Maria NUȚU

Machine Learning based Solutions for Text Processing and Speech Synthesis

Editura Universității Transilvania din Brașov

2024

EDITURA UNIVERSITĂȚII TRANSILVANIA DIN BRAȘOV

Adresa:	Str. Iuliu Maniu nr. 41A
	500091 Brașov
Tel.:	0268 476 050
Fax:	0268 476 051
E-mail:	editura@unitbv.ro

Editură recunoscută CNCSIS, cod 81

ISBN 978-606-19-1769-3 (ebook)

Referenți științifici:

Prof. univ. dr. Călin ENĂCHESCU, Universitatea de Medicină, Farmacie, Științe și Tehnologie „George Emil Palade” din Tîrgu Mureș

Prof. dr. Dan CRISTEA, m. c. A. R., Universitatea Tehnică „Gheorghe Asachi” din Iași

Contents

Acknowledgements

This volume is the result of seven years of research at the Babeș Bolyai University of Cluj-Napoca. First of all, I would like to express my gratitude towards my scientific supervisor, professor Horia F. Pop, who wisely guided me throughout my entire research process.

I would like to thank to associate professor Marius Păun from Transilvania University of Brașov who opened me the perspective of the doctoral studies. Wihtout him and Cristi Irimia, PhD, General Manager at Siemens Industry Software SRL (former LMS Romania SRL), I would have never started this research journey.

A substantial part of the research was developed within a research project (supported by a grant[1] of the Romanian Ministry of Research and Innovation) within the Technical University of Cluj-Napoca. Here I would like to express my gratitude to professor Mircea Giurgiu and to associate professor Adriana Stan for their cooperation and for their willingness to accept me in this project as a research assistant.

I would also want to thank to lecturer Beata Lazar-Lorincz and lecturer Adriana Coroiu for their support and collaboration, but above all, for their friendship over all these years.

I want to mention professor Claudia Marțiș, from the Department of Electrical Machines and Drives, Technical University of Cluj-Napoca, who guided me during an interdisciplinary research at the beginning of my doctoral studies.

At last, but not least, I am grateful to my family for their understanding, encouragements and cheering ups during all this process, especially in the cold moments.

[1]PCCDI – UEFISCDI, project number PN-III-P1-1.2-PCCDI-2017-0818/73, within PNCDI III

List of Figures

List of Tables

List of Abbreviations

ADR	Automatic Diacritics Restoration
AI	Artificial Intelligence
ANN	Artificial Neural Network
BiLSTM	Bidirectional Long Short Term Memory cells
CNN	Convolutional Neural Network
CWE	Character-level Word Embeddings
DNN	Deep Neural Networks
FFNN	Feed Forward Neural Network
GRU	Gated Recurrent Units
HMM	Hidden Markov Models
k-NN	K Nearest Neighbor
IDF	Inverse Document Frequency
LDA	Latent Dirichlet Allocation
LE	Letter Encodings
LSTM	Long Short Term Memory cell
MCD	Mel Cepstral Distortion
ML	Machine Learning
MLP	MultiLayer Perceptron
MOS	Mean Opinion Score
MSD	Mel Spectrogram Distortion
MSD	Morpho-Syntactic Descriptions
MuSHRA	MUltiple Stimuli with Hidden Reference and Anchor
NLP	Natural Language Processing
NLTK	Natural Language Toolkit
OHE	One-Hot Encoding
POS	Part-Of-Speech
ReLU	Rectified Linear Unit
RNN	Recurrent Neural Network
RPOS	Root of Part-Of-Speech
seq2seq	Sequence-to-sequence
SPSS	Statistical Parametric Speech Synthesis
SVM	Suport Vector Machine
TF	Term Frequency
TF-IDF	Term Frequency-Inverse Document Frequency
TTS	Text-to-Speech
WE	Word Embeddings

Introduction

This chapter presents the theme of this book as well as the motivation of addressing the selected aspects.

Motivation

The main objective of this work is focused on processing the text and on synthesising the speech. Going deeper, we enriched the text processing tools which automatically solves tasks for texts written in the Romanian language, such as diacritics restoration, lemmatization and part-of-speech tagging. To gain more experience we went beyond the linguistic field in an attempt to cover certain gaps within the medical field that could be automated. For the speech synthesis part we worked on methods to enrich the expressivity of the artificially created voice together with ways of improving its quality.

These two direction (text processing and speech synthesis) would eventually sum up in the near future, in order to obtain a solid tool that is able to produce a high quality expressive synthetic voice from the Romanian texts.

Medical text data processing

In the era of Big Data, more and more information is available almost everywhere in any form (written, drawn, audio, video, etc.) and in a variety of communication styles: from formal (technical online courses, job descriptions, invitation to business or professional events, etc.) to informal (social networks, written blogs, etc.). Processing such large amounts of data can slow the daily activities, leading to fatigue or to exceed the deadline of daily tasks.

One of the domains which operates with the above mentioned large datasets is the medical domain. Medical physicians must not only make the right decisions based on the patient's history, but also fit in the time allocated to a person's consultation. Beyond analysing and correlating different aspects from the patient's life, the medical doctor should think on the spot a treatment scheme compatible with all the pre-existing ailments or diseases of the consulted patient. Having all these aspects in mind, the researchers investigated the impact of machine learning algorithms on developing better tools to analyze medical data. For example, machine learning algorithms can be used in medical imaging (namely X-rays or Magnetic resonance imaging -MRI- scans) using pattern recognition to search the patterns that indicate a particular disease [1]. Another application of machine learning in the medical domain is to gain insight in the written information of every patient. More precisely, using the topic modelling techniques, we can automatically find one person's diagnostic by analysing the personal medical records. Thus, we not only ease the medical doctor's routine work, but we also avoid the effects of fatigue in making erroneous decisions. These automations do not suppress the human role, in the sense that there

will always be the need of specialised human intervention, in order to offer a customised and contextualised interpretation, but at the same time, avoiding doing the repetitive and monotone work is priceless.

Starting from all the above mentioned aspects, we analysed medical records from a medical physician in order to find a topic modelling machine learning tool to automatically find one person's diagnostic. We processed an English written dataset, containing notes on patients' health conditions, manually gathered by a medical family doctor. Based on the state-of-the-art analysed in Section 3.1.2 from Chapter 3, we applied topic modelling techniques, namely the Latent Dirichlet Allocation (LDA) and the Latent Semantic Indexing, to cluster the medical documents based on the diagnostics described through similar symptoms. Our original results are described and discussed in Section 3.1.3 from the same Chapter and published in the research paper [2].

When it comes to medical data, an important role is played by the written content obtained from questionnaires' responses (to determine different traits, communication styles, psychological personality, future trends in shopping or marketing, etc.). Therefore, it becomes imperative necessarily to discover a methodology of automatically gaining insight from the collected data. Recent studies widely addressed this aspect obtaining worth mentioning results, which we have described in Section 3.2.1.

