural networks (DNN) based speech processing applications, most of the signal pre-processing, feature extraction and linguistic annotations are part of the inherent neural learning. This means that systems for automatic speech recognition (ASR) and text-to-speech synthesis (TTS) can be easily trained using only pairs of audio and orthographic transcript [1, 2, 3]. A major advantage of this approach is that training data can be easily and readily found, and that there is no language dependency in the development stage—other than the language specific speech resources. Although this approach yields satisfactory results for most end-user applications, when it comes to high quality systems, found speech data and orthographic input does not suffice [4]. Most of the high-end commercial applications still make use of large amounts of studio recordings and elaborate text processing modules [2, 5].

Hence, there is still a need for tools which can facilitate the development of domain or speaker specific training data, as well as tools which can generate expert linguistic features in a variety of languages. In this context, the first version of the RECOApy

tool is introduced. RECOApy was designed with the main purpose of enabling end-users to record their own data and prepare it for end-to-end speech processing applications. It provides an easy to use interface for prompted speech recording which includes several monitoring and data processing options (see Section 2), as well as a set of highly accurate pre-trained neural network models able to phonetically transcribe the prompts in eight languages.

The task of building grapheme-to-phoneme converters is not novel, but depending on a language's orthographic transparency and onset entropy [6], G2P can be solved using simple rule-based systems (e.g. Finnish) or can pose serious problems even for the most advanced deep learning algorithms (e.g. English). The modern G2P converters aim at solving the problem of phonetic transcription in multiple languages at once. But phonetic lexicons are not readily available in most languages, and researchers are now investigating the use of collaborative online resources, such as Wiktionary,¹ as an alternative. [7] does just this by extracting the phonetic transcriptions in six languages from Wiktionary and validates them over manually crafted lexicons. The authors of [8] also use several online repositories to train and adapt the models from high-resource languages to related low-resource languages. Multilingual G2P was also addressed by changing the grapheme representation: [9] proposes a model which uses byte-level input representation to accommodate different grapheme systems, along with an attention-based Transformer architecture. Ancillary audio data was also used to learn a more optimal intermediate representation of source graphemes in a multi-task training process for multilingual G2P [10].

As the grapheme-to-phoneme task is inherently a sequence to sequence (*seq2seq*) mapping problem, the G2P converter in RECOApy uses this type of learning architecture. Similar approaches were introduced in [11]. The authors map the entire input grapheme sequence to a vector, and then use a recurrent neural network to generate the output sequence conditioned on the encoding vector. [12] describes a G2P model based on a unidirectional LSTM with different output delays and deep bidirectional LSTM with a connectionist temporal classification layer. Milde *et al.* [13] investigate how multitask learning can improve the performance of sequence-to-sequence G2P models. A single *seq2seq* model is trained on multiple phoneme lexicon datasets containing several languages and phonetic alphabets. Esch *et al.* [14] train recurrent neural network-based models to predict the syllabification and stress patterns of the input text for TTS, while also deriving phonetic transcriptions in the process. The use of entire phrases as input to LSTM, biLSTM and CNN-based neural networks and their evaluation in English, Czech and Russian is presented in [15].

¹www.wiktionary.org

Starting from this overview of multilingual and neural networks-based training schemes, RECOApy's G2P module incorporates the use of online collaborative phonetic lexicons and lexicon-tailored seq2seq neural network architectures derived with the help of an evolution strategy. The RECOApy tool, along with the parsed lexicons and complete set of trained models are made freely available. The G2P module can be used as a standalone tool as well.

The paper is organised as follows: Section 2 introduces the recording app and its features. Section 3 presents the phonetic transcription tool development and hyperparameter tuning using evolution strategies. Results of the phonetic converters are discussed in Section 3.3, and conclusions are drawn in Section 4.

2. RECOApy GUI

Recording prompted speech by end-users can be easily performed with any of the numerous free general purpose recording tools available, such as Audacity² or Wavesurfer [16]. But this means that in order to obtain phrase-length speech segments, the continuous recording stream needs to be manually segmented and aligned to the prompts. Or that the recording operator needs to start and stop the recording after each prompt reading. In both cases incorrect readings need to be marked or deleted. This makes the methods tedious, time consuming and error prone.

RECOApy was developed with the main objective of streamlining the end-user speech recording process through a series of pre- and post-processing steps. The GUI application is implemented in Python 3.7 with Tkinter³ and PyAudio.⁴ Its interface is shown in Figure 1. Each prompt is individually displayed to the speaker. Once the recording starts, the input amplitude is monitored and its peak value is displayed such that any signal distortion or low level input can be detected. For additional monitoring, the lower panels of the interface display the waveform and spectrogram of the recorded prompt. Parameters such as sampling frequency and bit depth can be set from the configuration file and depend on the available hardware. The recording operator can easily navigate through the prompts and re-record any of them without any extra setup. Additional features of RECOApy include waveform normalization and silence trimming, as well as a *Safe Copy* option. This means that if the recording operator is unsure of the correctness of the current recording, a backup copy can be saved and later inspected.

Alongside the orthographic form of the prompts, the phonetic transcription can also be displayed. This enables the speaker to read the prompts as intended by the developer. The phonetic transcription may already be available in the prompts, or can be generated and saved from within RECOApy, as introduced in the next section.

3. G2P conversion module

To further enhance the usability and applicability of the recording tool, and given the results of [4], RECOApy can perform an accurate phonetic transcription of the prompts in eight languages: Czech, English, French, German, Italian, Polish, Romanian and Spanish. The data and methods used to develop the grapheme-to-phoneme converters are described next.



Figure 1: RECOApy GUI

3.1. Phonetic lexicons

Even for the mainstream languages, large, manually annotated lexicons are not easily and readily accessible. And most research groups have developed their internal resources [9, 14]. An alternative to this individual effort is the collaborative online resource called Wiktionary. It contains word definitions in 171 languages, of which 45 languages include more than 100,000 entries. The usability of Wiktionary as an alternative to the hand crafted resources has already been studied—[17] shows its great impact on the future directions of lexicography. A significant number of the dictionary entries also include phonetic transcriptions. Their use in G2P methods has been tackled before [7, 8], and can therefore constitute the base for the work presented in this paper.

However, as this resource is constantly expanding, processing the latest database dumps is beneficial [7].⁵ A first step for preparing the lexicons was to determine the list of words which include phonetic entries and to extract these pronunciations. Because the data is crowd-controlled, there is no guarantee that the transcriptions are correct and consistent, or that the entries pertain to a single language. To mitigate these issues, a part of the transcriptions were discarded: entries containing graphemes outside the standard alphabet of the respective language; entries containing phonemes whose occurrence is less than 100 across the respective lexicon; and entries with a phonetic transcription significantly longer than the orthographic form, which might be indicative of two or more pronunciation versions entered in the same field. There was also a set of identical entries (same word, same phonetic transcription), and these were collapsed into a single entry. All lexical stress symbols, if present, were removed. The final number of entries in each lexicon can be found in Table 2.

Due to the potential transcription errors present in Wiktionary, which might affect the performance of the G2P conversion networks, two well-established manually checked lexicons were also included in the evaluation: the English CMU Pronunciation Dictionary⁶ and the Romanian MaRePhor lexicon [18]. Version 0.7b of CMUdict was used and all entries containing numbers and any other symbol except the apostrophe were discarded. The lexical stress in the pronunciation was removed.

²<https://www.audacityteam.org/>

³<https://wiki.python.org/moin/TkInter>

⁴<https://pypi.org/project/PyAudio/>

⁵*wiktionary-20200301** versions of the database were used here.

⁶<http://www.speech.cs.cmu.edu/cgi-bin/cmudict>

3.2. G2P conversion networks

Given the variable lengths of the orthographic and phonetic representations of a word, the task of grapheme-to-phoneme conversion is inherently a sequence-to-sequence mapping problem [19]. Within the set of sequence-to-sequence deep learning algorithms, the most prominent are those based on recurrent (RNN), convolutional (CNN) and full-attention (Transformer) architectures. Although the RNN seq2seq is a highly efficient and adequate method to process temporal or order dependent sequences, it exhibits a slow convergence and high computational complexity. As a result, more and more NLP tasks have been addressed with CNN or hybrid seq2seq alternatives [20, 21]. Along the CNN-based architecture, the Transformer network has been successfully applied in machine translation tasks [22], and G2P conversion networks [9, 23].

These two seq2seq architectures were selected as the starting point in the development of RECOApy’s G2P module. The CNN network’s encoder and decoder are composed of 1D convolution, activation and normalization layers. An attention layer merges the hidden representations of the encoder and decoder. The attention context is concatenated with the decoder representation and passed through another set of 1D convolution layers—denoted as *decoder output*—to generate a softmax output. No residual connections or embedding layers are used. The Transformer network closely follows the architecture of [22], with multi-head self-attention layers combined with fully connected ones in the encoder, decoder and decoder output modules. A positional embedding layer pre-processes the inputs.

For these two neural architectures, the topologies which obtained the best results for English are described in [9, 23, 24]. However, taking into consideration the G2P complexity across languages, as well as the variable dimension of each phonetic lexicon, the architectures’ hyperparameters need to be optimized [25]. Genetic algorithms and evolution strategies manage to provide near-optimal solutions for complex tasks, such as image classification [26] and reinforcement learning [27]. For the current task of G2P conversion across multiple languages and datasets, an evolution strategy (ES) similar to the one described in [26] was adopted. The genes represent various topology parameters, such as number of layers in the encoder or the decoder, the hidden dimensions of the layers or the activation function. The fitness of a genome is determined on its ability to predict a set of word-level phonetic transcriptions. The initial population is randomly selected from the genome pool. In each new generation, the fittest individuals are maintained and bred to create new individuals by random recombinations and mutations. A small sample of the less fit individuals are also bred in order to explore the gene space more thoroughly.

3.3. G2P results

The neural network architectures’ hyperparameters were optimized over 10 generations each with a population size of 10. The fitness of a genome was assessed in terms of the word error rate (WER) computed over a held-out test set of 500 samples at the end of a 20 epoch training process. The small number of epochs and evaluation samples was chosen so that the evolution strategy did not fit the respective train-test split. The number of lexicon entries used for hyperparameter optimisation was limited to 150,000 samples.⁷ The set of genes and gene values for each neural architecture is shown in Table 1. This set does by no means explore the entire hyperparameter search space, but it

Table 1: Set of genes and gene values used in the evolution strategy. The first column marks the gene ID within the genome.

Gene ID		CNN seq2seq
G1	encoder layers	2, 3, 4
G2	encoder layers dimension	32, 64, 128, 256
G3	decoder layers	2, 3, 4
G4	decoder layers dimension	32, 64, 128, 256
G5	decoder output layers	2, 3, 4
G6	decoder output layers dim.	32, 64, 128
G7	activation	ReLU, Linear
G8	optimizer	Adam, RMSprop
G9	batch size	32, 64, 128, 256, 512
		Transformer seq2seq
G1	encoder layers	2, 3, 4
G2	decoder layers	2, 3, 4
G3	embedding dimension	32, 64, 128
G4	attention heads	2, 4
G5	dropout rate	0.01, 0.05, 0.1, 0.15
G6	hidden layer dimension	32, 64, 128, 256 512, 1024
G7	batch size	32, 64, 128, 256, 512

does address some of the key topological variables. The fittest individual for each neural architecture, language and lexicon was selected and trained further on the entire set of entries. An early stopping criterion set to monitor variations of less than 1% in the loss metric over 50 steps prevented overfitting. An 80-20 split with random sampling was employed for training and testing the networks, respectively. The split was different from the one used in the evolution strategy, and the fitness computation data was discarded.

Table 2 shows the results of the G2P conversion module. It includes the total number of entries in each lexicon next to the number of unique entries and phonetic symbols. The number of phonetic symbols represent the set of symbols used in the phonetic transcriptions. For the Wiktionary lexicons these might not fully overlap with the language’s phoneme set. For each neural architecture the genes of the fittest individual are also presented. The accuracy of the G2P is reported in terms of word error rate (WER) and Levenshtein distance-based phoneme error rate (PER) [28]. For entries with multiple pronunciations, the target which minimized the PER and WER was selected.

The best performing architecture varies across languages, as well as in between lexicons of the same language, but the error rate differences are not truly significant. For example, the Romanian Wiktionary lexicon is better fitted by the CNN seq2seq, while for MaRePhor, the Transformer achieves lower WER and PER. For English, both lexicons are better fitted by the Transformer. The dataset’s dimension does not seem to favour any of the architectures either, even though the number of trainable parameters is largely different. For example, the MaRePhor CNN model has 173,672 trainable parameters, and the transformer has only 71,146. But by inspecting the comparable sized lexicons in Czech and Spanish, the Transformer achieves better WER and PER for Czech, yet falls short of the CNN seq2seq in Spanish. This happens despite the fact that Czech and Spanish also exhibit comparable orthographic transparency levels [6]. One conclusion that can be directly drawn from here is that there is no universal recipe to solve the G2P task, and each solution and architecture needs to be tailored to the particular language, phonetic representation, and available resources. The absolute error rates for each language presented here are comparable or lower than the ones in [7] and [14]. But

⁷See Table 2 for the number of entries in each lexicon.

