THE PUBLISHING HOUSE
OF THE ROMANIAN ACADEMY

PROCEEDINGS OF THE ROMANIAN ACADEMY, Series A,
Volume 21, Number 4/2020, pp. 395–402

TOWARDS A ROMANIAN END-TO-END AUTOMATIC SPEECH RECOGNITION BASED ON DEEPSPEECH2

Andrei-Marius AVRAM, Vasile PAIŞ, Dan TUFIŞ

Research Institute for Artificial Intelligence, Romanian Academy
Corresponding author: Dan Tufiş, E-mail: tufis@racai.ro

Abstract. This paper presents an implementation of an ASR system for the Romanian language that uses a multi-layer neural network architecture to transcribe the input speech, augmented with a statistical language model to improve the correctness of the transcriptions. The neural model was trained on 230 hours of speech for 70 epochs and, together with the language model, it achieved a word error rate of 9.91% and a character error rate of 2.81%. We further tested the response time of the ASR and we obtained an average latency of 70 ms, while running on a GPU, which can be suitable for near real-time transcriptions.

***Key words*:** automatic speech recognition, Romanian, natural language processing, deep neural networks.

1. INTRODUCTION

Automatic Speech Recognition (ASR) consists in translating human spoken utterances into a textual transcript, and it is a key component in voice assistants such as Apple's Siri, Amazon's Alexa, Microsoft's Cortana or Google's Assistant (Sainath et al. [23], Lopatovska et al. [25]), in spoken language translation systems (Di Gangi et al. [8]) or in generating automatic transcriptions for audio and videos (Noda et al. [27]). Most of the ASR systems before the deep learning revolution used variations of Hidden Markov Models (HMM) (Garg et al. [10]), and although they achieved good Word Error Rates (WER), they became very slow for large vocabularies and could not be used for open domain real-time transcriptions.

When AlexNet (Krizhevsky et al. [21]) won the ILSVRC-2012 competition (Russakovsky et al. [31]) by a large margin[1] of approximately 10% with deep neural networks, a lot of research effort emerged in this direction and many deep models pushed the state-of-the-arts (SOTA) in various areas like computer vision (He et al. [15]), natural language processing (Brown et al. [4]) or speech recognition (Collobert et al [5]). One of the models that stood out was also DeepSpeech2 (Amodei et al. [1]), a model that was used to automatically transcribe both English and Mandarin, achieving a WER under 10% for both languages on most of the tested datasets. To attain these results, they trained a 11 layered deep neural network on 11,940 of speech hours for English and on 9,400 of speech hours for Mandarin, and they also used beam search to increase the decoding performance, together with a n-gram model to correct the transcriptions.

In the context of the ROBIN[2] project, an ASR system for Romanian language was needed to allow human interaction with a robot by using voice. ROBIN is a complex project, user centered, aiming to develop software and services for interaction with assistive robots and autonomous vehicles (Tufiş et al. [35]). The envisaged human-robot dialogues are well defined, based on a closed-world scenario, controlled by a Dialog Manager (DM) described in details (Ion et al. [19]). The ASR component presented there, was replaced by the one presented here, with much better accuracy and response time.

Previous work (Georgescu et al. [13]) considered the application of neural networks to Romanian ASR systems using the Kaldi[3] toolkit. However, as reported in the paper, the internal system model is comprised of many neural layers which may have an impact on runtime speed. Since in the ROBIN project we were

[1] http://image-net.org/challenges/LSVRC/2012/results.html
[2] http://aimas.cs.pub.ro/robin/en/
[3] https://kaldi-asr.org

interested in a near real-time system we considered an approach with less neural layers and a different architecture. Nevertheless, we show in the Evaluation section that our approach produces results comparable to those presented in the paper of Georgescu et al. [13].

The rest of the work is structured as follows. In section 2, we present the datasets used for training the speech recognition model and the language model. In section 3, we outline the architecture of the system and in section 4 we present the experimental setup for training, and the results obtained by the system with different configurations. Finally, we draw conclusions and present the future work in section 5.