Enhancing the Romanian Text-to-Speech systems

Starting with the first years of life, the human specie learns how to communicate using words. Through speech, we express our needs, ideas, emotions or feelings. Thus, the speech synthesis, or the process of generating spoken language from a written given text, earned its place on the top of artificial intelligence's researchers' interest. Nowadays, with the help of modern technologies and deep learning [3], [4], [5], we can obtain high quality artificial speech, close to the natural human speech. However, in most of the cases, the text-to-speech systems manage to transmit only the information comprised by the text, with no content about the speaker's emotions, characteristics or tones (sarcasm, irony, etc). This lead to a linear message, sometimes different in meaning from the original intended idea.

Maybe one of the most useful applications of speech synthesis is helping people diagnosed with severe illnesses that lead to voice loss (among which throat cancer and motor neuron disease), either by recreating their original voice using their older audio recordings, whenever is possible, or by using an artificial voice output by a Text-to-Speech (TTS) system. A current and mundane example is the case of the American movie actor Val Kilmer who lost his voice after being diagnosed with throat cancer. When it comes to movies, the verbal communication is crucial, as acting involves transmitting a message both verbally and especially artistically, with different tones, intonations and emotions, leading to hidden meanings. Today Val Kilmer continues to play in movies by using an artificially produced voice[2].

Another famous example is that of the scientist Stephen Hawking[3] who lost his voice after falling ill with an early-onset slow-progressing form of motor neuron disease, which slowly paralysed him, leading to the incapacity to speak. In this

[2]Videos and samples of his reconstructed voice are available online: https://www.youtube.com/watch?v=OSMue60Gg6s

[3]More information is available here: https://www.hawking.org.uk

case too, the original scientist's voice could be recreated, as there are available many audio recordings with his voice, describing his scientific findings and research.

However, despite these two examples presented above, for the majority of patients, audio samples or recordings with their original voices are not always available. This implies creating an artificial voice only with the existing datasets tailored especially for this purposes [6], [7]. From this point, two questions arise:

1. How can we create voices that convey the speaker's emotions?

2. How can we create quality voices based on small data sets (for low-resourced languages), being known that current deep learning technologies require large input datasets for training?

Having those ideas in mind, many researchers focused their work on overcoming these aspects. We address these issues in the second part of this book. Chapter 4, in Sections 4.3 and 4.4, presents the main ideas as well as a brief state of the art for both emotional TTS (Figure 4.4) and speech synthesis for low-resourced languages (Figure 4.3). Our original contributions are described in Chapter 5.

As a first step, we focused on ways to improve the quality of the obtained synthesized voice, since there are few large datasets available for the Romanian language [8], so necessary for synthesis processes. Therefore, we investigated different techniques of post-filtering the obtained synthesized voice in order to correct the artifacts that can appear after training the text-to-speech system with a limited set of input data. The results are presented in our original research paper [9].

Another step was to create MARA[4] [10], a dataset with expressive data to be used in future research. Based on the newly created dataset, we then analyzed different ways of artificially increasing the volume of expressive data, as well as the impact of this new data on subsequent syntheses. The results are presented in our original research [10].

Research work as a whole

In order to obtain a more expressive voice within the text-to-speech synthesis process, we should model and control the prosody (intonation in speech) in a way close to natural speech. Prosody can be shaped both by the characteristics of the voice (intonation, stress, tonality, etc.) and by various annotations of the written text (accent, parts of speech, etc.). Therefore, as future work, we intend to create a software product which will integrate both parts of the current book: Romanian Natural Language Processing (NLP) and Expressive TTS. More precisely, the input text, processed and annotated using the systems developed in [11], [12], [13] will be passed through a deep TTS system leading to a more expressive synthesised output.

On the other hand, when we applied the NLP mechanisms for data from the medical domain, we took into consideration only the texts written in English. When it comes to the Romanian language, written text should obey certain rules. Diacritics play an important role in understanding the meaning of a given text. For instance, the written form *„peste"* without diacritics and no other contextual information, can mean both *pește (En. fish)* or *peste (En. over)*.The systems developed within our research [11], [12], [13] offer us the possibility to preprocess text written in Romanian,

[4]The dataset is available online: `https://speech.utcluj.ro/sped2021_mara/`

making it appropriate to be given as input to different machine learning classification learners. As future work, we intend to use the systems [11], [12], [13] to gain more insight from these medical Romanian texts.

Book structure

The present book is structured in two parts, as we addressed two correlated domains, namely Text Processing and Speech Synthesis.

I The first part of the volume comes to offer a solution to automatize the text processing tasks, as follows:

- **Chapter 1** describes the theoretical background for the Natural Language Processing field. We presented the core areas with their main applications, together with correlated research papers, as synthesised in Figure 1.3.
- **Chapter 2** introduces our original contributions in solving linguistic problems using deep learning algorithms, such as restoring the diacritics for a written text [12] and finding the lemma [11] or the part of speech of certain given words [13]. All the experiments were conducted on texts written in the Romanian Language. The results are intended to be used in correlation with the findings from Chapter 5.
- **Chapter 3** presents our personal contributions in processing the written text from the medical domain by applying the machine learning (ML) algorithms for two main tasks: identifying the medical diagnostic through the topic modelling techniques [2] and interpreting the psychological questionnaires results with the aid of the classification learners [14]. All the experiments are based on the texts written in English Language.

II The second part of the volume focuses on improving the Romanian Text-to-Speech systems in terms of expressivity and speech quality.

- **Chapter 4** offers a brief theoretical background of the main speech processing aspects addressed within the current research and the state-of-the-art in the field of Text-to-Speech systems, using the Machine Learning methods. We focus on Expressive TTS and on Speech Synthesis for low resourced languages. The information collected in this chapter facilitates understanding the research published in [9] and [10].
- **Chapter 5** presents our personal contributions in improving the Romanian TTS systems by addressing two main aspects: improving the quality of the synthesised voice [9] and enhancing the expressivity of the TTS system's results [10]. The experiments were developed within a research project, 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. Our project is described in detail online at Sintero Project.

Original Contributions

The current book derives from the theoretical and experimental research done in two main domains: Text Processing and Speech Synthesis.

I For the Natural Language Processing field, we have offered solutions to:

- automatically restore the diacritics for a text written in Romanian. We have compared 6 deep learning architectures trained using only parallel input-output pairs of texts, with and without diacritics. [12]
- automatically determine the lemma for the Romanian words. We have analysed 24 systems based on Deep Neural Networks, trained on labelled pairs of words and the corresponding lemmas, using at most the part-of-speech tag as morphological information.[11]
- automatic Romanian Part of Speech tagging. We have analysed two types of architectures:
  (a) simple long short-term memory networks (LSTM) - based networks
  (b) sequence-to-sequence architecture (seq2seq) based on LSTM layers - with different types of encodings for the input data (one hot encoding or letter encoding)

  resulting in 10 systems to be compared.[13]

II From the perspective of Speech Synthesis, myself along with the Sintero[5] colleagues have:

- created a large speech dataset containing more dynamic intonation patterns, the MARA Dataset[6] [10]
- trained and tested 6 deep learning TTS systems to improve the expressivity of the synthesised voice, in the context of lacking expressive datasets. The results are discussed in the original research paper [10].
- trained and tested 20 deep learning TTS systems in 3 postfiltering scenarios in order to evaluate the impact of each approach on the quality of the synthesised voice [9].