Table 2: *Lexicon descriptions, network hyperparameters and accuracy results of the grapheme-to-phoneme module . The phonetic symbols column indicates the number of distinct phonemes found in the respective lexicon. The gene IDs are listed in Table 1. Best results for each lexicon are highlighted in boldface.*

Lang	Lexicon	Entries	Unique entries	Phonetic symbols	Model	G1	G2	G3	G4	G5	G6	G7	G8	G9	WER	PER
EN	CMUdict	132,585	123,874	39	CNN	2	128	2	128	3	128/64/32	ReLU	RMSp	512	29.82	11.41
					Transformer	4	3	64	4	0.01	512	64	-	-	23.16	8.03
	Wiktionary	71,332	48,773	39	CNN	2	128	2	128	3	128/64/32	ReLU	RMSp	256	28.92	12.39
					Transformer	4	4	32	4	0.01	128	128	-	-	22.50	8.23
RO	MaRePhor	72,375	72,375	40	CNN	3	64	2	32	3	64/32/32	Lin	Adam	128	2.64	0.5
					Transformer	2	4	32	2	0.05	64	64	-	-	2.30	0.42
	Wiktionary	63,013	62,733	32	CNN	3	128	2	32	3	128/64/32	Lin	Adam	512	3.00	0.50
					Transformer	3	2	64	2	0.05	64	256	-	-	3.58	0.71
CZ	Wiktionary	42,014	41,419	41	CNN	2	32	4	128	3	64/32/32	Lin	RMSp	128	11.69	3.84
					Transformer	2	2	32	2	0.05	64	32	-	-	9.45	2.37
DE	Wiktionary	327,296	315,793	51	CNN	3	128	3	32	3	128/64/32	ReLU	Adam	512	5.50	1.43
					Transformer	4	2	64	2	0.05	32	64	-	-	8.80	2.24
ES	Wiktionary	49,346	42,732	31	CNN	3	128	4	64	2	128/64	ReLU	Adam	128	9.81	2.20
					Transformer	2	4	32	4	0.05	32	32	-	-	11.90	2.95
FR	Wiktionary	1,121,714	1,115,343	35	CNN	3	128	3	32	3	128/64/32	ReLU	Adam	512	4.38	1.02
					Transformer	2	3	64	2	0.05	128	64	-	-	4.78	0.97
IT	Wiktionary	29,826	29,242	28	CNN	2	128	4	128	2	64/32	ReLU	RMSp	256	18.67	4.44
					Transformer	2	2	64	2	0.01	512	64	-	-	19.04	5.00
PL	Wiktionary	35,646	35,544	48	CNN	4	64	2	128	2	128/64	ReLU	Adam	128	3.59	1.84
					Transformer	3	2	64	4	0.05	1024	128	-	-	2.98	1.34

the different lexicon versions and train-test splits make a direct, fully correct comparison impossible. As an overview of the architectures’ performance, the average WER across lexicons for the CNN seq2seq is 11.80%, and the PER is 3.95%. For the Transformer, the average WER is 10.95% and PER is 3.22%.

Inspecting the performance over the supervised lexicons, for MaRePhor the results are in line with previous studies [29]. The CMUdict error rates obtained here (23.16% WER and 8.03% PER) are slightly lower than the ones reported in the state-of-the-art methods ([23]: 22.1% WER and 5.1% PER). However, the CMUdict versions and train-test splits are different. When applying the same architecture⁸ on this version of the CMUdict, the results were 22.8% WER and 7.19% PER. It is interesting, however, to notice that the ES evolved a rather similar architecture for the Transformer seq2seq. It may be the case that an evaluation of the fitness over larger number of epochs and validation set, would yield the same architecture, and therefore same performance. One other interesting fact in the results reported here is the high WER for Italian. When analysing the decoded sequences from both networks, it was found that over 60% of the erroneous words had only a single incorrect phoneme, and it was mostly the case of vowel-semivowel substitutions.

Looking at the inference duration, the MaRePhor CNN seq2seq model processes 5000 words in approximately 55 seconds, while the Transformer seq2seq does it in around 120 seconds.⁹ Given the large difference in inference time and only minor drops of accuracy for some of the lexicons, RECOApy integrates the CNN-based models alone. However, the trained Transformer models are available in the tool’s webpage.

⁸The authors of [23] kindly provided their implementation.

⁹On an NVidia GeForce RTX 2080 Ti GPU with 12GB vRAM.

4. Conclusions

This paper introduced RECOApy, a tool for data recording, pre-processing and phonetic transcription of training data aimed at speech-based end-to-end applications. The tool enables fast and accurate recording of text prompts at various sampling rates and bit depths, while offering the recording operator the possibility to supervise the quality of the process as well. Additional automatic options to normalise the audio and to discard the start and end silence segments are also available. One other important feature of RECOApy is that of automatic phonetic transcription of the prompts in eight languages: Czech, English, French, German, Italian, Polish, Romanian and Spanish. The G2P module consists of state-of-the-art neural network based architectures achieving low word and phoneme error rates across all languages. As a conclusion, the RECOApy tool can most certainly be used as a reliable means to develop the training data for end-to-end speech-based applications. In fact, our research group has already collected over 50 hours of prompted speech from non-expert volunteers using this recording tool. The tool, lexicons and models are available here: www.gitlab.utcluj.ro/sadriana/recoapy/.

Future developments of the tool include the addition of more languages in the G2P module, a more in-depth analysis of the hyperparameter space, as well as the augmentation of the prompts with syllabification and lexical stress assignment. A potential significant development would be to also include prosodic cues—similar to [30].

5. Acknowledgement

This work was funded through a grant from the Romanian Ministry of Research and Innovation, PCCDI – UEFISCDI, project number PN-III-P1-1.2-PCCDI-2017-0818/73.

6. References

- [1] A. Y. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Sathesh, S. Sengupta, A. Coates, and A. Y. Ng, “Deep speech: Scaling up end-to-end speech recognition,” *ArXiv*, vol. abs/1412.5567, 2014.
- [2] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. J. Skerry-Ryan, R. A. Saurous, Y. Agiomyriannakis, and Y. Wu, “Natural TTS synthesis by conditioning wavenet on mel spectrogram predictions,” in *Proceedings of ICASSP*, 2018.
- [3] S. Ö. Arik, M. Chrzanowski, A. Coates, G. Diamos, A. Gibiansky, Y. Kang, X. Li, J. Miller, J. Raiman, S. Sengupta, and M. Shoyebi, “Deep Voice: Real-time Neural Text-to-Speech,” in *Proceedings of ICML*, 2017.
- [4] J. Fong, J. Taylor, and K. Richmond, “A Comparison of Letters and Phones as Input to Sequence-to-Sequence Models for Speech Synthesis,” in *Proc. of Interspeech*, 2019, pp. 223–227.
- [5] V. Wan, C. an Chan, T. Kenter, J. Vit, and R. Clark, “Chive: Varying prosody in speech synthesis with a linguistically driven dynamic hierarchical conditional variational network,” in *Proceedings of the 36th International Conference on Machine Learning (ICML 2019)*, 2019, pp. 3331–3340.
- [6] G. Gillon, *Phonological Awareness 2nd Edition: From Research to Practice*. The Guilford Press, 2018.
- [7] T. Schlippe, S. Ochs, and T. Schultz, “Web-based tools and methods for rapid pronunciation dictionary creation,” *Speech Communication*, 118, January 2014., vol. 56, p. 101, 2014.
- [8] A. Deri and K. Knight, “Grapheme-to-phoneme models for (almost) any language,” in *Proceedings of the 2016 Conference of the Association for Computational Linguistics*. Berlin, Germany: Association for Computational Linguistics, August 2016.
- [9] M. Yu, H. Nguyen, A. Sokolov, J. Lepird, K. Sathyendra, S. Choudhary, A. Mouchtaris, and S. Kunzmann, “Multilingual Grapheme-to-Phoneme Conversion with Byte Representation,” in *Proc. of ICASSP*, 2020.
- [10] J. Route, S. Hillis, I. Czeresnia Etinger, H. Zhang, and A. W. Black, “Multimodal, Multilingual Grapheme-to-Phoneme Conversion for Low-Resource Languages,” in *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 192–201.
- [11] K. Yao and G. Zweig, “Sequence-to-sequence neural net models for grapheme-to-phoneme conversion,” in *Proc. of Interspeech*, 2015.
- [12] K. Rao, F. Peng, H. Sak, and F. Beaufays, “Grapheme-to-phoneme conversion using long short-term memory recurrent neural networks,” in *ICASSP*, 2015.
- [13] B. Milde, C. Schmidt, and J. Köhler, “Multitask Sequence-to-Sequence Models for Grapheme-to-Phoneme Conversion,” in *Proc. of Interspeech*, 2017.
- [14] D. van Esch, M. Chua, and K. Rao, “Predicting pronunciations with syllabification and stress with recurrent neural networks,” in *Proceedings of Interspeech*, 2016.
- [15] M. Juzová, D. Tihelka, and J. Vit, “Unified Language-Independent DNN-Based G2P Converter,” in *Proceedings of Interspeech*, 2019.
- [16] K. Sjölander and J. Beskow, “Wavesurfer - an open source speech tool,” in *Proceedings of Interspeech*. ISCA, 2000, pp. 464–467.
- [17] C. M. Meyer and I. Gurevych, “Wiktionary: A new rival for expert-built lexicons? exploring the possibilities of collaborative lexicography,” in *Electronic Lexicography*. S. Granger and M. Paquot, Eds. Oxford: Oxford University Press, November 2012, pp. 259–291.
- [18] S.-A. Toma, A. Stan, M.-L. Pura, and T. Barsan, “MaRePhoR - An Open Access Machine-Readable Phonetic Dictionary for Romanian,” in *Proceedings of the 9th Conference on Speech Technology and Human-Computer Dialogue (SpeD)*, Bucharest, Romania, July, 6-9 2017.
- [19] I. Sutskever, O. Vinyals, and Q. V. Le, “Sequence to Sequence Learning with Neural Networks,” in *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, ser. NIPS’14, 2014, p. 3104–3112.
- [20] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, “Convolutional Sequence to Sequence Learning,” in *Proceedings of the 34th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 70, 2017, pp. 1243–1252.
- [21] W. Yin, K. Kann, M. Yu, and H. Schütze, “Comparative study of CNN and RNN for natural language processing,” *CoRR*, vol. abs/1702.01923, 2017.
- [22] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 5998–6008.
- [23] S. Yolchuyeva, G. Németh, and B. Gyires-Tóth, “Transformer based grapheme-to-phoneme conversion,” *Proceedings of Interspeech*, Sep 2019.
- [24] —, “Grapheme-to-Phoneme Conversion with Convolutional Neural Networks,” *Applied Sciences*, vol. 9, no. 6, p. 1143, 2019.
- [25] G. Melis, C. Dyer, and P. Blunsom, “On the state of the art of evaluation in neural language models,” *CoRR*, vol. abs/1707.05589, 2017.
- [26] T. Hinz, N. Navarro, S. Magg, and S. Wermter, “Speeding up the hyperparameter optimization of deep convolutional neural networks,” *International Journal of Computational Intelligence and Applications*, vol. 17, pp. 1 850 008:1–1 850 008:15, 2018.
- [27] F. P. Such, V. Madhavan, E. Conti, J. Lehman, K. O. Stanley, and J. Clune, “Deep Neuroevolution: Genetic Algorithms Are a Competitive Alternative for Training Deep Neural Networks for Reinforcement Learning,” *CoRR*, vol. abs/1712.06567, 2017.
- [28] V. Levenshtein, “Binary Codes Capable of Correcting Deletions, Insertions and Reversals,” *Soviet Physics Doklady*, vol. 10, p. 707, 1966.
- [29] A. Stan, “Input Encoding for Sequence-to-Sequence Learning of Romanian Grapheme-to-Phoneme Conversion,” in *Proceedings of the 10th IEEE International Conference on Speech Technology and Human-Computer Dialogue (SpeD)*, Timisoara, Romania, October, 10-12 2019.
- [30] R. Wilhelms-Tricarico, J. B. Reichenbach, and G. Marple, “The Lessac Technologies Hybrid Concatenated System for Blizzard Challenge 2013,” in *Proceedings of Blizzard Challenge*, 2013.

24th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems

Concurrent phonetic transcription, lexical stress assignment and syllabification with deep neural networks

Beáta Lőrincz^{1,2}¹ Technical University, Communications Department, 400027, Cluj-Napoca, Romania² Babeş-Bolyai University, Faculty of Mathematics and Computer Science, 400084, Cluj-Napoca, Romania

Abstract

This paper evaluates four different sequence-to-sequence deep neural network architectures aimed to jointly solve the tasks of: phonetic transcription, lexical stress assignment and syllabification. These text processing tasks are considered essential components for high quality text-to-speech or automatic speech recognition systems, with the phonetic transcription being the most frequently used in these types of applications.

Although each of the tasks has been individually and extensively analyzed in the scientific literature, there are few studies which target a concurrent solution for them. In general, the lexical stress assignment and syllabification are used as augmenting input features to the phonetic transcription model and not considered as target features.