2. DATASETS

2.1. Speech Datasets

For training the ASR model, high quality alignment of speech to text was needed. For this purpose, the main audio resource used was the speech component of the representative corpus of contemporary Romanian language (CoRoLa). CoRoLa has been jointly developed, as a priority project of the Romanian Academy, by two institutions: Research Institute for Artificial Intelligence "Mihai Drăgănescu" (from Bucharest) and the Institute of Computer Science (from Iași) The oral texts in CoRoLa are mainly professional recordings from various sources (radio stations, recording studios). They are accompanied by the written counterpart: the transcription either from their provider or made by the project partners. Therefore, different principles applied in their transcription.

Another part of the oral corpus is represented by read texts: read news in radio stations, texts read by professional speakers recorded in studios, and extracts from Romanian Wikipedia read by non-professionals, by volunteers, recorded in non-professional environments. In their case, the written component is provided by the sources, or was collected by the project partners (Mititelu et al. [26]).

The speech component of the CoRoLa corpus can be interrogated by means of the Oral Corpus Query Platform (OCQP)[4]. This allows searching for words and listen to their spoken variant, based on the alignment between text and speech (Tufiș et al. [36]).

In the context of the RETEROM project[5], the CoBiLiRo platform (Cristea et al. [7]) was built to allow gathering of additional bimodal corpora with one of the final goals being to enrich the CoRoLa corpus. Thus, additional corpora with speech and text alignments were considered. This includes: Romanian Digits (RoDigits) (Georgescu et al. [11]), Romanian Common Voice (RCV) (Ardila et al. [2]), Romanian Speech Synthesis (RSS) (Stan et al. [33]), Romanian Read Speech Corpus (RSC) (Georgescu et al. [12]).

The RoDigits corpus contains 37.5 hours of spoken connected digits from 154 speakers[6] whose ages vary between 20 and 45. Each speaker recorded 100 clips of 12 randomly generated Romanian digits, and after the semi-automated validation, the final corpus contained 15,389 of audio files.

The common voice corpus is a massively multilingual dataset of transcribed speech that, as of October 2020, contains over 7,200 hours of transcribed audio in 54 languages from over 50,000 speakers. The Romanian version is one of the recently added languages and its corresponding corpus contains 7 hours of transcribed audio recorded by 79 speakers, from which only 5 hours are validated. The corpus sentences were collected from Wikipedia using a sentence collector, and each sentence must be approved by two out of three reviewers before reaching the final version of the corpus.

The RSS corpus was designed for speech synthesis and it contains 4 hours of speech from a single female speaker using multiple microphones. The speaker read 4000 sentences that were extracted from novels, newspapers chosen for diphone coverage and fairytales. RSS was also extended with over 1700 utterances from two new female speakers, comprising now 5.5 hours of speech.

RSC is a publicly available speech corpus for the Romanian language, comprising 100 hours collected from 164 native speakers, mainly students and staff of the Faculty with an age average of 24 years. Out of the 133,616 files, approximately 100k audio files have a duration under 2.5 seconds, these being isolated word utterances, while only less than 200 have a duration of more than 15 seconds. The sentences were

[4] http://corolaws.racai.ro/corola_sound_search/index.php
[5] http://www.racai.ro/p/reterom/
[6] Almost all the speakers were Romanians native, except for one Albanian native.

selected from novels in Romanian, online news and from a list of words that covered all the possible syllables in Romanian.

After gathering samples from all the available Romanian speech resources aligned with their corresponding text versions, the final dataset used for training the ASR system consisted of 230 hours of audio.

2.2. Text Datasets

Additional to the speech data, needed for the actual ASR system training, more text resources were needed to train a language model able to correct recognition errors originating within the ASR system. For this reason, we considered two large corpora of Romanian texts, each with its own characteristics, as described in the following paragraphs.

The CoRoLa corpus is a large, growing, collection of contemporary Romanian texts, currently containing 941,204,169 tokens. Various annotation levels were employed and the corpus can be queried through various interfaces (Cristea et al. [6]), including KorAP (Banski et al. [3]). CoRoLa statistics, regarding the available domains and styles are available in Tufiș et al. [36].