[5]https://speech.utcluj.ro/sintero/
[6]https://speech.utcluj.ro/corpora/mara.html

Part I

Solutions for Natural Language Processing (NLP) problems

Chapter 1

Theoretical insights into NLP

In this chapter we present the background knowledge and the state-of-the-art in the field of Natural Language Processing problems, using Machine Learning methods. The information collected in this chapter facilitates the research published in [2], [11], [12], [13], [14].

1.1 The beginnings of NLP

Natural Language Processing (NLP) means empower the computers to interpret and/or to understand a message expressed in a natural language (either written or spoken) in a similar manner the human brain does. The first notable attempts in NLP research were done during the World War II when people become aware of the importance of translating and transmitting the messages across the battlefield in an encrypted and, ideally, an automated manner. We mention here the Enigma machine [15], the *Colossus Computer* [16] and the electromechanical *Bombe*[17].

In [18], Alan Turing proposed the *Imitation game* (known nowadays as the *Turing test*) with the purpose of determining if the computers can think. The setup is the following: a human interrogator should discriminate between a human and a machine/computer based on the responses given to a set of questions, posed in a written form. If the evaluator cannot correctly distinguish between human and machine, the computer has passed the test. The answers are evaluated in terms of human-predictability rather than the content's correctness. As the communication is restricted to written channels, the machine speech ability is not interfering.

Years before Turing, René Descartes prefigures aspects of the Turing test in his 1637 "Discourse on the Method"[19] when he writes:

" [H]ow many different automata or moving machines can be made by the industry of man . . . For we can easily understand a machine's being constituted so that it can utter words, and even emit some responses to action on it of a corporeal kind, which brings about a change in its organs; for instance, if touched in a particular part it may ask what we wish to say to it; if in another part it may exclaim that it is being hurt, and so on. But it never happens that it arranges its speech in various ways, in order to reply appropriately to everything that may be said in its presence, as even the lowest type of man can do. [19] "

Although Descartes predicts to some extend the Turing test's concept being aware of the automata linguistic limits within a conversation, he fails to consider that these limits might be overcome over time due to the scientific research and discoveries.

Denis Diderot formulates in his 1746 book "Pensées philosophiques" [20] a Turing-test criterion, but he restricts the participants to natural living beings, rather than considering the created machines:

"If they find a parrot who could answer to everything, I would claim it to be an intelligent being without hesitation." [20]

This statement expresses the general way of thinking during that period, known as *materialism*.

Since Turing introduced his test, it has been both highly influential (with applications from medicine [21], [22], text processing [23], image processing - face recognition [24] to speech synthesis [25] and industry [26]) and widely criticised [27], and has become an important concept in the philosophy of Artificial Intelligence.

Around 1958, one researcher with important contribution in NLP development was Noam Chomsky [28]. He started from the idea that models of language recognized sentences that were nonsense but grammatically correct as equally irrelevant as sentences that were nonsense and not grammatically correct. For instance, the sentence "Colorless green ideas sleep furiously" was classified as improbable to the same extent that "Furiously sleep ideas green colorless". A native English speaker can discriminate the former as grammatically correct and the latter as incorrect. Chomsky opinionated this should be expected of machine models too [28].

Before 1970s researchers were split into two groups. The firsts developed symbolic NLP [29], based on formal languages and syntax generation, while the others developed stochastic NLP [30], based on statistics and probabilistic models, with applications in pattern matching between texts or optical character recognition (OCR).

After 1970's, with the development of the NLP techniques, the researchers split more specifically. One group focused on the logic-based paradigms, interested in the applications of encoding rules into mathematical logic, later creating the Prolog programming language [31]. Another group remained focus on the natural language understanding tasks, starting form Terry Winograd's SHRDLU program [32].

In the following section we will focus on more recent approaches for the NLP tasks, exclusively based on machine learning in general and deep learning in particular.

1.2 Processing the NLP tasks before Deep Learning (DL)

With a large amount of unlabelled data, one of the main challenges in solving NLP tasks is to learn a data representation from the inner data structure itself. This leads to Unsupervised Feature Learning, an approach to obtain a lower dimensional representation of the data from the higher-dimensional initial space. Techniques as Decision Tree Based Model, Support Vector Machine, Random Forest, Classification based on instances (k-NN), Logistic Regression or Principal Components Analysis have been successfully applied to solve NLP tasks as Topic Modelling [33], [34], Sentiment Analysis [35], [36], Text Classification [37], [38], [39].

In our research, we also evaluated the above mentioned algorithms. The experiments were introduced in our research papers [14] and [2], described in Chapter 3.

However, during the last years, once with the revival of the neural networks, the traditional approaches have been almost totally replaced.

1.3 DL for NLP

An artificial intelligence (AI) goal might be to upgrade from generating, communicating and storing the data to processing the available data. With a daily increasing of the data volume, deep learning seems to be the solution of AI for analysing these large amounts of data. Deep learning (DL) consists in a set of mechanism -

which we will briefly describe below - which can generate optimal solutions given an appropriate input dataset. In most of the cases, these algorithms equal or even outperform the human capabilities. Although there is not a standardized definition accepted by all the researchers, a neural network with two or more hidden layers is called deep neural network. The main differences between the different types of neural networks consists in:

- the number of layers
  - sequential neural networks - each node links the earlier layer with the next one (Feed Forward Neural Network - FFNN)
  - deep neural networks
- the way the nodes communicate between layers:
  - horizontally sharing the weights - Convolutional Neural Networks [40]
  - vertically sharing weights - Recurrent Neural Networks [41]
  - skipping layers - Residuals Neural Networks [42]
  - simply deactivating certain nodes - Dropout [43]
  - forcing neurons to focus on certain pieces of input information - Attention Mechanism and Transformers [44]
  - adversarial learning - Generative Adversarial Networks [45]

Among the diversity of the neural networks, we will focus our brief theoretical background only on those we analysed in our research studies [11], [12], [13].

Multilayer perceptron (MLP) [46] is a basic machine learning model, consisting in at least three layers: *input, hidden and output*. Each neuron within wihtin a layer is connected with all the neurons form both the previous and the following layers. However, the neurons do not communicate within th esame layer. The inputs are combined with certain initial weights in a weighted sum and the result is passed to an activation function. This function's output is fed into the next layer in a similar weighted manner. The weights adjust during each training epoch as the network's target is to minimize a loss/cost function value, computed between the network's predicted output and the desired output. Thus, the MLP is known as the simplest Feed Forward Neural Network (FFNN). Figure 1.1 illustrates a MLP arhitecture. Study [46] theoretically describes the MLP together with some applications.