The proposed network architectures include recurrent, convolution and attention neural layers and were evaluated on hand-checked English and Romanian datasets. The accuracy of the models was evaluated in terms of accuracy for the concurrent prediction of all three tasks, as well as by discarding the syllabification or lexical stress predictions. The best results were obtained with a combination of convolution and attention layers, where the accuracy of the joint prediction for the three tasks was of 58.96% for English and 86.64% for Romanian. The same model for English obtains an accuracy of 59.70% when syllables are discarded and 64% when the prediction of lexical stress is ignored. With the same best performing model for Romanian an accuracy of 88.83% without syllables and 93.84% without lexical stress is obtained.

© 2020 The Authors. Published by Elsevier B.V.

This is an open access article under the CC BY-NC-ND license (<https://creativecommons.org/licenses/by-nc-nd/4.0>)

Peer-review under responsibility of the scientific committee of the KES International.

Keywords: phonetic transcription; syllabification; lexical stress assignment; neural networks; sequence-to-sequence; English; Romanian

1. Introduction

Natural language processing (NLP) tasks are still vital components in many speech and language processing flows, such as text-to-speech (TTS) or automatic speech recognition (ASR) applications. Knowing how the words are pro-

E-mail address: beata.lorincz@com.utcluj.ro (Beáta Lőrincz^{1,2}).

nounced is essential for producing high quality speech synthesis and speech recognition systems [1]. The exact pronunciation of a word in a language depends mostly on its constituent phones. However, the syllable structure and lexical stress play an important role in assigning the correct meaning and emphasis to it [2].

None of these three tasks is a novel or lacks extended research. However, an important aspect of the speech processing applications is that they require real-time processing. Having separate modules for each of the tasks translates into an overhead for the processing flow. As a countermeasure in this work a concurrent solution for the phonetic transcription, lexical stress assignment and syllabification is proposed. The goal is to evaluate the feasibility of a single model which jointly predicts all three linguistic information. The languages used in the evaluation are English and Romanian.

In terms of algorithmic approach, recent studies showed that deep neural network based sequence-to-sequence models are highly efficient in solving the tasks of phonetic transcription, lexical stress assignment and syllabification in various languages [3]. Recurrent neural networks (RNN) such as Long Short-Term Memory (LSTM) or bidirectional LSTM (BLSTM) networks, as well as convolutional neural networks (CNN) are applied in studies achieving state-of-the-art results. Starting from this observation, the current study employs recurrent, convolution and attention-based neural network architectures as the main training strategies, and evaluates several combinations and variations of a sequence-to-sequence learning scenario. The paper is structured as follows: Section 2 presents some of the state-of-the-art research for the selected tasks, Section 3 describes the tasks and network architectures, and Section 4 details the datasets and training parameters used in the experiments. In Section 5 the results are discussed and the conclusions are summarized in Section 6.

2. Related work

Although there are numerous studies which apply traditional machine learning techniques to solve each of the three tasks, this section will focus on the more recent neural and deep neural approaches. Results reported using recurrent network for the task of phonetic transcription for English are presented first, followed by the application of CNN and other architectures used for the same task. After the results on the lexical stress assignment and syllabification for English, studies addressing all three tasks for the Romanian language are introduced.

[1] reports that "joint-sequence n-gram models and sequence-to-sequence models" are used most frequently for the task of phonetic transcription. The authors of the paper present experiments and results for twenty languages using LSTM and BLSTM networks trained to predict the stress or phonetic transcription. The best results for English are accuracies of 93.1% for stress prediction using a parallel input of phonemes and graphemes, while for pronunciation prediction the best accuracy is 64.2% when the input is enhanced with stress and syllabification information. In [4] the phonetic transcription task is approached with the combination of LSTM networks and n-gram models and an accuracy of 78.7% is obtained on the CMU dataset [5]. Sequence-to-sequence LSTM and BLSTM models are applied in [6] for phonetic transcription on three English datasets, reaching an accuracy of 76.45% on the CMU dataset. RNN architectures are also used in [7] reporting an accuracy of 72.36% using BLSTMs with alignment constraints and in [3] where sequence-to-sequence models with LSTM and attention modules obtain an accuracy of 75.12%, both on the CMU dataset.

CNNs and mixed CNN and RNN architectures were also successfully applied in the context of phonetic transcription. [8] presents a sequence-to-sequence model that uses convolutional and pooling layers in its encoder and a BLSTM as a decoder that results in an accuracy of 74.87% on the CMU dataset. [9] discusses a CNN sequence-to-sequence model with a non-sequential greedy decoder where instead of using the outputs of the previous time step the input is selected based on the highest probability at each time step. The model predicts the phonemes with an accuracy of 75.90%.

[10] applies attention mechanism for the task of phonetic transcription. The authors report results of an accuracy of 77.90% on the CMU dataset with mentioning that the model outperforms sequence-to-sequence models of recurrent or convolutional architectures and has a considerably smaller model size compared to previous approaches.

Regarding the individual tasks of syllabification and lexical stress assignment, these are generally approached with rule or decision tree based algorithms. The neural network based methods applied to these tasks are referred to as data-driven methods. [11] describes experiments with BLSTM networks used to segment speech data into syllables, in which the model correctly detects boundaries with an accuracy of 90.25% on syllable level. Detection of syllables

in speech data using RNNs is also presented in [12], where the accuracy achieved is 94%. Both of these studies performed experiments on the TIMIT dataset [13]. Classification of lexical stress patterns from speech data using a single multilayer perceptron neural network is presented in [14]. Neural networks composed of three feed forward layers are used in [15] achieving an accuracy of 86% for stress prediction on the CELEX dataset [16].

[17] summarizes results for the task of phonetic transcriptions in the Romanian academic literature and lists the available datasets for the Romanian language. The author presents experiments using sequence-to-sequence methods composed of LSTM layers where the input is augmented with syllabification and/or lexical stress assignment. The best results reported is an accuracy of 97.90% on word level where the input data is encoded using grapheme embeddings and is enhanced with syllabification and lexical stress.

A comparison of traditional and deep neural network based methods for all three individual tasks applied to the Romanian language is presented in [18]. The authors report an accuracy of 96% for stress assignment, 98% for syllabification and 99.60% for phonetic transcription using deep neural networks on Romanian datasets.

All the works presented above aim to solve the tasks of phonetic transcription, lexical stress assignment and syllabification individually, in some cases the input data being augmented with additional linguistic information in order to enhance the prediction output for the selected task [1], whereas the goal of our experiments is to predict all three tasks simultaneously.

3. Method overview

The three NLP tasks selected for this evaluation were chosen especially due to their importance in TTS systems. Even though the current deep neural TTS architectures can implicitly solve these tasks, they still play an important role in the intelligibility and prosodic characteristics (rhythm and intonation) of the synthetic speech. It is, therefore, essential to provide accurate phonetic transcription, lexical stress assignment and syllabification at the input of the system. The following subsections describe the tasks of phonetic transcription, lexical stress assignment and syllabification for English and Romanian with language specific details which are relevant to the experiments, as well as the network architectures and layers used to solve these tasks.

3.1. Selected NLP Tasks

Phonetic transcription or grapheme-to-phoneme conversion refers to the process of representing the written form of a word into an acoustically-derived form suited for a correct reading or pronunciation of the respective word. Specific standard alphabets, such as the International Phonetic Alphabet (IPA)¹ are used in this process. In this respect, the language choice for the experiments touches two extremes of the complexity degree: Romanian is a newly reformed language and its spelling is very close to the acoustic realization of the words, while English poses some of the most complex phonetic transcription problems due to its dialect varieties and rich historic background. A SAMPA notation² example of this difference is given below:

<i>economy</i>	<i>iy k aa n ah m iy</i>
<i>economia</i>	<i>e k o n o m ij a</i>

It can be observed that in Romanian the surface form of the word is closely related to its phonetic transcription, while in English the surface form does not correlate with the constituent phones.

Lexical stress assignment refers to determining the most prominent syllable(s) within a word. The correct lexical stress is important especially in the context of homographs. Romanian does not have predefined rules for placing the stress, but the majority of words have the penultimate or final syllable stressed [19]. This rule does not apply to the derivative and inflected word forms and neither does to neologisms [18]. In the case of polysyllabic words, Romanian assigns the stress to a single syllable within a word as opposed to English where the lexical stress varies in its prominence for syllables of the word [20]. The syllables in English can have a primary or secondary stress, or

¹ <https://www.internationalphoneticassociation.org/>

² <https://www.phon.ucl.ac.uk/home/sampa/>

no stress. The majority of words (approximately 80%) have the primary stress placed on the first syllable [21]. An example of lexical stress assignment in English and Romanian is given below:

economy *eco'nomy*
economia *economi'a*

Syllabification is the process of splitting a word into smaller units, generally pertaining to a single pronunciation block of phonemes. This task, as well, is highly dependent on the language. Syllables define units within words that also contribute to the pronunciation rhythm of words. The Romanian language has 7 basic rules for segmenting the words into syllables with exceptions applied in the case of compound or hyphenated compound words [22]. In contrast, English has many more rules and so far there is no study which claims to present an exhaustive list of syllabification rules. Although some rules are presented in several papers and linguistic books [23, 24, 25]. To exemplify, the syllabification of the word *economy* in English and *economia* in Romanian is given below:

economy *e-co-no-my*
economia *e-co-no-mi-a*

By combining all three lexical information presented above into a single representation, the target sequence form for the prediction model is obtained:

economy *iy - k aa ' - n ah - m iy*
economia *e - k o - n o - m ij ' - a*

3.2. Prediction network architectures

In recent work, next to the recurrent networks such as LSTMs or BLSTMs, deep convolutional networks are frequently used for solving NLP tasks [1, 7, 26]. The recurrent or convolution based architectures can be embedded into sequence-to-sequence learning methods [6], as these are suitable for tasks where the input and target sequences are of different lengths. Recurrent or convolutional models extended with attention layers are also efficiently used [3] or are replaced entirely by attention based models [10]. In our experiments sequence-to-sequence models composed of recurrent, convolutional and attention layers are explored for the concurrent solution of the proposed tasks. The properties of these architectures in regards to the selected tasks are discussed shortly in the following paragraphs.

The **recurrent architectures** allow for preserving context related information that is beneficial when modelling sequential data. A connection between the current and previous states is introduced by RNNs. This is further enhanced in LSTM units where this connection is replaced with a memory cell that facilitates learning long-term dependencies [27]. LSTMs can retain useful information from previous features, but cannot make use of future input. To solve this [28] introduces bidirectional RNNs where both forward and backward hidden states are stored in order to profit from future and past context. BLSTMs are LSTM networks that can model long-term dependencies in the input data using the forward and backward states for accessing past and future features.

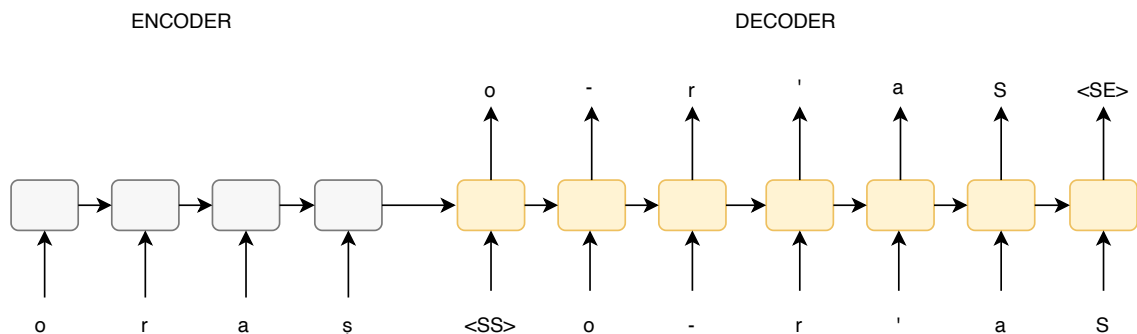


Fig. 1: Input and output for sequence-to-sequence model

The **convolutional architectures** are suitable for computer vision as images have a compositional structure; but also for NLP tasks as texts have a similar formation consisting of sentences, words, n-grams composed of characters [26]. Convolution layers learn features or aspects of the input data using filters. These filters slide over the representation of the input data and calculate the dot product between the filter and input to obtain the activation maps that are passed to the next layer. The output of the network is passed as final step through a fully-connected layer.

The **attention mechanism** is successfully applied in NLP tasks such as machine translation as it allows the encoder and decoder to look at the entire input sequence. This is beneficial compared to recurrent architectures where long-term dependencies can be remembered, but still remain a challenge. The attention is a method that helps the model focus on important information by calculating the relevance of a set of values based on keys and queries. The attention weights refer to the relevance of the encoder hidden states (the values) that are calculated based on the decoder hidden states (query) and encoder hidden states (keys). The output is finally calculated based on the weighted sum of the values [29].

As the input sequence and output sequences of the tasks targeted in this study are of different lengths, **sequence-to-sequence learning** methods [30] are a suitable approach. These architectures include an encoder and a decoder where the encoder learns a vector representation of the input sequence. Based on this representation the decoder learns how to predict the next character of the output sequence. Both the encoder and decoder are neural networks that can be composed of different types of layers. An example input and output sequence of the model is shown in Figure 1.