OSCAR is an open-source huge multilingual corpus that was obtained by filtering the Common Crawl[7] and by grouping the resulting text by language. The Romanian version contains approximately 11 GB of deduplicated shuffled sentences. Even though CoRoLa is a representative corpus of the Romanian language, we considered that adding even more text to the training of a language model could benefit in terms of accuracy. The impact of this addition is further investigated in section 4, below.

Because the language model was very sensitive to errors in text, we further cleaned each corpus, obtaining 4.1 GB of text from CoRoLa and 6.1 GB of text from OSCAR. The main cleaning steps that we applied were the following:

- Removed all lines that did not have a minimum length of 20 characters because the short sentences were usually titles or references.
- Removed all lines that did have an average word length higher than 14, because in the OSCAR corpus words can be concatenated, resulted from missing spaces.
- Removed lines that did not have diacritics or did not use the correct character codes. This rule was considered due to words being written incorrectly, without or with wrong codes for diacritics, especially in the OSCAR corpus, even though it is possible to have correct sentences without any words with diacritics, although it is not probable.
- Removed all lines that were not detected as Romanian[8].
- Removed all lines that had “?” or “!” inside a word because they could have been replacement for diacritics or crawling artefacts.
- Removed the lines that had over 30% digits from the total length.
- Replaced all the letters “î” that are inside a word and that do not have a prefix before them with “â”, in order to improve the correctness of the language model.

3. SYSTEM ARCHITECTURE

3.1. Speech to Text Model

From an architectural point of view, the model we developed followed very closely the original DeepSpeech2, with only small differences in the number of layers and the number of parameters, scaled down to match the size of the data. To generate the input of the model, we split the speech into fixed-sized windows of 20 ms and computed the Mel-frequency cepstral coefficients (MFCC) (Logan et al. [24]) on each. The spectrograms are then feed into the two convolutional layers (LeCun et al. [22]) with 32 filters: the first one with a kernel size of (41, 11) and a stride of (2, 2), and the second one with a kernel size of (21, 11) and a stride of (2, 1). The resulting sequence of filter maps is then processed by four bidirectional long short-

[7] https://commoncrawl.org/

[8] We used *langdetect* to detect Romanian sentences: https://github.com/Mimino666/langdetect.

term memory (LSTM) (Hochreiter et al. [17]) layers that have 768 units. The output of the LSTM layers is then fed into a lookahead layer (Amodei et al. [1]) that learns the activation of each neuron t steps into the future. Finally, using a fully connected layer, each output in the sequence is projected into a vector of size 33, that represents a distribution of probabilities over the Romanian characters, together with the space character and the blank index[9]. To make the training more stable and the model to converge faster, we also use batch normalization after each layer except the last one (Ioffe et al. [18]).

Because the utterance of a character may take more than the size of a window (20 ms), the resulting sequence of characters is usually repetitive, we use the Connectionist Temporal Classification (CTC) (Graves et al. [14]) loss to train the network. The CTC loss takes the sum of all the possible alignments of the ground-truth text by collapsing the repeated characters from the sequence outputted by the neural model that are not marked by a blank index. The architecture of the neural model is further depicted in Fig. 1.