Our experiments described in [9] are obtained using a feedforward neural network based the Text-to-Speech system [47]. Thus, the FFNN are suitable not only for NLP tasks, but also in other various domains.

Recurrent Neural Networks (RNN) Applied in our experiments from [11], [12], [13], Recurrent Neural Networks (RNNs) are a MLP-based neural network in which the output of the current time step is conditioned on the output of the previous time step. As a result, the RNNs are commonly used to model temporal sequences. However, a major problem with vanilla RNNs is that they cannot model sequences in which the temporal dependencies are stretched across multiple time steps.

The solution for this problem is to use more advanced network nodes, in which an internal state of the node can memorize the data snippets which are of interest to

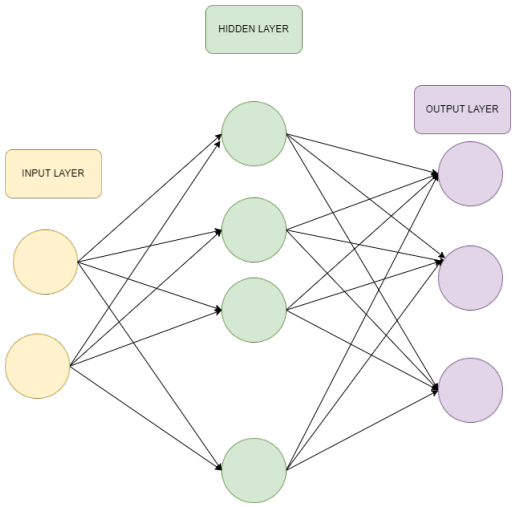


FIGURE 1.1: A Multilayer perceptron architecture

the current prediction, or forget the irrelevant data parts. One such specialized node is the Long Short Term Memory (**LSTM**) cell [41].

A LSTM cell (graphically depicted in Figure 1.2) contains the following elements:

- forget gate f_t - a neural network (NN) with sigmoid activation
- input gate i_t - a NN with sigmoid activation
- output gate o_t - a NN with sigmoid activation
- hidden state h_t - a vector
- memory state c_t - a vector

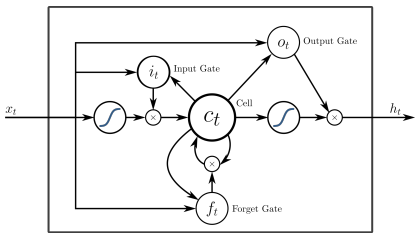


FIGURE 1.2: LSTM memory cell [48]

The input gate selects what new information should be stored in the current cell at a time step t. The forget-gate expresses the amount of information which will be discarded, while the output-gate will provide the activation to the final output of the LSTM block. The hidden state is calculated from the cell state passed through an activation function and element-wise multiplied with the output vector at the time step t.

We applied the RNN architectures for our NLP experiments from [11], [12], [13] described in Chapter 2.

Convolutional Neural Networks (CNN) As described in [12], Convolutional Neural Networks (CNN), originally used in image processing, are another type of deep networks largely used for the pattern recognition tasks.

A simple CNN architecture contains the following elements:

- a convolutional layer
- a non-linear activation layer
- a pooling (or sub sampling) layer
- a fully connected (softmax) output layer.

The convolutional layer defines a non-linear filter bank (or kernel), which is shifted over the input features using a fixed stride and generates a multi-dimensional feature map, which is processed by a non-linear activation function. The pooling layer reduces the representation of the convolutional layer's output, as well as decreases the memory requirements. In general, the polling layer is placed between the convolutional layers. The features with the highest values (maxpool) are fed into a fully connected layer, whose activations are finally passed into a softmax layer. The output of the softmax function represents the estimated probability distribution over the output labels. In some cases, a normalization layer is stacked on the pooling layer to normalize the data, with mean 0 and variance 1. The normalization step ensures the network's stability.

For the NLP field, the input of the CNN architecture consists in sentences, paragraphs or documents encoded as multidimensional matrices. Each smaller phrase segmentation (word or character) represents a row within the input matrices. During training, the CNN learns the text representation within the input language. Among the various NLP domains were the CNN outperforms we mention Sentiment Analysis [49], [50], [51], Topic Modelling [52], [53], [54], Relation Extraction [55], Relation Classification[56].

We applied the CNN architectures for our NLP experiments from [11], [12], [13] but also for the Speech Synthesis tasks from [10]. The results are described in Chapter 2 and Chapter 5 respectively.

1.4 NLP - core areas and applications

The research of the Natural Language Processing field is usually split into two main categories (most often with a slight or uncertain border between them two):

1. Core Areas - the study of fundamental problems (Figure 1.3)
2. Applications - combine two or more core areas in order to solve more specific practical problems (Figure 1.4)

Recent research studies [104], [105], [106] survey the state-of-the-art research works. We summarize main works in Figure 1.3 and Figure 1.4, based on the two categories above mentioned.

1.5 Sequence-to-sequence approach in the field of NLP

The sequence-to-sequence (seq2seq) architecture translates one sequence into another. It is formed of two parts: an *encoder* and a *decoder*, each of them being a