4. Evaluation

4.1. Datasets

For English, a modified version of the Carnegie Mellon University (CMU) [5] Pronouncing Dictionary was used. The dictionary was extended by [31] with syllabification information using a Support Vector Machine method. The results of this method applied on the CELEX lexical database [16] exhibit a 98% accuracy. Examples of the English training samples are listed in Table 1. The numbers mark the primary, secondary or no stress information for each vowel, while the hyphen marks the syllable boundaries.

In the Romanian part of the experiments, an augmented version of the MaRePhoR dictionary [32] was used. The dictionary was extended by [17] with syllabification and lexical stress assignment using the RoSyllabiDict [33] and DEX Online Database [34] dictionaries. Samples from the dataset entries are exemplified in Table 1. In contrast to the English entries, there is only a single stress in the Romanian words marked with an apostrophe in the examples. The hyphen marks the syllable boundaries.

The datasets were split into training and test sets with a ratio of 80 to 20, where the entries for both of the sets were randomly selected. The same split was maintained for all the evaluations. The number of train and test samples for each dataset is shown in Table 2.

Table 1: Dictionary entry samples from the syllabified CMU and augmented MaRePhor datasets. The syllabification, lexical stress and phonetic transcriptions are all combined into one target string in the training data.

English CMU		Romanian MaRePhor	
Word	Lexical information	Word	Lexical information
SYSTEM	s ih1 - s t ah0 m	BRUTĂRIE	b r u - t @ - r ' ij - e
CASHIERS	k ae2 - sh ih1 r z	PACHETUL	p a - k - j = ' e - t u l

Table 2: Number of training and test samples per dataset

Dataset	Total samples	Training samples	Test samples
English CMU	129,402	103,522	25,880
Romanian MaRePhor	62,873	56,586	6,287

In preparation for the network input, the training data was one-hot-encoded (OHE) based on the number of distinct characters used in the input including the special characters used to mark the syllabification and lexical stress. For Romanian, letter embeddings (LE) obtained with the help of Gensim³ library were also used. The order of the embedding was set to 30 and derived from the Romanian Wikipedia pages' dump.⁴

4.2. Training architectures and parameters

The details of the four network architectures selected for this study are presented next.

The **CNN with Attention** model was provided an input layer with the size of the number of input characters (31 for Romanian and 30 for English), followed by 3 convolutional layers with a hidden dimension of 128, kernel size of 3 and the ReLU activation function. The decoder's input layer size was set to the number of phonemes tallied up with the characters marking the syllable boundaries and accents (42 for Romanian and 43 for English). The output of the encoder and decoders were combined with dot product in the attention module. The output was passed through a softmax activation layer and the dot product of the result and decoder output were concatenated. This combined decoder output was passed through 2 convolutional layers followed by 2 dense layers. The architecture of the model is illustrated in Figure 2.

The **LSTM with Attention** model uses a similar architecture with the difference that the encoder and decoder are composed of a single LSTM layer with a hidden size of 256, and the attention module outputs calculated with a dot product are only passed through two dense layers.

The **LSTM** and **BLSTM** models' encoder and decoder were composed of a single LSTM layer with 128 units, or a BLSTM layer with 64 units. These models were initialized with the same input data as the models using attention modules. Compared to the LSTM model, the BLSTM encoder was extended with the concatenation of the forward and backward states. The architecture of the BLSTM model is depicted in Figure 3.

The RMSprop optimizer with a categorical cross-entropy loss was used for all the training architectures. The number of epochs varied from 200 to 1000 with a batch size of 256 and the latent dimensions in the range of 64 to 256. These sizes were selected based on initial tests. For Romanian, the networks were also trained for a higher number of epochs (i.e. 1000), but this did not result in higher values for the accuracy. All implementations were derived with the help of the Keras⁵ deep learning library.

5. Results and Discussions

The prediction results of each network architecture for both languages was evaluated in terms of accuracy. This was calculated for all three tasks jointly by dividing the number of correct predictions by the total number of entries in the test set. The predictions were considered accurate only if the phonetic transcription, lexical stress and syllabification were all correct. A separate accuracy measure was calculated by discarding the lexical stress or syllabification, but using the same predictions made by the networks trained on all three tasks. The accuracy was not calculated for the combination of lexical stress assignment and syllabification, as the phonetic transcription task was considered essential for the purpose of creating an enhanced input for the TTS systems.

The results of the English dataset are listed in Table 3. The CNN with Attention model performed the best, achieving an accuracy of 58.96% on all three concurrent tasks. Character level accuracy calculation was also analyzed as both the phonetic transcription and lexical stress assignment in English are more complex tasks compared to Romanian. The highest character level accuracy of 94.94% on all three tasks was achieved by the BLSTM model.

Accuracy results of the Romanian dataset are summarized in Table 4. Again, the CNN with Attention network performed the best with an accuracy of 86.64% when validated on all three tasks. The prediction values are better when the lexical stress assignment is ignored, this might be caused by the fact that the Romanian language does not have specific rules for lexical stress assignment. The OHE and LE input data encoding achieved similar accuracy values.

³ <https://radimrehurek.com/gensim/>

⁴ <https://linguatools.org/tools/corpora/wikipedia-monolingual-corpora/>

⁵ <https://keras.io/>

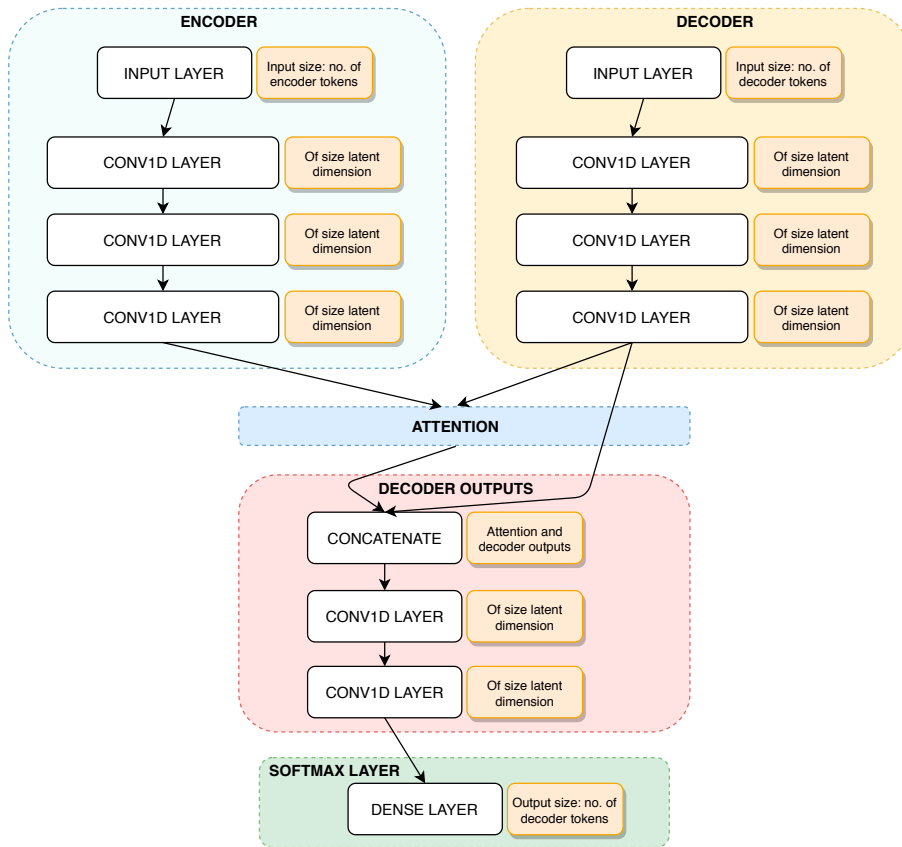


Fig. 2: CNN with Attention model architecture. The encoder and decoder are composed of an input layer followed by 3 convolution layers. The hidden state outputs of the encoder and decoder are combined in the attention module. The output of the decoder and attention modules are concatenated and passed through 2 convolution layers and finally the output is calculated by the fully-connected dense layer.

Table 3: Network parameters and accuracy results on the English dataset. Best results are marked in boldface. The columns *without syllabification* and *without lexical stress* refer to the same target data but discard the predictions made for the respective lexical information.

Architecture	Data input format	Epochs / Batch size / Latent dimension	Accuracy (%)			
			all tasks	without syllabification	without lexical stress	character level
CNN w Attention	OHE	200 / 256 / 128	58.96	59.70	64.00	85.53
LSTM	OHE	200 / 256 / 128	52.81	53.41	56.33	90.30
BLSTM	OHE	200 / 256 / 64	56.02	56.72	59.71	94.94
LSTM w Attention	OHE	200 / 256 / 128	53.79	54.43	57.24	81.15

For both datasets the CNN model with attention achieved the best results. The BLSTM model achieved an accuracy similar to the CNN based model, but in terms of training time the CNN model was the fastest to train, while the BLSTM model the slowest. The LSTM models produced the smallest accuracy values, adding an attention layer to the model increased the accuracy with approximately 1% for both languages. Results validate the addition of the attention layer, as it increased the accuracy values for the LSTM and CNN networks. The LSTM and BLSTM networks without attention layers achieved similar accuracy values with the BLSTM slightly outperforming the LSTM model.

These results are similar to those reported in previous studies and are comparable to the ones reported in [1]. Their best accuracy reported on English for the task of phonetic transcription is 64.2% where the input data is augmented with the lexical stress and syllabification information. The CNN with Attention model achieves an accuracy

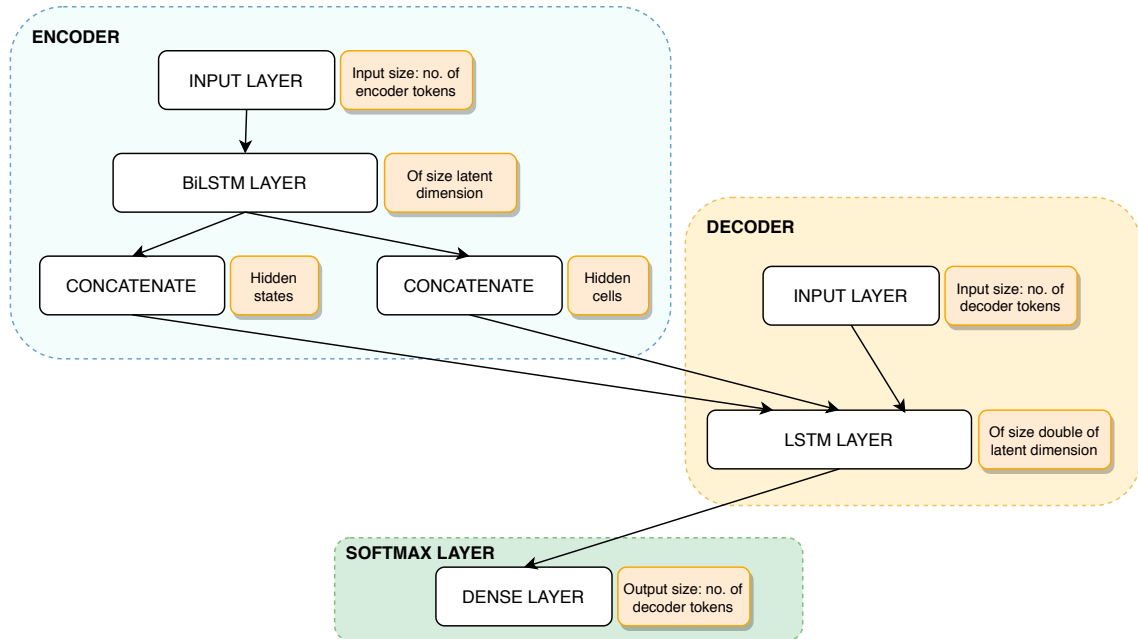


Fig. 3: BLSTM model architecture. The encoder is composed of a BLSTM layer. The hidden states and cells of this layer are concatenated and together with the decoder input data passed to the LSTM layer of the decoder. The output is calculated by passing the output of the decoder through a fully-connected dense layer.

Table 4: Network parameters and accuracy results on the Romanian dataset. Best results are marked in boldface. The columns *without syllabification* and *without lexical stress* refer to the same target data but discard the predictions made for the respective lexical information.

Architecture	Data input format	Epochs / Batch size / Latent dimension	Accuracy (%)		
			all tasks	without syllabification	without lexical stress
CNN w Attention	OHE	200 / 256 / 128	86.64	88.83	93.84
LSTM	OHE	200 / 256 / 128	83.17	85.37	90.09
BLSTM	OHE	200 / 256 / 64	85.19	87.64	91.51
LSTM w Attention	OHE	200 / 256 / 128	84.25	86.62	90.90
CNN w Attention	OHE	1000 / 256 / 128	80.91	83.35	88.44
LSTM	OHE	1000 / 256 / 128	84.60	86.89	91.19
BLSTM	OHE	1000 / 256 / 64	86.10	88.13	92.73
CNN w Attention	LE	200 / 256 / 128	85.67	87.82	92.52
LSTM	LE	200 / 256 / 128	83.75	85.89	90.14
BLSTM	LE	200 / 256 / 64	85.26	87.62	91.65
LSTM w Attention	LE	200 / 256 / 128	86.26	88.68	92.87

of 64% when the lexical stress prediction is discarded and the accuracy is calculated based on the correct prediction of phonemes and syllable boundaries. Our results were achieved on models trained on 103,522 samples compared to the dataset of 340,265 used in [1].