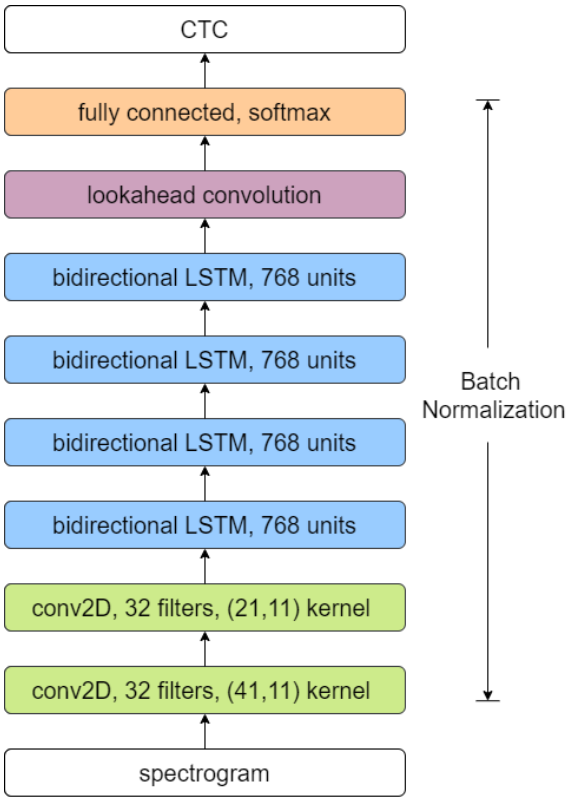

Fig. 1 – The neural network architecture used in the ASR system.

3.2. Language Model

To further improve the performance of the system, we used a language model that helps the neural network to assign higher probabilities to the more likely words, given the previous n words. As recommended by Amodei et al. [1], we choose the Kneser-Ney smoothing method of the 5-gram language model that was trained using the KenLM (Heafield et al. [16]) toolkit[10] on cleaned text, as described in section 2.2. The language model implements two data structures: PROBING for improving the query speed, and TRIE that reduces the memory costs. The KenLM toolkit allows the configuration of two hyperparameters: alpha – that controls the contribution of the model to predicting a word, and beta – the probability of inserting a new word into the sequence. Moreover, the toolkit allows the pruning of the model to reduce its size, so we pruned it by removing the 2-grams, 3-grams, 4-grams and 5-grams that had less than 2 appearances, reducing the memory cost with over 40%.

[9] The blank index ("_" in the vocabulary) is a special character that solves the repeated reoccurrence problem of the data. For example, "acceptat" is encoded as "ac_ceptat" to mark not to collapse the character "c" while decoding.

[10] https://github.com/kpu/kenlm

The language model was used simultaneously with a beam search decoding, that instead of greedily choosing the most likely character to decode, it expands all the possible next characters and keeps only the k most likely.

3.3. Deployment

We deployed the model as a REST web service, using the Flask framework[11] and the Waitress web server[12], that accepts POST requests on the "/transcribe" method, using a "file" parameter. The method will invoke the ASR model and the language model, will transcribe the attached wav file and will return a JSON with two fields: the status of the call and the corresponding transcription or, in case of an exception, an error message. The service is also configurable and allows the tuning of the system parameters such as the beam width, whether to use or not the language model, or whether to use the GPU or only run on the CPU. The raw output from the ASR system was further post-processed to improve readability. This involves capitalizing the first letter of words present on a known named entity list and adding hyphens to construct correct Romanian words. The raw output of the ASR system does not take into account hyphens. Thus, expressions like "mi-am" or "s-au" are recognized as "miam" or "sau". In some cases, such as "miam" vs "mi-am" the correction may seem trivial since "miam" is not a Romanian word. However, in the case of "sau" vs "s-au" the correction process is more complex since both are valid Romanian words and the decision regarding a certain form must consider the context.

Bigram and unigram models were trained on the CoRoLa corpus considering both words with hyphens and words without hyphens. To reduce the model's size, for models containing words without hyphens, we considered only words allowing also a form with a hyphen. Then, for text correction we first determine from the bigram model the frequencies of the word in context with hyphen and without. If the frequencies of apparition are equal, we look in the unigram model to identify the most frequent form. To reduce the computation time required we consider only windows starting with the current word (the one that may need correction). Furthermore, since in some cases there may be several hyphenation variants associated with a certain word (either because they are valid or because of errors in the corpus), we considered all of these forms when determining the frequencies. The final algorithm is presented in Fig. 2. The actual implementation further optimizes on this by computing the frequencies only when they are actually needed and makes use of hash tables to efficiently query the models.