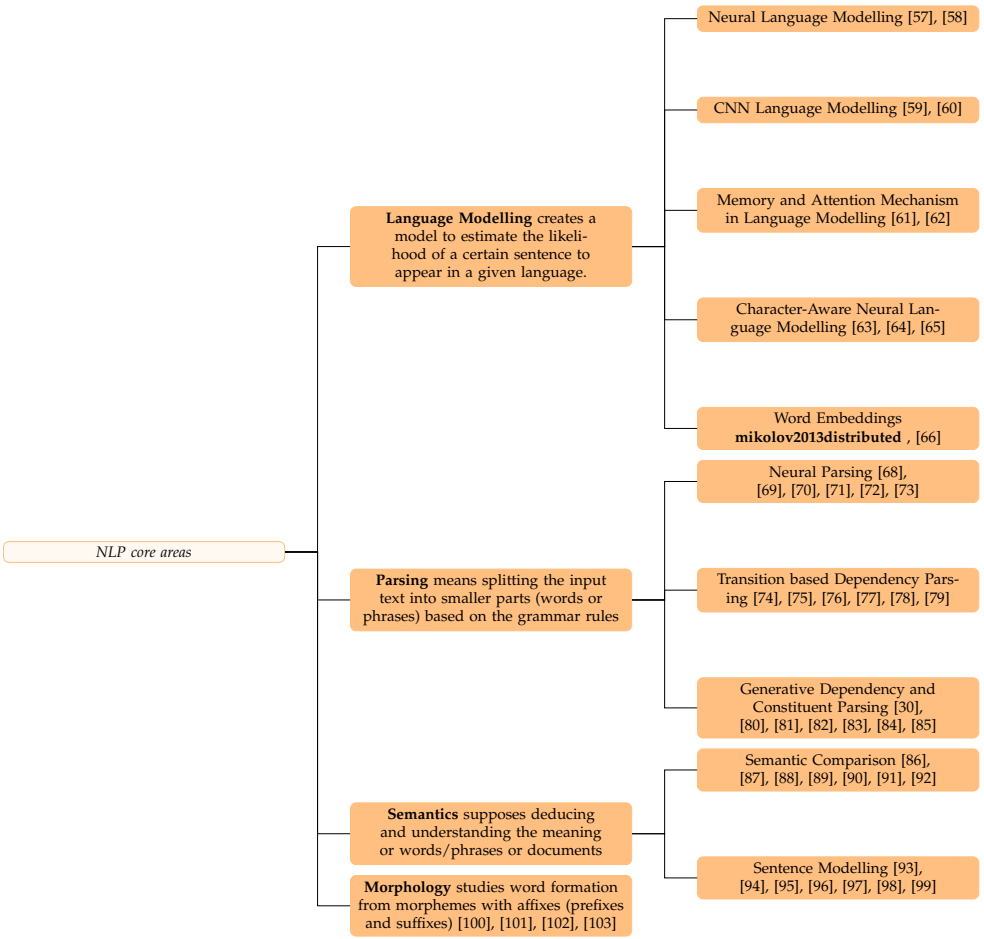


FIGURE 1.3: NLP core areas with studies

separate neural network. The encoder is responsible for understanding the input and representing it in a lower dimensional space. The output of the encoder will then be used to condition the decoding network's prediction. Figure 1.5 presents a seq2seq model for the word "masa" as input and "masă" as output. The tags <SS> and <SE> mark the start and the end of the sequence. The most prevalent architectures behind the encoders/decoders are the recurrent and convolutional neural networks.

The model illustrated in Figure 1.5 was analysed in our research paper [12] as we proposed to train an automatic diacritics restoration system for the Romanian language. We applied the seq2seq architecture in other NLP tasks, such as lemmatization [11] or POS tagging [13], all particularised for the Romanian language as well. The experimental setup and results are discussed in Chapter 2.

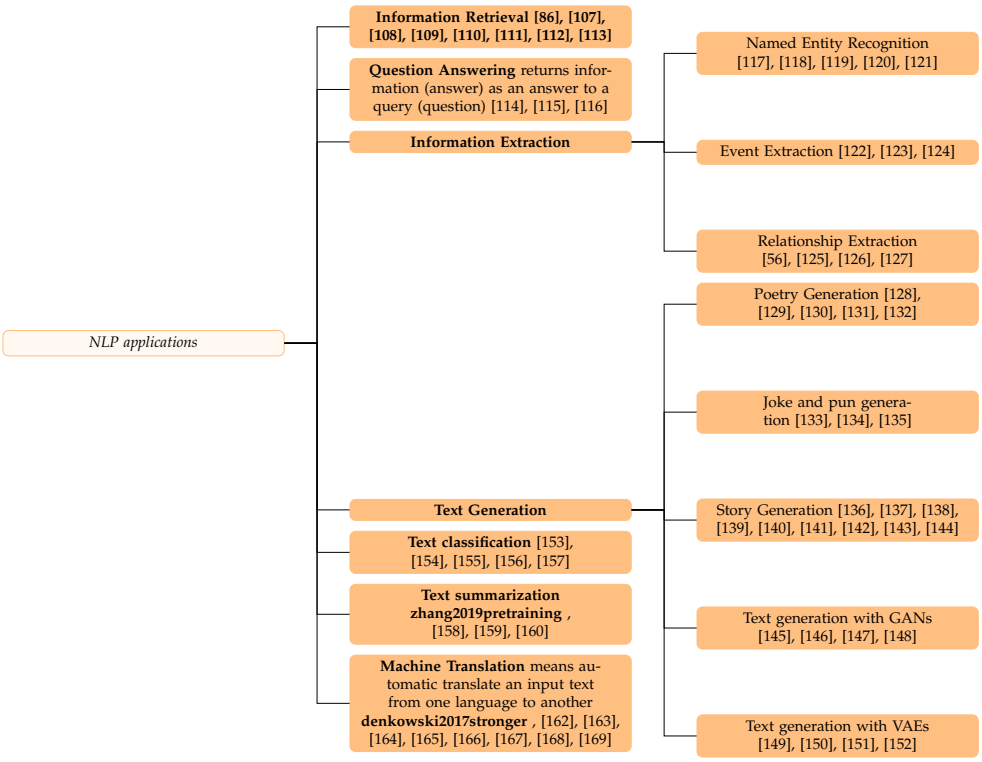


FIGURE 1.4: NLP applications with studies

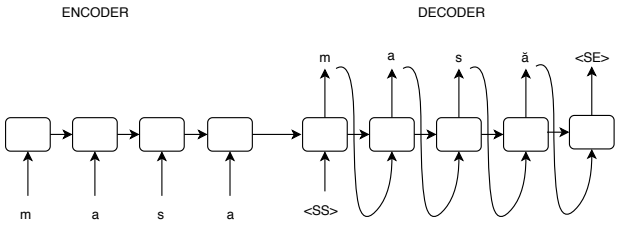


FIGURE 1.5: Sequence-to-sequence architecture in NLP [12]

1.6 Evaluation methods

The neural NLP systems output are predictions of the results desired in a certain task. The most common evaluation metrics which measures the NLP systems' quality are described below. First, we define the following terms:

- **TP** = **True Positive** - number of correctly classified instances as belonging to the class of interest
- **TF** = **True Negative** - number of correctly classified instances as not belonging to the class of interest
- **FP** = **False Positive** - number of incorrectly classified instances as belonging to the class
- **FN** = **False Negative** - number of incorrectly classified instances as not belonging to the class

1. **Classification accuracy** is determined by the ratio of the number of correctly classified instances to the total number of classified instances [14].

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1.1}$$

2. The **sensitivity** metric (as known in statistics domain) or **recall** metric (as known in Information Engineering domain) is given by the ratio between the number of correctly classified data as belonging to the class of interest and the sum of the number of data correctly classified as belonging to the interest class and the number of data incorrectly classified as not belonging to the class of interest.

$$Sensitivity = Recall = \frac{TP}{TP + FN} \tag{1.2}$$

3. The metric of **specificity** is given by the ratio of the number of data correctly classified as not belonging to the interest class and the sum of the number of correctly classified data as not belonging to the interest class and the number of data incorrectly classified as belonging to the class of interest.

$$Specificity = \frac{TN}{TN + FP} \tag{1.3}$$

4. The **precision** metric is given by the ratio between the number of data correctly classified as belonging to the class of interest and the sum of the number of data correctly classified as belonging to the interest class and the number of data incorrectly classified as belonging to the class of interest [171]:

$$Precision = \frac{TP}{TP + FP} \tag{1.4}$$

5. The F_1 **score** is the harmonic mean of the precision and recall.

$$F1 = \frac{2}{\frac{1}{Recall} + \frac{1}{Precision}} = \frac{TP}{TP + \frac{1}{2}(FP + FN)} \tag{1.5}$$

In our research papers [11], [12], [13], [14] we analysed the models' performance using the five evaluation metrics described above. However, as a system performance depends on the task to be analysed, different evaluation metrics have been proposed in **clarkson2001improved** , [172], [173], [174].