In both English and Romanian, next to the phonetic transcription that is considered the base task, the lexical stress assignment prediction fails more often compared to syllabification. Again, this could be interpreted based on the complexity of the lexical stress assignment in both languages.

6. Conclusions

Four different model architectures concurrently predicting the phonetic transcription, syllabification and lexical stress assignment on English and Romanian datasets were evaluated in this work. The output of these models aims to provide enhanced input for TTS systems by reducing the computational cost of deriving each linguistic information individually. For both languages, the networks composed of convolution layers combined with attention modules performed the best, achieving an accuracy of 86.64% for Romanian and 58.96% for the English dataset. The recurrent architectures comprising LSTM and BLSTM layers achieved similar accuracy and the addition of attention layers to the models increased the accuracy results. The fact that there are only slight differences between the architectures' results means that the sequence-to-sequence models with attention layers are suited for this task's complexity, and that any increase in the accuracy could be derived from larger training datasets, or by employing incremental learning methods.

Acknowledgment

This work was supported by a grant of the Romanian Ministry of Research and Innovation, PCCDI – UEFISCDI, project number PN-III-P1-1.2-PCCDI-2017-0818/73, within PNCDI III.

References

- [1] D. van Esch, M. Chua, K. Rao, Predicting Pronunciations with Syllabification and Stress with Recurrent Neural Networks., in: INTERSPEECH, 2016, pp. 2841–2845.
- [2] P. Taylor, Text-to-Speech Synthesis, Cambridge University Press, 2009.
- [3] B. Milde, C. Schmidt, J. Köhler, Multitask Sequence-to-Sequence Models for Grapheme-to-Phoneme Conversion., in: INTERSPEECH, 2017, pp. 2536–2540.
- [4] K. Rao, F. Peng, H. Sak, F. Beaufays, Grapheme-to-phoneme conversion using Long Short-Term Memory recurrent neural networks, in: 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2015, pp. 4225–4229.
- [5] Carnegie Mellon University (CMU) Pronouncing Dictionary.
URL <http://www.speech.cs.cmu.edu/cgi-bin/cmudict>
- [6] K. Yao, G. Zweig, Sequence-to-sequence neural net models for grapheme-to-phoneme conversion, arXiv preprint arXiv:1506.00196.
- [7] A. E.-D. Mousa, B. W. Schuller, Deep Bidirectional Long Short-Term Memory Recurrent Neural Networks for Grapheme-to-Phoneme Conversion Utilizing Complex Many-to-Many Alignments., in: Interspeech, 2016, pp. 2836–2840.
- [8] S. Yolchuyeva, G. Németh, B. Gyires-Tóth, Grapheme-to-phoneme conversion with convolutional neural networks, Applied Sciences 9 (6) (2019) 1143.
- [9] M. Chae, K. Park, J. Bang, S. Suh, J. Park, N. Kim, L. Park, Convolutional sequence to sequence model with non-sequential greedy decoding for grapheme to phoneme conversion, in: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2018, pp. 2486–2490. doi:10.1109/ICASSP.2018.8462678.
- [10] S. Yolchuyeva, G. Németh, B. Gyires-Tóth, Transformer based Grapheme-to-Phoneme Conversion, Proc. Interspeech 2019 (2019) 2095–2099.
- [11] C. Landsiedel, J. Edlund, F. Eyben, D. Neiberg, B. Schuller, Syllabification of conversational speech using Bidirectional Long-Short-Term Memory Neural Networks, in: 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2011, pp. 5256–5259.
- [12] A. Hunt, Recurrent neural networks for syllabification, Speech communication 13 (3-4) (1993) 323–332.
- [13] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, DARPA TIMIT acoustic-phonetic continuous speech corpus CD-ROM. NIST speech disc 1-1.1, NASA STI/Recon technical report n 93.
- [14] M. A. Shahin, B. Ahmed, K. J. Ballard, Classification of lexical stress patterns using deep neural network architecture, in: 2014 IEEE Spoken Language Technology Workshop (SLT), IEEE, 2014, pp. 478–482.
- [15] J. Arciuli, J. Thompson, Improving the assignment of lexical stress in text-to-speech systems, in: Proceedings of the 11th Australian International Conference on Speech Science and Technology, 2006, pp. 6–8.
- [16] R. H. Baayen, R. Piepenbrock, H. Van Rijn, The CELEX lexical database (CD-ROM). Linguistic data consortium, Philadelphia, PA: University of Pennsylvania.
- [17] A. Stan, Input Encoding for Sequence-to-Sequence Learning of Romanian Grapheme-to-Phoneme Conversion, in: Proceedings of the 10th IEEE International Conference on Speech Technology and Human-Computer Dialogue (SpeD), IEEE, Timisoara, Romania, 2019, pp. 1–6.
- [18] A. Stan, M. Giurgiu, A Comparison Between Traditional Machine Learning Approaches And Deep Neural Networks For Text Processing In Romanian, in: Proceedings of the 13th International Conference on Linguistic Resources and Tools for Processing Romanian Language (ConsILR), 2018.
- [19] V. Franzén, M. Horne, Word stress in Romanian (1997).
- [20] A. Cutler, Lexical Stress in English Pronunciation, Wiley Online Library, 2015, Ch. 6, pp. 106–124. doi:10.1002/9781118346952.ch6.

- [21] J. Arciuli, P. Monaghan, N. Seva, Learning to assign lexical stress during reading aloud: Corpus, behavioral, and computational investigations, *Journal of Memory and Language* 63 (2) (2010) 180–196.
- [22] A. Stan, M. Giurgiu, Romanian language statistics and resources for text-to-speech systems, in: 2010 9th International Symposium on Electronics and Telecommunications, IEEE, 2010, pp. 381–384.
- [23] J. Zhang, H. J. Hamilton, Learning English syllabification rules, in: Conference of the Canadian Society for Computational Studies of Intelligence, Springer, 1998, pp. 246–258.
- [24] R. Treiman, A. Zukowski, Toward an understanding of English syllabification, *Journal of Memory and Language* 29 (1) (1990) 66–85.
- [25] J. C. Wells, Syllabification and allophony, *Studies in the pronunciation of English: A commemorative volume in honour of AC Gimson* (1990) 76–86.
- [26] A. Conneau, H. Schwenk, L. Barrault, Y. Lecun, Very deep convolutional networks for text classification, arXiv preprint arXiv:1606.01781.
- [27] Z. Huang, W. Xu, K. Yu, Bidirectional LSTM-CRF models for sequence tagging, arXiv preprint arXiv:1508.01991.
- [28] A. Graves, A. Mohamed, G. Hinton, Speech recognition with deep recurrent neural networks, in: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, 2013, pp. 6645–6649. doi:10.1109/ICASSP.2013.6638947.
- [29] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in neural information processing systems, 2017, pp. 5998–6008.
- [30] I. Sutskever, O. Vinyals, Q. V. Le, Sequence to sequence learning with neural networks, in: Advances in neural information processing systems, 2014, pp. 3104–3112.
- [31] S. Bartlett, G. Kondrak, C. Cherry, On the syllabification of phonemes, in: Proceedings of human language technologies: The 2009 annual conference of the north american chapter of the association for computational linguistics, 2009, pp. 308–316.
- [32] S.-A. Toma, A. Stan, M.-L. Pura, T. Barsan, MaRePhoR - An Open Access Machine-Readable Phonetic Dictionary for Romanian, in: Proceedings of the 9th Conference on Speech Technology and Human-Computer Dialogue (SpeD), Bucharest, Romania, 2017, pp. 1–6. URL http://adrianastan.com/papers/2017_SPEd_Marephor.pdf
- [33] A.-M. Barbu, Romanian Lexical Data Bases: Inflected and Syllabic Forms Dictionaries., in: LREC, 2008.
- [34] The Romanian Explicative Dictionary (DEX) online. URL www.dexonline.ro

Designing a Synthesized Content Feed System for Community Radio

KRISTEN M. SCOTT, Madeira Interactive Technologies Institute, Portugal

SIMONE ASHBY, ITI / LARSyS, Portugal

ADRIANA STAN, Technical University of Cluj-Napoca, Romania

The use of text-to-speech to generate radio content is largely unexplored, despite the importance of radio in remote parts of the world, where TTS offers a robust means of transforming data into media for low-literate audiences and those without regular internet access. How suitable are TTS voices for meeting the expectations of radio listeners and what type of content are these voices best suited to deliver? We present an application for generating automated daily synthesized weather forecasts for selected locations and language varieties, based on the provision of a regularly updated weather data service. We present results from a pilot listener study aimed at exploring people's reactions to this and other synthesized audio content, as we begin to explore best practices around the design of a synthesized content feed system for community radio.

CCS Concepts: • **Human-centered computing** → *Empirical studies in ubiquitous and mobile computing*; **Accessibility technologies**; • **Applied computing** → *Media arts*.

Additional Key Words and Phrases: synthetic speech; text-to-speech; voice; radio; Romania; ubiquitous computing

ACM Reference Format:

Kristen M. Scott, Simone Ashby, and Adriana Stan. 2020. Designing a Synthesized Content Feed System for Community Radio. XX, X, Article XXX (X 2020), 6 pages. <https://doi.org/XXXXXXX>

[Good morning] [location]. The forecast for this **morning** from [6:00] is [clear sky], with a temperature of [10] degrees, with wind of [8,2] meters per second in the [easterly] direction. The forecast for the [afternoon] from [12:00] to [18:00] is [cloudy], with a temperature of [10] degrees, with wind of [5,8] meters per second in the [south-west] direction. Weather forecast provided by YR.no application, the Norwegian Meteorological Institute and NRK.

1 INTRODUCTION

As DIY synthetic speech voices become easier to generate - for example using free, open source toolkits such as Idlak [8] - we can expect the use of text-to-speech (TTS) to expand to a greater number of use cases over the coming years. Through our interactions with Alexa, Siri and other personal digital assistants, we are gradually becoming aware of some of the possibilities and limitations of interacting with high-quality TTS voices [1, 6, 7] and the complex social implications of speech technology voice choices [3, 4]. We are also encountering an increasing array of uncanny valley

Authors' addresses: Kristen M. Scott, kristen.scott@m-iti.org, Madeira Interactive Technologies Institute, Funchal, Madeira, Portugal; Simone Ashby, simone.ashby@m-iti.org, ITI / LARSyS, Funchal, Madeira, Portugal; Adriana Stan, adrianac.stan@gmail.com, Technical University of Cluj-Napoca, Cluj-Napoca, Romania.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2020 Association for Computing Machinery.

Manuscript submitted to ACM

Manuscript submitted to ACM

phenomena around the use of TTS voices in varying contexts, such as the recent Google Assistant haircut appointment demo, described by one YouTube commenter as "[a]n AI with near perfect vocal fry" ¹.

However, personal digital assistants are far from being the only application of synthetic speech. Apart from [10], much remains to be explored for uncovering listener needs and preferences around the use of TTS in broadcasting, and radio in particular. And yet TTS is poised to play a pivotal role in the success and longevity of both commercial and community stations. As part of a European consortium dedicated to expanding and augmenting an open technology stack for low-power FM community radio stations [5], we are particularly focused on the needs of our community partners (e.g. in rural Romania) for ensuring that stations are sustainable and do not violate their state granted licensing agreements. For our Romanian stations, this means providing non-stop, 24/7 broadcast content. Thus, given the task of designing and integrating purpose-built TTS applications within our FM radio technology stack, we start with the initial assumption that *some* amount of TTS will be deemed acceptable, and even necessary, by our local partners and their listening audiences. However, questions remain as to what other types of TTS content would be appropriate, as well as the type of synthetic voice (or voices) to use.

We present an automated synthesized audio weather forecast generator, which was designed through collaborating stations managers and volunteers and which we plan to extend to further content types for broadcast. We include results of our preliminary listening study of the generated and proposed content with Romanian speakers residing in Romania.

2 AUTOMATED WEATHER FORECAST APPLICATION

In conversations between the Romanian station managers and the fishermen chief in one of the communities, it was determined that weather and wind speed and direction forecast information are crucial for them. Current practice is to view this data from a variety of more weather web-sites, however, not everyone has access to computers and internet, and a regularly updating forecast heard over the radio would be valuable.

Forecast data is pulled from Yr.no, an open source weather data service. The application is scheduled to update three times daily. On update it pulls the most current forecast data from the service and inserts it into a pre-written script which is then converted to synthetic speech using the Cerecloud API [2]. The resulting audio file is saved to a server location which is configured to be accessed as an RSS feed, allowing the updated audio file to be added to the schedule of station software such as RootIO [5].

[Buna Dimineata] [Sfântu Gheorghe]. Prognoza de [dimineată] până la ora [6:00], astăzi [Cer senin], cu o temperatură de [10] grade, cu vânt de [8,2] metri pe secundă din direcția [est]. Prognoza de [la noapte] de la ora [24] până la ora [06:00], este [Înnorat], cu o temperatură de [10] grade, cu vânt de [5,8] metri pe secundă din direcția [sud-vest]. Prognoza meteo furnizată de aplicația yr.no, a Institutului Meteorologic din Norvegia și a NRK.