1. For each word W_k in ASR output (without hyphen)
1.1. If W_k allows one or more forms with hyphen
1.1.1. For each form with hyphen H_j associated with W_k
1.1.1.1. $FH2_j$=Freq(H_j, W_{k+1}), $FH1_j$=Freq(H_j)
1.1.2. FH2=max($FH2_j$), FW2=Freq(W_k,W_{k+1})
1.1.3. FH1=max($FH1_j$), FW1=Freq(W_k)
1.1.4. If FH2>FW2 then correct W_k using form H_j, j=argmax($FH2_j$)
1.1.5. If FH2==FW2 and FH1>FW1 then correct W_k using form H_j, j=argmax($FH1_j$)

Fig. 2 – Hyphen restoration algorithm

The hyphen restoration algorithm's implementation was also exposed as a REST web service with a single method, "/correct", receiving the unhyphenated text and returning the corrected, hyphenated, form. The implementation is available in GitHub[13]. The ASR system, together with the post-processing algorithm, were further integrated in the RELATE platform[14] (Păiș et al. [30]), allowing users to upload a recorded wav file or make a recording directly in the platform and run it through the ASR system. The recognized text can then be analysed using the available annotation mechanisms within the RELATE platform.

[11] https://flask.palletsprojects.com/en/1.1.x/
[12] https://docs.pylonsproject.org/projects/waitress/en/stable/
[13] https://github.com/racai-ai/RobinASRHyphenationCorrection
[14] https://relate.racai.ro/index.php

4. EVALUATION

The audio input was resampled to 16 kHz, split into windows of 20 ms and then, the MFCC were extracted from each window. To reduce the bias, we augmented the sound waves by adding random tempo and gain perturbations, and the spectrograms with simple spectral augmentation techniques, as described in (Park et al. [28]). The waves to be augmented were selected randomly, in each independent batch, with a probability of 20%.

The model was trained for 70 epochs with a batch size of 80 that occupied around 10 GB of RAM memory. We used the Adam optimizer (Kingma et al. [20]) with a high learning rate of $2e{-}4$ used to accelerate the training, and a learning rate decay of 5% after each epoch that helps the model converge to a local minimum and avoid oscillation (You et al. [37]). Because recurrent neural networks usually suffer from exploding or vanishing gradients in long sequences, we clip the gradients whose norm exceed 400, as described in (Pascanu et al. [29]). The final training dataset was created by combining the 10 datasets described in Section 2. We further removed the audio files whose length exceeded 25 seconds and split the dataset into a train set, a validation set (5000 samples) and a test set (5000 samples).

Because the default alpha and beta were not optimal for our language model, we used a grid search to find the best values, for alpha in the [0, 1.5] interval and for beta in the [0, 3] interval, with discrete steps of 0.1 for each parameter. For the beam search, we used a beam width of 128 that ran on 4 threads.

We evaluated the model on the test set that contained 5000 samples extracted randomly from the 10 datasets, and it obtained a WER of 15.572% and a CER of 4.524%. By using the KenLM trained on the concatenated CoRoLa and OSCAR corpus, with the default parameters for alpha (0.6) and beta (0.7), we managed to improve the WER by over 5.6% and the CER by over 0.9%. Finally, by optimizing the parameters of the language model with grid search (α=0.32 and β=1,65), the system obtained a final performance of 9.91% WER and 2.81% CER. We further show the results obtained by the DeepSpeech2 model, with different combinations of optimal/non-optimal language models, in Table 1.

We also tested the response time of the ASR and we noticed an improvement from an average response rate of approximately 600 ms to an average response rate of approximately 70 ms when we switched from using an Intel i7-7700K CPU to using a NVIDIA 1080 Ti GPU.