Chapter 2

Addressing linguistic problems using Machine Learning (ML) models

2.1 Automatic Romanian Diacritics Restoration

In this subchapter we address the issue of automatic diacritics restoration (ADR) for the Romanian language using the deep learning strategies. The method was introduced in our original research paper [12].

2.1.1 Motivation

Automatic Diacritics Restoration (ADR) is the process of restoring the diacritic symbols in the orthographic texts. The applications of this process are numerous and include: spelling checkers, lexical disambiguation, part-of-speech tagging, natural language understanding, etc. The lack of diacritics is predominant in electronic texts where the user does not use adequate text editing software, or is not technologically proficient so as to use the diacritic symbols specific to his or her native/acquired language.

Most European languages contain different sets of diacritic symbols in their alphabets, the most numerous in French and Slovak. The set of diacritics used in European languages based on the Latin alphabet are illustrated in Table 2.1.

Language	Diacritics	Language	Diacritics
Albanian	ç ë	Italian	á è é ì í ï ò ó ù ú
Basque	ñ ü	Lower Sorbian	ċ č ě ł ń ŕ ś š ź ž
Breton	â ê ñ ú ö	Maltese	ċ ġ ż
Catalan	à ç è é í ï ò ó ú ü	Norwegian	å æ ø
Czech	á č é í ñ ó ř š ý z	Polish	ą ę ć ł ń ó ’s ’z ż
Danish	å æ ø	Portuguese	â ã ç ê ó ô ũ ü
Dutch	ë	Romanian	ă â î ş ţ
English	none	Sami	á ï ĉ d- ń ņ š t- ž
Estonian	ä č õ ö ž	Serbo-Croatian	ć č d- š ž
Faroese	á æ d- ó øú ý	Slovak	á ä č d’ é Í ñ ó ô ŕ š ť ú ý ž
Finnish	ä å ö š ž	Slovene	č š ž
French	á â æ ç é è ë ê î œ ù û ÿ	Spanish	á é í ó ú ü ñ
Gaelic	á é í ó ú	Swedish	ä å ö
German	ä ö ü ß	Turkish	ç ğ ö ş ü
Hungarian	á é í ó ö ő ú ü ű	Upper Sorbian	ć č ě ł ń ó ř š ž
Icelandic	á æ ∂ é í ó ö ú ý	Welsh	ă ě ĭ ŏ ŭ ŵ ŷ

TABLE 2.1: Diacritics in European Languages with Latin based alphabets

The Romanian language uses 5 diacritic letters: *ă, â, î, ş* and *ţ*. Although not all the words have alternative spellings with and without diacritics, in some cases, a missing diacritic could completely change a word's meaning (e.g *peste* = over vs. *peşte* = fish), while in other cases, the absence of the appropriate diacritic in the word's ending letter makes it impossible to discern between the definite or indefinite form of a noun (*mamă* = a mother vs. *mama* = the mother).

Tufiş et al. [176] reports that between 25% and 45% of the Romanian words contain diacritics, while in a random French text, only 15% of the words contain diacritic symbols [177]. The diacritic percentage across the European languages is reported in [178].

Motivated by the relevance of the diacritic restoration across various text-based applications, we address the Romanian ADR problem using the sequence-to-sequence deep learning architectures based on convolutional and recurrent neural networks.

2.1.2 Related work

With the increasing use of the electronic devices across different social and cultural categories, the need for high-quality ADR applications is more prevalent, and so is the number of published scientific studies. Simard [177] employs Hidden Markov Models trained at word level on French texts. For the Vietnamese language, Nguyen et al. [179] combine Adaboost and C4.5 decision tree classifiers with a letter-based feature set in five different strategies: learning from letters, learning from semi-syllables, learning from syllables, learning from words, and learning from bi-grams.

A deep learning approach for diacritics restoration is proposed by Náplava et. al. in [180] and uses Bidirectional Neural Networks combined with a language model. The model was tested for 23 languages, including among others: Czech, Slovak and Romanian.

For the Romanian language, in particular, the works of Mihalcea et al. [178], [181] explore instance based learning at letter level, using the Tilburg memory and the C4.5 decision tree classifier, scoring an overall F-measure of 98.30 %.

Tufiş et al. [176] propose a Part-Of-Speech tagger and the use of two lexicons to solve the ambiguity problem in Romanian ADR. An overall accuracy of 97.4 % is achieved at word level.

Ungureanu et. al [182] propose a word classification schema, based on the occurrence of the diacritics in each word (words always written with diacritics, words with no diacritics at all and words with different diacritical written pattern - words which change their meaning as diacritics are missing, as shown in Section 2.1.1). Then these categories are distilled into two dictionaries. During training and testing, the two lexicons are used to improve the ADR results, obtaining an overall F-measure of 99.34%.

In [183] Petrică presents a diacritics restoration system trained on unreliable raw datasets. First, the correctly spelled sections are identified and used as training data for the ADR. Second, the trained ADR is applied to the remaining parts of the initial text.

The previously described approaches use language models and linguistic information extracted from the texts at different levels. In this work, we propose a deep learning approach to solve the ADR problem for Romanian using only grapheme sequences, without any expert linguistic knowledge.

Sequence-to-sequence learning

The sequence-to-sequence (seq2seq) [184] architecture is designed to handle input and output sequences with different lengths. The most common applications for this architecture include automatic machine translation, video captioning, speech recognition and speech synthesis.

Broadly speaking, the seq2seq architecture is formed of two parts: an *encoder* and a *decoder*, each of them being a separate neural network. The encoder is responsible for understanding the input and representing it in a lower dimensional space. The output of the encoder will then be used to condition the decoding network's prediction. Figure 2.1 presents a seq2seq model for the word "masa" as input and "masă" as output. The tags <SS> and <SE> mark the start and the end of the sequence. The most prevalent architectures behind the encoders/decoders are the recurrent and the convolutional neural networks.

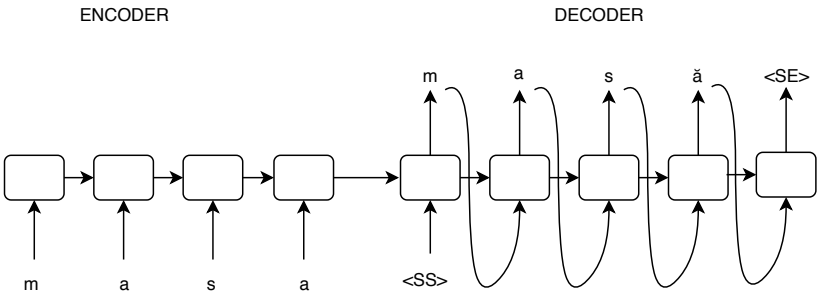


FIGURE 2.1: Sequence-to-sequence flow

The characteristics highlighted above make the seq2seq learning a good candidate for the Romanian ADR problem. Our systems [12] are based on convolutional neural networks and recurrent neural networks, theoretically introduced in Chapter 1 and technically described in the next subsection.