Fig. 1. An example weather forecast script with variable values in bold. Translation: *Good morning Sfântu Gheorghe. The forecast for this morning from 6:00 is clear sky, with a temperature of 10 degrees, with wind of 8,2 meters per second in the easterly direction. The forecast for the afternoon from 12:00 to 18:00 is cloudy, with a temperature of 10 degrees, with wind of 5,8 meters per second in the south-west direction. Weather forecast provided by YR.no application, the Norwegian Meteorological Institute and NRK*

¹<https://www.youtube.com/watch?v=yDI5oVn0RgM>

3 LISTENING TEST

In order to assess the viability of the generated content and to better understand potential preferences around broadcasting different types of TTS content, we conducted a listener test using clips of synthesized voices reading out content generated by our synthesized weather feed application. In addition to the weather clips, we also presented listeners with clips of synthesized news and cultural content as an initial effort to explore different TTS voice and content pairings, and for guiding future design decisions with respect to leveraging TTS for community radio.

I looked for it in flowers but their sophisticated geometry was inaccessible to me. I looked for her in love but it was so ephemeral. I have sought her wisely, but I have wandered among many paths. I looked for her in the newborn baby but I forgot the purity. I searched for it in monasteries, but the mystery was terrifying to me.

Fig. 2. A translated excerpt of one of the cultural content scripts. Original taken from <http://casapentrucultura.ro/>

3.1 Methodology

A total of 17 participants, nine female and eight male ages 22 to 56 took part in an online listening test. All were Romanian nationals who had a background in engineering or speech engineering. Four participants, additionally participated in a 15-minute follow-up interview, which were conducted in English. Participants were asked to assess each audio clip in terms of four adjectival descriptors (coherent, suitable for radio, trustworthy, attractive)², as measured on a seven-point Likert scale. All audio and written survey content was presented in Romanian using a modified version of the webMUSHRA audio survey software [9].

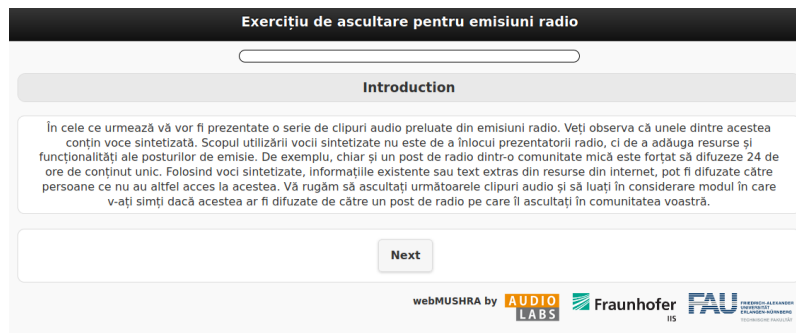


Fig. 3. Introduction to the online listening test. Translation: *In the following you will be presented with a series of audio clips of radio shows. You will notice that some of them contain synthesized voice. The purpose of using synthesized voice is not to replace radio presenters, but to add resources and functionality to radio stations. For example, even a radio station in a small community is forced to broadcast 24 hours of unique content. Using synthesized voices, existing information or text extracted from Internet resources, can be broadcast to people who have no other access to them. Please listen to the following audio clips and consider how you would feel if they were broadcast by a radio station that you listen to in your community.*

Participants rated a total of 15 audio clips, each of which featured ambient radio noise, such as tuning static, in-between clips. The clips distributed represent three content types - news, weather (see figure 1) and culture (see figure 2) - and were presented in a randomized order. Three different TTS voices were included in the test: two SWARA voices

²In Romanian: *coerent adecvat pentru radio, de încredere, atractiv*. - with the Romanian words in italics

[11], including a male (IPS) and a female (BAS) voice; and a high-quality 'characterful' female voice, created by Cereproc (CER) [2]. The two SWARA voices were generated using an HMM-based statistical parametric speech synthesiser, while the Cereproc voice was derived using a proprietary unit selection method that is generally perceived to yield more human-sounding TTS voices.

3.2 Results

3.2.1 Ratings by Content Type. The graph in Figure 4a shows the differences in ratings between clips of different content types. The differences were then analyzed using one-way repeated measure ANOVA tests through the statsmodels package in Python (version 3.7.5). Differences between content type ratings were found to be significant for the descriptors "coherent", "suitable for radio" and "attractive", while differences among "trustworthy" ratings were not statistically significant. A post-hoc analysis comparing ratings by content type was conducted for the descriptors "coherent", "suitable for radio" and "attractive" using pairwise t-tests. Cultural clips were rated as significantly less coherent ($M=4.44$; $SD=1.52$) than news clips ($M=4.88$; $SD=1.52$). Cultural clips were also rated as significantly less suitable for radio ($M=3.94$; $SD=1.62$) than news ($M=5.03$; $SD=1.65$) or weather ($M=4.96$; $SD=1.59$) clips. There was no significant difference between ratings of news and weather on ratings of coherence. Cultural and weather clips were rated as significantly less attractive (cultural $M=3.41$, $SD=1.77$; weather $M=3.73$ $SD=1.64$) than news clips ($M=4.33$ $SD=1.63$).

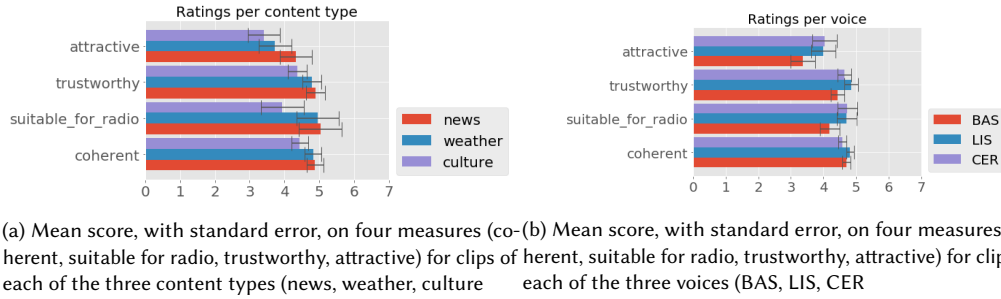


Fig. 4. Results of listening test by content type and voice

3.2.2 Ratings by voice. We additionally used one-way repeated measure ANOVA tests were used to examine the differences in ratings per synthesized voice; the results are shown in Figure 4b. Differences between TTS voices were found to be significant for the descriptors "suitable for radio" and "attractive", while differences for the descriptors "coherent" and "trustworthy" were not statistically significant. Again, a post-hoc analysis comparing ratings by synthetic voice was conducted for the descriptors "attractive" and "suitable for radio" using pairwise t-tests. Clips featuring the BAS voice (i.e. the lower quality SWARA female voice) were rated significantly lower on both attractiveness ($M=3.36$; $SD=1.66$) and suitability for radio ($M=4.19$; $SD=1.76$) than clips featuring the CER (attractive $M=4.03$, $SD=1.80$; suitable $M=4.73$, $SD=1.70$) and IPS (attractive $M=4.0$, $SD=1.67$; suitable $M=4.71$, $SD=1.64$) voices. There was no significant difference distinguishing clips with the CER and IPS voices on measures of attractiveness and suitability for radio.

3.3 Discussion

The results indicate that the category of content being read out by our three synthetic voices can have an impact on the audio clip's perceived coherence, suitability for radio and attractiveness. However, we saw no significant effect of content type on the perceived trustworthiness of audio clips. Cultural content was rated below weather and news in terms of the clips' relative coherence and suitability for radio. With respect to perceived attractiveness, news ranked higher than both weather and culture. The overall acceptability of synthetic speech for broadcasting weather information has been documented in previous work. However, the current study appears to show that though it may be considered acceptable, it may not be considered 'attractive'. Our results further suggest that the use of TTS voices to broadcast news content may be deemed as good as or better than using TTS to broadcast weather information. The lower ratings observed for cultural clips indicates that additional research is needed to explore appropriate uses of synthesized content that go beyond "informative" to conveying more general interest or esoteric types of information.

When interviewed about their radio listening habits, participants identified a wide variety of preferred content (music, current events debates, listener call-in shows on politics, culture, and etc.). However, all four interviewees stated that they stop listening when the programming segues into content that they deem as being uninteresting. P1, for example, explained that they do not listen to political content because "the way politics is covered ... is not relevant to my life right now, I can't relate to it." We expect this dynamic will pose even more challenges for the appropriate design of synthesized radio content.

In terms of the effects the different TTS voices had on our listener judgments, the only significant difference we observed concerned clips read by the female SWARA 'BAS' voice, which participants rated as less attractive and less suitable for radio than the other two TTS voices. In interviews, participants described the BAS voice as high-pitched, "annoying", and featuring unnaturally pauses (P1); and as "too rapid" and "hard to listen to for more than a few minutes."

Overall, participants rated the voices as sufficiently intelligible, as supported by the fact that no significant difference in coherence rating per voice was observed. Interestingly, interviewees provided some subjective and unexpected perceptions of the voices they heard during the listening test. Two of the interviewees reported hearing four or more different voices, while another interviewee made frequent mention of a "baby" or "child" voice, which they found disconcerting.

4 LIMITATIONS AND FUTURE WORK

Given the paucity of studies focused on TTS for broadcasting, and for non-English language use cases in general, this preliminary listener study offered some interesting initial observations on perceptions of long-form synthetic speech as media content. Our future work will focus on further examining the interaction between voice and content type, as we seek to better understand and document best practices in the design of widely acceptable forms of synthesized news and weather content. Additionally, the acceptability and usefulness of this kind of content within the context of a specific community radio stations needs to be examined further through research and design within the given communities.

ACKNOWLEDGMENTS

We gratefully acknowledge the support of ITI/LARSyS and the European Commission's Horizon 2020 Research and Innovation Programme (H2020-ICT-2016-2017-780890).

REFERENCES

- [1] Matthew P. Aylett, Benjamin R. Cowan, and Leigh Clark. 2019. Siri, Echo and Performance: You have to Suffer Darling. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems - CHIEA '19*. ACM Press, Glasgow, Scotland Uk, 1–10. <https://doi.org/10.1145/3290607.3310422>
- [2] Matthew P. Aylett and Christopher J. Pidcock. 2007. The CereVoice Characterful Speech Synthesiser SDK. In *Intelligent Virtual Agents*, Catherine Pelachaud, Jean-Claude Martin, Elisabeth André, Gérard Chollet, Kostas Karpouzis, and Danielle Pelé (Eds.). Vol. 4722. Springer Berlin Heidelberg, Berlin, Heidelberg, 413–414. https://doi.org/10.1007/978-3-540-74997-4_65
- [3] Julia Cambre, Jessica Colnago, Jim Maddock, Janice Tsai, and Jofish Kaye. 2020. Choice of Voices: A Large-Scale Evaluation of Text-to-Speech Voice Quality for Long-Form Content. *Proceedings of the ACM on Human-Computer Interaction* (2020), 13.
- [4] Julia Cambre and Chinmay Kulkarni. 2019. One Voice Fits All?: Social Implications and Research Challenges of Designing Voices for Smart Devices. *Proceedings of the ACM on Human-Computer Interaction* 3, CSCW (Nov. 2019), 1–19. <https://doi.org/10.1145/3359325>
- [5] C. Csikszentmihályi and J. Mukundane. 2016. RootIO: ICT + telephony for grassroots radio. In *2016 IST-Africa Week Conference*. 1–13. <https://doi.org/10.1109/ISTAFRICA.2016.7530700>
- [6] Ewa Luger and Abigail Sellen. 2016. "Like Having a Really Bad PA": The Gulf between User Expectation and Experience of Conversational Agents. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems - CHI '16*. ACM Press, Santa Clara, California, USA, 5286–5297. <https://doi.org/10.1145/2858036.2858288>
- [7] Judith A Markowitz. 2017. Speech and Language for Acceptance of Social Robots: An Overview. *Voice Interaction Design* 2 (2017), 11.
- [8] Blaise Potard, Matthew P. Aylett, David A. Baude, and Petr Motlicek. 2016. Idlak Tangle: An Open Source Kaldi Based Parametric Speech Synthesiser Based on DNN. 2293–2297. <https://doi.org/10.21437/Interspeech.2016-1188>
- [9] Michael Schoeffler, Sarah Bartoschek, Fabian-Robert Stöter, Marlene Roess, Susanne Westphal, Bernd Edler, and Jürgen Herre. 2018. webMUSHRA — A Comprehensive Framework for Web-based Listening Tests. *Journal of Open Research Software* 6, 1 (Feb. 2018), 8. <https://doi.org/10.5334/jors.187> Number: 1 Publisher: Ubiquity Press.
- [10] Kristen M Scott, Simone Ashby, and Julian Hanna. 2020. (accepted) "Human, All Too Human": NOAA Weather Radio and the Emotional Impact of Synthetic Voices. *Proceedings of the ACM on Human-Computer Interaction* (2020), 9.
- [11] Adriana Stan, Florina Dinescu, Cristina Tiple, Serban Meza, Bogdan Orza, Magdalena Chirila, and Mircea Giurgiu. 2017. The SWARA speech corpus: A large parallel Romanian read speech dataset. In *2017 International Conference on Speech Technology and Human-Computer Dialogue (SpED)*. IEEE, Bucharest, Romania, 1–6. <https://doi.org/10.1109/SPED.2017.7990428>

AN EVALUATION OF WORD-LEVEL CONFIDENCE ESTIMATION FOR END-TO-END AUTOMATIC SPEECH RECOGNITION

Dan Oneață¹, Alexandru Caranica¹, Adriana Stan², Horia Cucu¹

¹University POLITEHNICA of Bucharest, Romania

²Technical University of Cluj-Napoca, Romania

ABSTRACT

Quantifying the confidence (or conversely the uncertainty) of a prediction is a highly desirable trait of an automatic system, as it improves the robustness and usefulness in downstream tasks. In this paper we investigate confidence estimation for end-to-end automatic speech recognition (ASR). Previous work has addressed confidence measures for lattice-based ASR, while current machine learning research mostly focuses on confidence measures for unstructured deep learning. However, as the ASR systems are increasingly being built upon deep end-to-end methods, there is little work that tries to develop confidence measures in this context. We fill this gap by providing an extensive benchmark of popular confidence methods on four well-known speech datasets. There are two challenges we overcome in adapting existing methods: working on structured data (sequences) and obtaining confidences at a coarser level than the predictions (words instead of tokens). Our results suggest that a strong baseline can be obtained by scaling the logits by a learnt temperature, followed by estimating the confidence as the negative entropy of the predictive distribution and, finally, sum pooling to aggregate at word level.