Table 1

DeepSpeech2 performance by using various KenLMs

	Non-Optimized		**Optimized**	
Model	**WER**	**CER**	**WER**	**CER**
DeepSpeech2	15.572	4.524	-	-
DeepSpeech2+KenLM$_{\text{CoRoLa}}$	10.467	3.685	10.450	2.961
DeepSpeech2+KenLM$_{\text{OSCAR}}$	10.753	3.795	10.666	2.898
DeepSpeech2+KenLM$_{\text{CoRoLa+OSCAR}}$	**9.916**	**3.614**	**9.911**	**2.809**

5. CONCLUSION

The paper presented a Romanian end-to-end automatic speech recognition system based on the DeepSpeech2 architecture, achieving a best score of 9.91% WER and 2.81% CER, with an average latency of 70 ms. The achieved runtime latency combined with the overall performance makes the ASR system suitable for deployment within the ROBIN project, for human-robot interaction. The code was open-sourced and is available at the following link: https://github.com/racai-ai/RobinASR and complements the previous open-sourced release of the dialog manager (Ion et al. [19]): https://github.com/racai-ai/ROBINDialog, while a web interface is available within the RELATE platform.

One limitation of the current system is that the length of the wav file is recommended to be under 25 seconds due to the bias introduced during training. In order to solve it, we plan to segment the input wav

file based on silence and send to the neural model sequences of parts of the audio signal, concatenating the predicted transcriptions in the end. Also, currently the web server accepts only files in wav format, with the recommended frequency of 16 kHz, mono, 16-bit. We intend to make it more flexible by introducing more formats like mp3 or mp4.

Additionally, even though the initial context for our research was offered by the human-robot interaction in the context of well-defined micro-worlds, we envisage the development of the ASR system towards a more general usage scenario, while keeping the low latency achieved in these experiments.

ACKNOWLEDGMENTS

This work was realized in the context of the ROBIN project, a 38 months grant of the Ministry of Research and Innovation PCCDI-UEFISCDI, project code PN-III-P1-1.2-PCCDI-2017-734 within PNCDI III.