2.1.3 Experimental setup

Training Data

For training and testing our models [12], we selected a subset of the CoRoLa text corpus [185]. The subset contains 51.043 sentences with 1 million tokens and 63.194 unique words. The style of the text is belletristic. The corpus is not purposely build for ADR tasks, but can be considered as a reliable source of correctly typed text (i.e. containing the correct diacritics) in Romanian, as it was manually annotated at word-level with several linguistic information. We subsequently split the dataset into disjoint training (80%) and testing (20%) sets, each of them being individually shuffled.

A few pre-processing steps were performed,including the following operations:

- converting text to lowercase
- striping the digits and punctuation
- striping the diacritics
- segmenting the text in trigrams
- creating pairs of input-target sequences

- appending a start-character ("\t") and an end-character ("\n") to the target trigram

An example of a pre-processed sentence is shown in Table 2.2. The obtained input-output pairs for the chosen sentence are illustrated in Table 2.3.

TABLE 2.2: An example of a pre-processed sentence

Initial sentence	"Mă uitasem la ceas, era încă ora 22.00."
Pre-processed sentence	"ma uitasem la ceas era inca ora"

TABLE 2.3: Input-output trigrams for a chosen sentence

Input sequence	Target sequence
ma uitasem la	\t mă uitasem la \n
uitasem la ceas	\t uitasem la ceas \n
la ceas era	\t la ceas era \n
ceas era inca	\t ceas era încă \n
era inca ora	\t era încă ora \n

When an unknown input sequence is decoded, we begin with the starting character and use the decoder to predict the next character until the ending character is generated. The trigrams were chosen to represent the context of the current sequence.

After the pre-processing steps, the train set ended-up containing 616.691 tokens, while the test set contained 162.791 tokens.

System architectures

For our initial tests, we selected 2 ADR systems [180], [186] previously applied for Romanian. The systems were retrained using our dataset, but preserving the original parameter values.

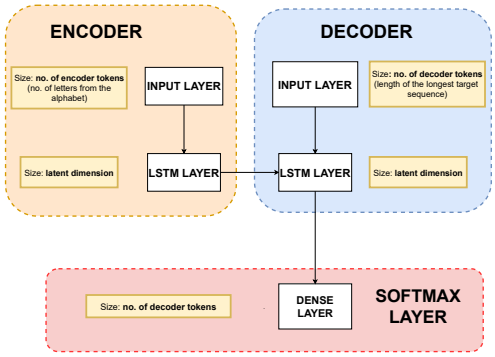
Inspired by the architectures described in these two systems, we analyzed four other architectures with various combinations of recurrent and convolutional layers. For the implementation, we relied on Keras[1] with the TensorFlow[2] as backend. The networks' hyperparameters were tuned using a small development set. The results were reported in our original research paper [12].

All the 6 architectures are described in the following subsections with the previously published works marked with an asterisk (*). All systems were trained over 50 epochs.

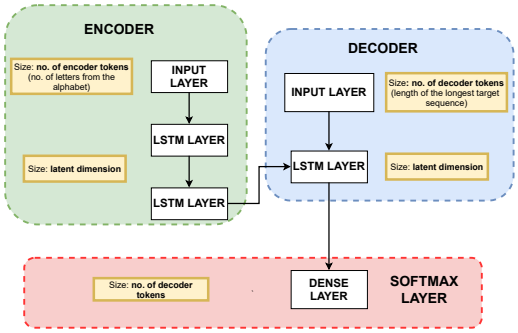
One layer LSTMs (ID: LSTM) In the RNN sequence-to-sequence architecture [12] the encoder and the decoder both included one LSTM layer. A latent dimension of 128 for both layers and a batch size of 512 were chosen. The input to the encoder and to the decoder was one-hot encoding at character level. The input of the decoder was also conditioned on the hidden state of the encoder. The output of the decoder LSTM layer is sent to a softmax dense layer with a dimension equal to the length of the one-hot encoded target character set. Figure 2.2a illustrates the design of the RNN architecture.

[1]https://keras.io/

[2]https://www.tensorflow.org/



(A) **The LSTM model architecture**. Both the encoder and the decoder are composed of one LSTM layer. The size of the encoder's input layer is given by the length of the dataset's alphabet. The size of the decoder's input layer is equal to the length of the longest target sequence. The output of the decoder is passed to a dense layer with a softmax activation.



(B) **The LSTM stacked model architecture**. The encoder is composed of two LSTM layers. The output of the second layer of the encoder is passed to the LSTM layer of the decoder. The result is fed to a dense layer with a softmax activation.

FIGURE 2.2: The LSTM model architectures for the task of automatic diacritics restoration

Stacked LSTMs (LSTM_stacked_*_) In order to improve the results, one additional LSTM layer was added to the encoder [12]. The newly obtained encoder was tested in two different contexts. First, we used one LSTM layer for the decoder (**ID: LSTM_stacked_1** - Figure 2.2b). Then, another LSTM layer was stacked in the decoder (**ID: LSTM_stacked_2**). The model **LSTM_stacked_1** was trained with a 256 latent dimension and a 128 batch size. For the model **LSTM_stacked_2** a batch size of 512 and a latent dimension of 128 were used.

Convolutional Sequence-to-Sequence (ID: CNN) In our experiments from [12], the CNN architecture contains 3 convolutional layers with 128 feature maps and a kernel of size 3, for both the encoder and the decoder networks. An attention architecture with a softmax activation follows the 3-layered convolutional decoder networks. The output is processed by another 2 convolutional layered architecture, with a softmax dense output. Figure 2.3 illustrates the model structure. The model is trained with a batch size of 1024 and a 128 latent dimension.

***RNN and CNN hybrid model (ID: hybrid_seq2seq)** The RNN and CNN hybrid model described in [186] uses two paths - at character level and at word level. For the character path, an embedding layer feeds the input to 3 stacked CNN layers. The word path goes through embedding and a bidirectional LSTM (biLSTM). The two paths are merged by projecting words to characters based on a projection matrix which is received as an additional input. Hence, the character and the word embeddings are jointly learned. These embeddings are fed to a stack of 3 convolutional layers. The output is predicted using a time distributed dense layer. For the experiments described in [12] we trained the network with a batch size of 32. The system's architecture is illustrated in Figure 2.4.