Index Terms— Confidence scoring, uncertainty estimation, automatic speech recognition, end-to-end deep learning

1. INTRODUCTION

Reasoning under uncertainty is one of the tenets of intelligence. The first step towards this goal is to endow systems with reliable uncertainty estimates of their predictions. Ideally, the larger the uncertainty the more likely the prediction is erroneous. Alternatively, one can solve the complementary problem of confidence estimation—in this case, the more confident a prediction, the more likely the output is correct.

In the context of automatic speech recognition (ASR) confidence estimation can be of crucial importance for many end-user applications, as it improves the robustness of the systems in safety-critical tasks, helps avoiding errors in human-computer dialogue systems and facilitates manual corrections in audio transcription tasks by flagging the errors. Moreover, previous research has leveraged confidence estimates for a

number of downstream tasks: propagating uncertainties for automatic speech translation [?], selecting confident predictions for self-training [?], manually annotating the less confident predictions for active learning [?].

In this paper we consider confidence estimation for *end-to-end* ASR systems, also known as lattice-free speech recognition [?]. End-to-end models for ASR are gaining traction recently as their performance matches the one of classical ASR and have the additional benefits of being conceptually simple and allowing unified training [?, ?, ?]. However, there is surprisingly little work on confidence estimation for end-to-end speech recognition systems, most of the ongoing research on confidence estimation being carried on computer vision tasks (image classification or segmentation). We believe that there are two main challenges of developing confidence scoring methods for ASR systems: the structured output of the ASR systems and the more granular predictions than what one is usually interested in (*e.g.*, tokens versus words).

ASR systems are structured models (mapping sequences to sequences) as opposed to usual recognition networks (such as, image classification) whose output is a single label. The sequential nature of the output imposes a decoding step, which complicates not only the prediction but also the confidence scoring algorithm, as we need estimate the confidence in an auto-regressive context (the already predicted sequence). For this reason, we fix the predictions based on a pre-trained ASR and apply the confidence scoring methods on top of token probabilities, which are conditioned on the fixed transcript.

In order to enable open vocabulary predictions, end-to-end ASR systems usually use subword tokens to represent the output (byte-pair encoded tokens or even graphemes). However, given that the tokens lack semantics, for many downstream applications we are interested in estimating the confidence of words. To this end, we explore ways of aggregating the token-level uncertainty measures to the larger units, corresponding to words; in fact, the presented techniques can be extended to even coarser predictions, such as sentence or utterance level.

In this context, our main contributions are the following: (i) we adapt several state-of-the-art uncertainty estimation methods to the end-to-end ASR pipeline; (ii) we propose and evaluate aggregation techniques to obtain user-relevant confidence estimates (*i.e.* word-level); (iii) we perform a thorough evalu-

ation on multiple speech benchmark datasets. To the best of our knowledge, this is the first study that provides an in-depth analysis of confidence measures for end-to-end ASR.

2. RELATED WORK

In this section we review two lines of research that are related to our work.

Confidence scoring for speech recognition. Most prior work on confidence scoring for ASR targets classical systems based on the HMM-GMM paradigm. These methods first extract a set of features from the decoding lattice, acoustic or language model, and then train a classifier to predict whether the transcription is correct or not. Typical examples of features include log-likelihood of the acoustic realization, language model score, word duration, number of alternatives in the confusion network [?, ?, ?]. More recently, Swarup *et al.* have augmented the feature set with deep embeddings of the input audio and the predicted text [?], while Errattahi *et al.* have shown that the benefits of domain adaptation on the extracted features [?]. The classifiers employed by the confidence scoring methods range from conditional random fields [?, ?] and multiple layer perceptrons [?] to bidirectional recurrent neural networks [?, ?, ?].

Confidence scoring in end-to-end systems. The baseline method for confidence estimation in neural networks is to use directly the probability of the most-likely prediction [?]. However the neural networks tend to be overconfident and the probability estimates can be improved through temperature scaling [?], which typically leads to better calibration [?, ?]. The most promising direction in terms of simplicity and usefulness involves Monte Carlo estimation: Gal and Ghahramani use dropout at test time to obtain multiple predictions, which are then averaged [?], while Lakshminarayanan *et al.* average the predictions over an ensemble of networks usually trained with different initializations [?]. The latter has been shown to be very reliable on challenging out-of-domain datasets [?], but coming at a high cost [?]. A different approach to confidence scoring is to learn a classifier (typically another neural network) directly on top of the network activations [?, ?].

At the intersection of these two research directions, there is the recent work of Malinin and Gales [?], which similar to us addresses the task of confidence estimation for end-to-end ASR systems. However, they are concerned with token and sentence uncertainty estimation, while we are interested in estimation at word level, and, consequently, provide more focus on the aggregation techniques. Furthermore, they employ ensembles as their primary method of confidence estimation, while we also evaluate temperature scaling and dropout methods. While dropout was previously used for obtaining confidence scores for ASR [?], the method is different from our approach. In [?] the authors generate multiple hypotheses via dropout and then assign confidences to words based on the frequency of their appearances in the aligned hypotheses. In contrast, we

aggregate the posterior probabilities and not the hypotheses, which simplifies the procedure as it avoids the alignment step.

3. METHODOLOGY

This section presents the confidence estimation methodology and proposed ways of improving the them. We first start with a description of the setup and the involved notation.

We consider a sequence-to-sequence model that maps an audio sequence \mathbf{a} to a sequence of tokens $\mathbf{t} = (t_1, \dots, t_T)$. The model is specified by the parameters θ , which are learned by minimizing losses such as the CTC or KL divergence on the training set. At test time the model outputs probabilities for the next token k in an autoregressive manner $p(t_k | \hat{\mathbf{t}}_{<k}, \mathbf{a}; \theta)$ based on the already predicted tokens $\hat{\mathbf{t}}_{<k}$. These probabilities are used for performing decoding via beam search to obtain the most likely sequence of tokens. Given that the conditioned output probability is a distribution over the V tokens in the vocabulary, we denote it by a V -dimensional vector, \mathbf{p}_k .

3.1. Confidence estimation

Our goal is to obtain a confidence score for each word in the output transcript of the ASR. We achieve this in two steps. First, using the posterior probabilities at each time step \mathbf{p}_k , we extract features to encode the confidence score of each token $s_k^{(t)}$. Second, we aggregate the token-level scores into word-level confidence scores $s_j^{(w)}$, based on the word boundaries. Next we detail these two steps; see also figure 1.

Feature extraction. To measure the confidence in a prediction at token level we use two variants:

- Log probability (log-proba) of the most probable prediction given by classifier, that is $s_k^{(t)} = \log \max \mathbf{p}$. This type of feature has been shown to yield a strong baseline for the related tasks of misclassification and out-of-distribution detection [?].
- Negative entropy (neg-entropy) computed over the vocabulary of tokens at each time stamp, that is $s_k^{(t)} = -\mathbf{p}^\top \log \mathbf{p}$. A large entropy means a large uncertainty or, conversely, a large negative entropy implies a confident prediction. While entropy is usually employed as a confidence measure when used with the dropout technique [?], it is by no means restricted to this usage, and can be also applied on the original probabilities.

Aggregation. To obtain word-level features from the token-level ones, we experiment with three types of aggregation functions: sum, average, minimum. Since both proposed features are negative, summing across tokens will result in smaller values and, hence, in lower confidences; this behaviour can be desirable as longer words are more likely to be erroneous (see figure 2). Also, when we sum the log probability of the tokens, we obtain a word-level score corresponding

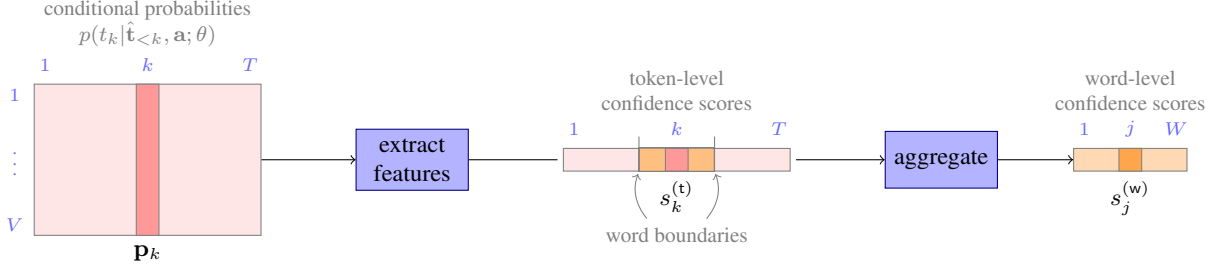


Fig. 1. Overview of the confidence scoring procedure. From an end-to-end ASR system we obtain probabilities \mathbf{p}_k of the k -th token given an utterance \mathbf{a} and previously predicted tokens $\hat{\mathbf{t}}_{<k}$. Based on these probabilities we extract token-level confidence scores $s_k^{(t)}$, which we then aggregate to obtain scores at word level $s_j^{(w)}$. The size of the token vocabulary is denoted by V , the number of tokens is denoted by T and the number of words by W .

to the log probability of the entire sequence. Taking minimum is justified by the fact that we might want a low confidence if at least one of the tokens has low confidence.

3.2. Improving the token probabilities

We propose three ways to make the token probabilities reliable: temperature scaling, dropout and ensembles of models. Our assumption is that by improving the token probabilities, we also improve the word-level scores.

Temperature scaling [?, ?] consists of dividing the logit activations (pre-softmax values) by a scalar τ (known as temperature). The value of τ ranges from zero to infinity and it controls the shape of the distribution: when $\tau \rightarrow 0$ we obtain a uniform distribution, when $\tau \rightarrow \infty$ we obtain a Dirac distribution on the most likely output. Based on τ we update token-level probabilities \mathbf{p} at each time stamp k , as follows:

$$\mathbf{p}'_k = \text{softmax}(\log(\mathbf{p}_k)/\tau). \quad (1)$$

We then extract features $s^{(t)}$ on the updated probabilities \mathbf{p}' , aggregate them into the word-level score $s^{(w)}$ and, finally, classify the word as either correct or incorrect:

$$P(\text{correct}) = \sigma(\alpha \cdot s^{(w)} + \beta). \quad (2)$$

The variables α , β and τ are parameters and are learnt by optimizing the cross-entropy loss on a validation set. The labels are set at word level by aligning at the groundtruth text with the transcription. Note that the parameters α and β are not changing the ranking of the predictions, but allow us to learn a calibrated confidence model.

Dropout [?] is a technique that masks out random parts of the activations in a network, making the network less prone to overfitting. In [?] it has been observed that the dropout induces a probability distribution over the weights of the network and can be consequently used for approximate Bayesian inference. We follow this idea and average the token probabilities obtained over multiple runs of dropout:

$$\mathbf{p}'_k = \frac{1}{N} \sum_n \hat{\mathbf{p}}_k \quad (3)$$

where $\hat{\mathbf{p}}$ specifies the dropout prediction. The updated probabilities are then used to extract either of the uncertainty features (log-proba or neg-entropy).

Ensembles [?] are based on the same idea of averaging predictions from multiple sources, but in this case the set of weights come from independently trained networks (different random seeds used in initialization, batch selection etc.) In our case, we average the token predictions over the models:

$$\mathbf{p}'_k = \frac{1}{N} \sum_n p(t_k | \hat{\mathbf{t}}_{<k}, \mathbf{a}; \theta_n), \quad (4)$$

where $\{\theta_n\}_{n=1}^N$ specifies the ensemble of models. Note that we need to have the same context $\hat{\mathbf{t}}_{<k}$ for all models in the ensemble, so we use the one given by a pre-trained model.

The three presented approaches can be combined; for example, we can first update the probabilities using temperature scaling then average them using dropout. In the experimental section we will evaluate all these combinations.

4. EXPERIMENTAL SETUP

In this section, we describe the datasets used for evaluation, the ASR systems for which we build confidence estimates, and the evaluation metrics.