REFERENCES

1. D. AMODEI, S. ANANTHANARAYANAN, R. ANUBHAI, J. BAI, E. BATTENBERG, C. CASE, J. CASPER, B. CATANZARO, Q. CHENG, G. CHEN, J. CHEN, *Deep speech 2: End-to-end speech recognition in English and Mandarin*, International conference on machine learning, New York City, USA, June 2016, pp. 173–182.
2. R. ARDILA, M. BRANSON, K. DAVIS, M. HENRETTY, M. KOHLER, J. MEYER, R. MORAIS, L. SAUNDERS, F.M. TYERS, G. WEBER, *Common voice: A massively-multilingual speech corpus*, arXiv preprint, arXiv:1912.06670, 2019.
3. P. BAŃSKI, P. FISCHER, E. FRICK, E. KETZAN, M. KUPIETZ, C. SCHNOBER, O. SCHONEFELD, A. WITT, *The new IDS corpus analysis platform: challenges and prospects*, Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC 2012), pp. 2905–2911.
4. T.B. BROWN, B. MANN, N. RYDER, M. SUBBIAH, J. KAPLAN, P. DHARIWAL, A. NEELAKANTAN, P. SHYAM, G. SASTRY, A. ASKELL, S. AGARWAL, *Language models are few-shot learners*, arXiv preprint, arXiv:2005.14165, 2020.
5. R. COLLOBERT, C. PUHRSCH, G. SYNNAEVE, *Wav2letter: an end-to-end convnet-based speech recognition system*, arXiv preprint, arXiv:1609.03193, 2016.
6. D. CRISTEA, N. DIEWALD, G. HAJA, C. MĂRĂNDUC, V.B. MITITELU, M. ONOFREI, *How to find a shining needle in the haystack. Querying CoRoLa: solutions and perspectives*, Revue Roumaine de Linguistique, **LXIV**, *3*, pp. 279–292, 2019.
7. D. CRISTEA, I. PISTOL, Ş. BOGHIU, A.D. BIBIRI, D. GÎFU, A. SCUTELNICU, M. ONOFREI, D. TRANDABĂŢ, G. BUGEAG, *CoBiLiRo: A research platform for bimodal corpora*, Proceedings of the 1st International Workshop on Language Technology Platforms (IWLTP 2020), pp. 22–27; Language Resources and Evaluation Conference (LREC 2020), Marseille, May 11-16, 2020.
8. M.A. DI GANGI, M. NEGRI, M. TURCHI, *Adapting Transformer to end-to-end spoken language translation*, Interspeech, pp. 1133–1137, Graz, Austria, 2019.
9. S.D. DUMITRESCU, T. BOROŞ R. ION, *Crowd-sourced, automatic speech-corpora collection – Building the Romanian Anonymous Speech Corpus*, Collaboration and Computing for Under-Resourced Languages (CCURL) in the Linked Open Data Era, Reykjavik, Iceland, pp. 90–94, 2014.
10. A. GARG, P. SHARMA, *Survey on acoustic modeling and feature extraction for speech recognition*, 3rd International Conference on Computing for Sustainable Global Development, New Delhi, India, 2016, pp. 2291–2295.
11. A.L. GEORGESCU, A. CARANICA, H. CUCU, C. BURILEANU, *Rodigits – A Romanian connected-digits speech corpus for automatic speech and speaker recognition*, University Politehnica of Bucharest Scientific Bulletin, Series C, **80**, *3*, pp. 45–62, 2018.
12. A.L. GEORGESCU, H. CUCU, A. BUZO, C. BURILEANU, *RSC: A Romanian Read Speech Corpus for automatic speech recognition*, Proceedings of the 12th Language Resources and Evaluation Conference, Marseille, France, 2020, pp. 6606–6612.
13. A. GEORGESCU, H. CUCU, C. BURILEANU, *Kaldi-based DNN architectures for speech recognition in Romanian*, Proceedings of the International Conference on Speech Technology and Human-Computer Dialogue (SpeD), Timisoara, Romania, 2019, pp. 1–6, DOI: 10.1109/SPED.2019.8906555.
14. A. GRAVES, S. FERNÁNDEZ, F. GOMEZ, J. SCHMIDHUBER, *Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks*, Proceedings of the 23rd International Conference on Machine Learning, Pittsburgh, United States, 2006, pp. 369–376.
15. K. HE, X. ZHANG, S. REN, J. SUN, *Deep residual learning for image recognition*, Proceedings of the IEEE conference on computer vision and pattern recognition, Las Vegas, USA, 2016, pp. 770–778.
16. K. HEAFIELD, *KenLM: Faster and smaller language model queries*, Proceedings of the Sixth Workshop on Statistical Machine Translation, Edinburgh, Scotland, 2011, pp. 187–197.
17. S. HOCHREITER, J. SCHMIDHUBER, *Long short-term memory*, Neural computation, **9**, *8*, pp. 1735–1780, 1997.
18. S. IOFFE, C. SZEGEDY, *Batch normalization: Accelerating deep network training by reducing internal covariate shift*, International Conference on Machine Learning, 2015, pp. 448–456.
19. R. ION, V.G. BADEA, G., CIOROIU, V.B. MITITELU, E. IRIMIA, M. MITROFAN, D. TUFIŞ, *A dialog manager for micro-worlds*, Studies in Informatics and Control, **29**, *4*, 2020 (in print),
20. D.P. KINGMA, J. BA, *Adam: A method for stochastic optimization*, arXiv preprint, arXiv:1412.6980, 2014.

21. A. KRIZHEVSKY, I. SUTSKEVER, G.E. HINTON, *Imagenet classification with deep convolutional neural networks*, Advances in Neural Information Processing Systems, Harrahs and Herveys, USA, 2012, pp. 1097–1105.
22. Y. LECUN, L. BOTTOU, Y. BENGIO, P. HAFFNER, *Gradient-based learning applied to document recognition*, Proceedings of the IEEE, **86**, *11*, pp. 2278–2324, 1998.
23. B. LI, T.N. SAINATH, A. NARAYANAN, J. CAROSELLI, M. BACCHIANI, A. MISRA, I. SHAFRAN, H. SAK, G. PUNDAK, K.K. CHIN, K.C. SIM, *Acoustic modeling for Google Home*, Interspeech, Stockholm, Sweden, August 20-24, 2017, pp. 399–403.
24. B. LOGAN, *Mel frequency cepstral coefficients for music modelling*, Ismir, pp. 1–11, 2000.
25. I. LOPATOVSKA, K. RINK, I. KNIGHT, K. RAINES, K. COSENZA, H. WILLIAMS, P. SORSCHE, D. HIRSCH, Q. LI, A. MARTINEZ, *Talk to me: Exploring user interactions with the Amazon Alexa*, Journal of Librarianship and Information Science, United Kingdom, 2019, pp. 984–997.
26. V.B. MITITELU, D. TUFIŞ, E. IRIMIA, *The reference corpus of the contemporary Romanian language (CoRoLa)*, Proceedings of the Eleventh International Conference on Language Resources and Evaluation, Miyazaki, Japan, 2018.
27. K. NODA, Y. YAMAGUCHI, K. NAKADAI, H.G. OKUNO, T. OGATA, *Audio-visual speech recognition using deep learning*, Applied Intelligence, **42**, *4*, pp. 722–737, 2015.
28. D.S. PARK, W. CHAN, Y. ZHANG, C.C. CHIU, B. ZOPH, E.D. CUBUK, Q.V. LE, *Specaugment: A simple data augmentation method for automatic speech recognition*, arXiv preprint, arXiv:1904.08779, 2019.
29. R. PASCANU, T. MIKOLOV, Y. BENGIO, *On the difficulty of training recurrent neural networks*, International Conference on Machine Learning, Atlanta, United States, 2013, pp. 1310–1318.
30. V. PĂIŞ, D. TUFIŞ, R. ION, *Integration of Romanian NLP tools into the RELATE platform*, Proceedings of the International Conference on Linguistic Resources and Tools for Processing Romanian Language (CONSILR), 2019, pp. 181–192.
31. O. RUSSAKOVSKY, J. DENG, H. SU, J. KRAUSE, S. SATHEESH, S. MA, Z. HUANG, A. KARPATHY, A. KHOSLA, M. BERNSTEIN, A.C. BERG, L. FEI-FEI, *ImageNet large scale visual recognition challenge*, International Journal of Computer Vision (IJCV), **115**, *3*, pp. 211–252, 2015.
32. A. STAN, F. DINESCU, C. ȚIPLE, Ş. MEZA, B. ORZA, M. CHIRILĂ, M. GIURGIU, *The SWARA speech corpus: A large parallel Romanian read speech dataset*, International Conference on Speech Technology and Human-Computer Dialogue (SpeD), Bucharest, Romania, 2017, pp. 1–6.
33. A. STAN, J. YAMAGISHI, S. KING, M. AYLETT, *The Romanian Speech Synthesis (RSS) corpus: Building a high quality HMM-based speech synthesis system using a high sampling rate*, Speech Communication, **53**, *3*, pp. 442–450, 2011.
34. P.J.O. SUÁREZ, B. SAGOT, L. ROMARY, *Asynchronous pipeline for processing huge corpora on medium to low resource infrastructures*, 7th Workshop on the Challenges in the Management of Large Corpora, Cardiff, United Kingdom, 2019.
35. D. TUFIS, V.B. MITITELU, E. IRIMIA, M. MITROFAN, R. ION, G. CIOROIU, *Making Pepper understand and respond in Romanian*, 22nd International Conference on Control Systems and Computer Science (CSCS), DOI: 10.1109/CSCS.2019.00122, Bucharest, Romania, 2019.
36. D. TUFIŞ, V.B. MITITELU, E. IRIMIA, V. PĂIŞ, R. ION, N. DIEWALD, M. MITROFAN, M. ONOFREI, *Little strokes fell great oaks. Creating CoRoLa, the reference corpus of contemporary Romanian*, Revue Roumaine de Linguistique, **LXIV**, *3*, pp. 227–240, 2019.
37. K. YOU, M. LONG, J. WANG, M.I. JORDAN, *How does learning rate decay help modern neural networks?*, arXiv preprint, arXiv:1908.01878, 2019.

Received October 18, 2020