4.1. Datasets

We have opted for multiple publicly-available and widely-used datasets for our experimental setup.

LibriSpeech [?] is a corpus of approximately 1000 hours of read audiobooks. The data was derived from the LibriVox project and has been carefully segmented and aligned. We use the dataset for both training and evaluation. For training we use the three splits `clean100`, `clean360` and `other500`, while for development and evaluation we use the standard `clean` and `other` splits.

TED-LIUM 2 [?] consists of talks and their transcripts collected from the TED website. We use the dataset for evaluation

Table 1. Size of the datasets (test split) used for confidence estimation evaluation.

dataset	no. utts.	duration
Libri clean	2.6K	5.4 h
Libri other	2.9K	5.3 h
TED	1.1K	2.6 h
CommonVoice	66K	72 h

and consequently employ only the pre-defined dev and test subsets.

CommonVoice [?] is a collaborative dataset of short transcripts that are read by people across the world. There are multiple releases of the dataset and we have used the first release.¹ We use the dataset for evaluation and we defined dev and test subsets by choosing 10% random samples for each of them.

Table 1 presents the test split sizes for each of the evaluation datasets.

4.2. ASR systems

The main ASR system is based on the pre-trained LibriSpeech model provided by the ESPNet toolkit [?]. The model implements the transformer architecture [?] and takes as input 80-dimensional Mel filter banks (extracted with the Kaldi toolkit [?]) and outputs a sequence of tokens. The token vocabulary has dimension 5000 and is obtained by subword segmentation based on a unigram language model [?]. The model is trained on the 960h of the LibriSpeech dataset, which is further augmented using the SpecAugment techniques (time warping, frequency masking, time masking) [?]. For decoding we use a language model, which is also implemented as a transformer and is trained on the LibriSpeech transcriptions and other 14,500 public domain books [?]. The vocabulary of the language model consists of the same 5000 tokens as used by the ASR model.

For the ensemble experiments we re-train the ASR system using the same architecture and data, but different random seeds. We repeat the process four times obtaining four independent models. Due to computational constraints, these models were trained for a shorter number of epochs than the main system (10 versus 120), but we observed that the validation loss function curve began to flatten and that the test performance is reasonable ($5.5\% \pm 0.4$ WER on Libri clean vs 2.7 obtained by the pre-trained model).

4.3. Evaluation metrics

Ideally, we want the confidence score to be correlated with the correctness of the transcription, that is, correct words should

¹https://common-voice-data-download.s3.amazonaws.com/cv_corpus_v1.tar.gz

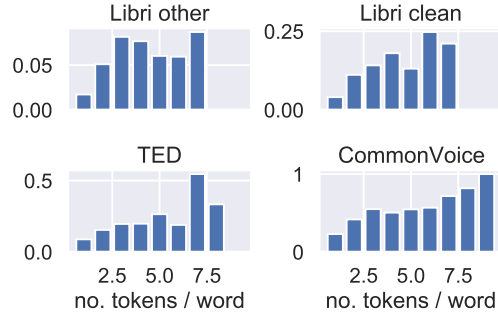


Fig. 2. Fraction of errors as a function of the word length. The fraction of errors is computed as the number of erroneous words divided by the total number of words, while the word length is measured as number of tokens.

have large confidence score, while incorrect ones, low score. Following previous work [?, ?, ?], we employ metrics that are generally used for evaluating binary classifiers, but which have the discrimination threshold varied. More precisely, we measure the area under precision-recall curve (AUPR) and the area under receiver operating characteristic curve (AUROC). However, depending on what we want to focus (errors or correct predictions) we obtain different variants: if we are interested in misclassifications, we will treat the errors as the positive class; on the other hand if we are interested in the correct classification, we will treat the successful detections as the positive class. Hence, for AUPR we use two variants $AUPR_e$ (when errors are treated as positives) and $AUPR_s$ (when successes are treated as positives). For AUROC the same value is obtained for either choice, so there is no need to make this distinction.

5. RESULTS AND DISCUSSION

This section presents the experimental results. We start with an evaluation of features and their aggregations (§5.1), and then report results for the improved variants involving temperature scaling, dropout (§5.2) and ensembles (§5.3). We conclude the section with a discussion.

5.1. Features and aggregation

We evaluate the proposed uncertainty features and aggregation techniques on the four datasets described in subsection 4.1. We use the pre-trained model to obtain text predictions for all the audio files in the test split of each dataset, and then estimate the confidence based on the methodology described in subsection 3.1. Table 2 presents the results for all combinations of features and aggregations.

Comparison of features. We observe that log probability features outperform the entropy features across all settings

Table 2. Confidence scoring results for combinations of features and aggregations on the four test splits. For all three metrics reported metrics (AUPR_e, AUPR_s, AUROC) larger values are better. We indicate the word error rate of the pre-trained ASR system on each of the dataset by the figures on the right of the name.

feat.	agg.	Libri clean / 2.7%			Libri other / 6.0%			TED / 13.3%			CommonVoice / 28.6%		
		AUPR _e	AUPR _s	AUROC	AUPR _e	AUPR _s	AUROC	AUPR _e	AUPR _s	AUROC	AUPR _e	AUPR _s	AUROC
1 log-proba	sum	21.55	99.21	82.41	29.99	98.10	81.75	39.97	95.88	79.95	48.98	77.71	64.84
2 log-proba	min	21.85	99.19	82.47	28.64	98.06	81.66	39.74	95.94	80.58	46.79	76.74	62.67
3 log-proba	avg	20.12	99.10	80.90	26.72	97.93	80.47	38.74	95.88	80.29	44.51	75.82	60.87
4 neg-entropy	sum	17.31	99.10	79.97	26.37	97.86	79.58	34.96	95.41	77.57	47.71	77.10	63.74
5 neg-entropy	min	19.94	99.09	80.55	26.75	97.82	79.64	37.55	95.56	79.01	45.51	76.00	61.21
6 neg-entropy	avg	17.55	98.95	77.72	24.26	97.59	77.46	36.28	95.42	78.29	42.64	74.83	58.75

(aggregations and datasets). The only notable exception is the CommonVoice dataset where the results are comparable.

Comparison of aggregations. Generally, the sum aggregation works better with log-proba features, while the min aggregation works better for entropy features. The sum might not be well suited for entropy features because their magnitude is larger than for log-proba and the word confidence gets penalized too much by the length; but, as we will see further, this behaviour can be alleviated by temperature scaling. Averaging is generally underperforming for both features, suggesting that length-invariant measures are detrimental. Indeed, a closer look at the frequency of errors with the length size indicates that the more tokens a words has the more likely is that is incorrect, see figure 2.

Comparison across datasets. As expected the pre-trained model performs best on in-domain data (2.7% WER on Libri clean and 6.0% on Libri other), the performance then dropping sharply as we evaluate on out-of-domain data (13.3% on TED and 28.6% on CommonVoice). In each of these settings the number of words that are correctly classified change, going from more on the Libri splits to fewer on TED and CommonVoice. This observation explains why the performance for AUPR_s drops as a function of the domain of the data, and, conversely, why the AUPR_e performance improves. Unfortunately, for this exact reason—the different performance of the base ASR system on the four datasets—it is impossible to compare the confidence methods across datasets, as they use a different groundtruth [?].

5.2. Temperature scaling and dropout

We benchmark the confidence scoring method after improving the token probabilities by two of the described techniques: temperature scaling and dropout. We use the pre-trained ASR system and report results only on the TED test set. The parameters for temperature scaling method are learnt on the dev split of the TED dataset for each setting of feature and aggregation. When temperature scaling is combined with dropout we first apply the temperature scaling (using the same temperature) and the follow with the aggregation over dropout. The dropout

Table 3. Confidence scoring results on the TED test set for combinations of features, aggregations and their improved variants – temperature scaling (TS) and dropout (D). The bullet sign • indicates whether a variant is employed. Bold results indicate the best results for the feature-aggregation combination; these results show that using both temperature scaling and dropout yields the best results.

feat.	agg.	TS	D	AUPR _e	AUPR _s	AUROC
1	log-proba	sum		39.97	95.88	79.95
2			•	41.41	96.81	82.78
3			•	40.92	96.19	81.11
4			•	42.99	97.14	84.10
5	log-proba	min		39.74	95.94	80.58
6			•	42.08	96.94	83.76
7			•	39.84	95.98	80.74
8			•	42.17	97.00	83.93
9	log-proba	avg		38.74	95.88	80.29
10			•	41.19	96.95	83.73
11			•	38.97	95.99	80.66
12			•	41.32	97.06	84.08
13	neg-entropy	sum		34.96	95.41	77.57
14			•	33.14	96.22	79.45
15			•	42.16	96.91	83.50
16			•	43.59	97.62	85.51
17	neg-entropy	min		37.55	95.56	79.01
18			•	38.75	96.53	81.98
19			•	41.23	96.87	83.50
20			•	42.23	97.60	85.51
21	neg-entropy	avg		36.28	95.42	78.29
22			•	38.01	96.51	81.85
23			•	40.22	96.53	82.48
24			•	41.15	97.43	85.18

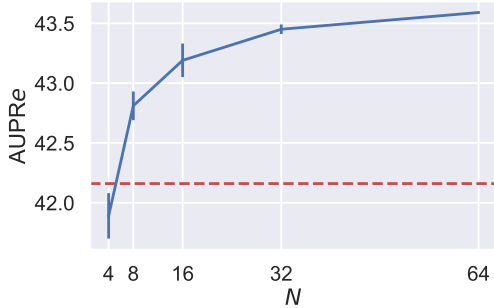


Fig. 3. AUPRe performance as a function of the number of dropout runs on the TED test set. The horizontal red line indicates the performance of the model without dropout. The model uses neg-entropy features, sum aggregation and temperature scaling.

Table 4. Confidence scoring results on the TED test set for combinations of temperature scaling (TS), dropout (D) and ensembles (E), using neg-entropy as features and sum as aggregation. The bullet sign • indicates whether a variant is employed.

	TS	D	E	AUPRe	AUPRs	AUROC
1				28.58	95.30	75.79
2	•			32.00	96.32	79.47
3		•		27.49	95.51	75.67
4			•	30.89	96.26	78.89
5	•	•		31.10	96.40	79.06
6	•		•	34.57	96.95	81.64
7		•	•	28.94	96.26	77.93
8	•	•	•	33.00	96.84	80.82

method averages 64 independent predictions. Table 3 presents the results for all combinations of features and aggregations and improvement techniques.

The results indicate that both proposed methods improve the results as is their combination, which gives overall the best result. We observe that log-proba features benefit more from dropout, while the neg-entropy feature yield more improvements when temperature scaling is used. Interestingly, the best results are now obtained for the neg-entropy with sum aggregation (row 16). Figure 3 shows that the dropout performance improves with the number of runs and plateaus around the chosen value of 64.

5.3. Ensembles

We present results for confidence scoring using ensembles of models and their combinations with the other improved

versions (temperature scaling and dropout). For each of the retrained models from the ensemble we use the predictions of the pre-trained model to select the transcription; the retrained model is just used for confidence scoring, by extracting the confidence features described previously. The results are presented in table 4. For the rows that do not use ensemble (rows 1, 2, 3 and 5) we evaluate each of the four single models independently and report the mean performance.

The pre-trained model (table 3, row 13) has generally a better performance the retrained ones (table 4 row 1), suggesting that the predictive performance of a model can correlate with its confidence scoring performance.

Among the three improvement methods, we note that temperature scaling gives the largest performance boost on all three metrics (row 2). Surprisingly, the dropout method improves only the AUPRs performance over the baseline (row 3). On combinations of two methods, temperature scaling and ensemble complement each other and obtain better performance.

5.4. Discussion

We briefly discuss different perspectives on our work.

Augmenting the feature set. The benchmarked methods have the benefit of being general, as we leverage the posterior token predictions, which are readily available in most, if not all, existing end-to-end ASR toolkits. However, the feature set could be extended with prior probabilities on the input audio or the generated text, or with duration information extracted from the attention weights.

Learning features. Following [?, ?], we have also experimented with learning a confidence scoring network on top of features extracted from the end-to-end model (specifically, logits and pre-logits activations). However, our experiments failed to show improvements over the presented results.

Dealing with deletions. To generate confidence scoring groundtruth we align the reference text to the predicted text and mark the correct words in the predicted text as positives and the substitutions and insertions as negatives. This approach is typical in the confidence scoring literature, but it misses the errors made by deleting words. Several works have addressed this problem [?, ?], and we leave it to future work to extend our approach for this task.

6. CONCLUSIONS

This paper presented an approach for word-level confidence scoring in end-to-end speech recognition systems. We carried a thorough ablation study on features and their aggregation on three well-known speech databases (LibriSpeech, TED-LIUM and CommonVoice) and further evaluated improved methods, which modify the token probabilities, and their combinations. Our main observation is that temperature scaling improves both features (max probability or entropy) and dropout or ensemble methods. Lastly, using a pre-trained model improves

the replicability and allows comparison with other confidence scoring models that will use the same ASR.

7. ACKNOWLEDGEMENT

The work has been funded by the Operational Programme Human Capital of the Ministry of European Funds through the Financial Agreement 51675/09.07.2019, SMIS code125125.