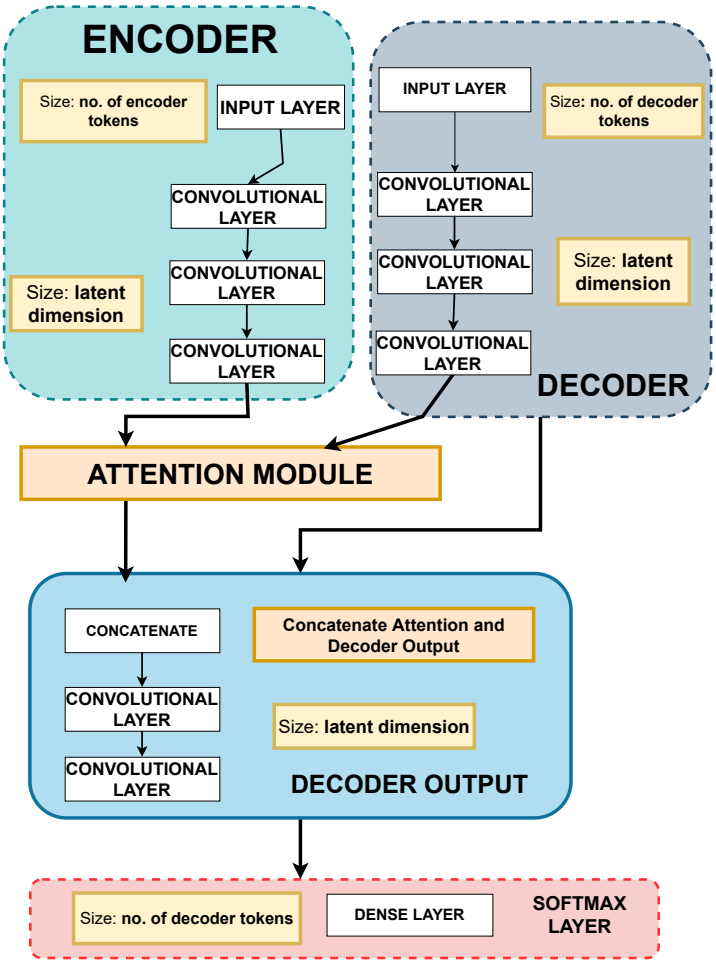


FIGURE 2.3: The CNN model architecture for the task of automatic diacritics restoration

***RNN with language model** In [180] a combination of character-level recurrent neural network based model and a language model are applied to automatic diacritics restoration.[3] The core model uses a bidirectional LSTM which deals with the previous and the snext letter contexts in the sequence.

The bidirectional RNN contains 2 stacked layers with residual connections, composed of 300 LSTM units. A batch size of 200 was chosen. The model language is based on the left-to-right beam search. At each time step, the output of the biLSTM layers is reduced by a fully connected layer to v-dimensional vectors, where v is the size of the output vocabulary. A non-linear rectified linear unit (ReLU) activation function is applied to the reduced vectors. The final output layer uses a softmax activation.

2.1.4 Evaluation and discussion

All the 6 system architectures introduced in [12] were evaluated using the *classification accuracy metric*, which is defined as the ratio between the correct predictions and the total number of samples. We computed the accuracy at three different levels:

[3]https://github.com/arahusky/diacritics_restoration

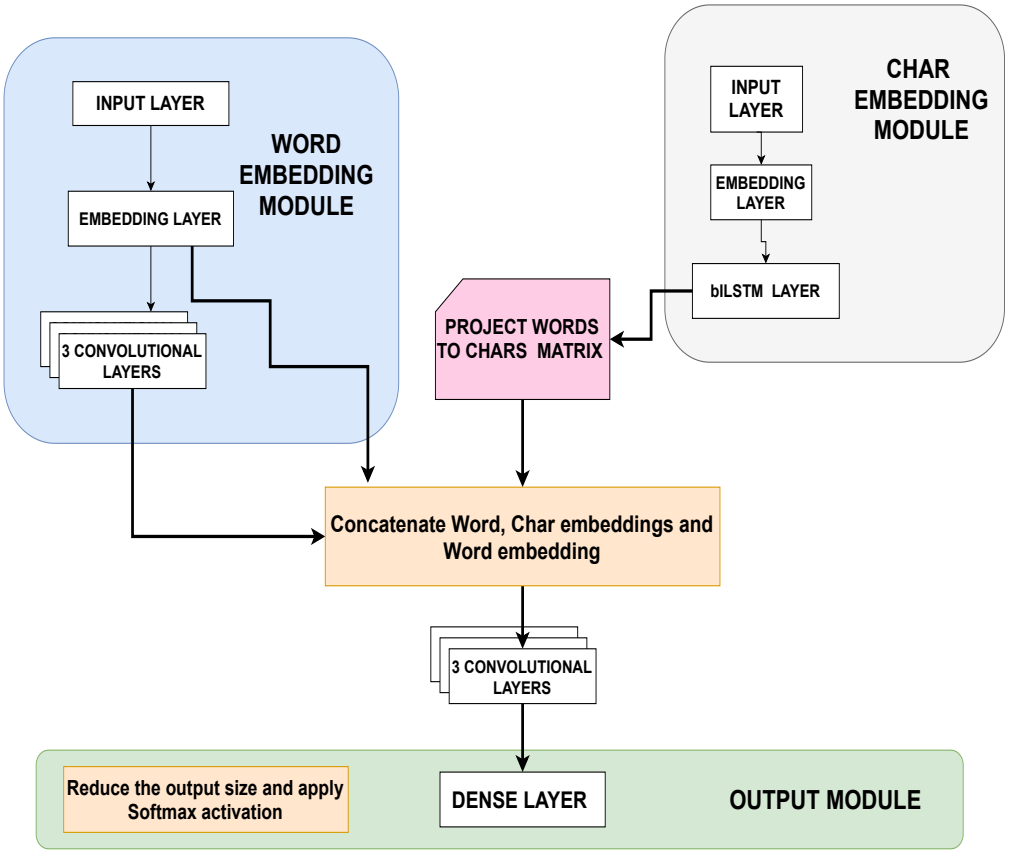


FIGURE 2.4: The **seq2seq-hybrid** model architecture for the task of automatic diacritics restoration

trigram, word and character level. At trigram and word level the accuracy reflects the number of correct predictions made by the system overall. At character level, we computed the accuracy only for the characters which may be written with diacritic symbols (*a, i, s, t)*. Accuracy results for all the systems are presented in Table 2.4.

A separate set of results is shown in Table 2.5, where the 4 ambiguous letter sets in Romanian (a-ă-â, i-î, s-ş, t-ţ) were analyzed individually.

TABLE 2.4: Network parameters and accuracy results

Architecture ID	Latent dimension	Batch size	Accuracy		
			3-gram level	Word level	Character level
LSTM	128	512	75.50%	89.98%	71.61%
LSTM_stacked_1	256	128	79%	93 %	78%
LSTM_stacked_2	128	512	84%	94 %	82%
CNN	128	1024	**91%**	**97 %**	89%
hybrid_seq2seq[186]	N/A	32	77%	92%	84%
LSTM_Language_model [180]	300	200	84 %	96 %	**90%**

The highest accuracy in terms of trigrams and words, was achieved for the convolutional network (ID **CNN**), while the single-layer LSTM system (ID **LSTM**) had the lowest accuracy. One explanation can be found in the recurrence of the LSTMs, which may require larger data context, as opposed to the CNN, which uses the attention layer and the sliding windows (kernels) to simulate the recurrence.

TABLE 2.5: Accuracy results for individual ambiguous pairs of the best performing system

Architecture ID	Accuracy			
	a-ă-â	i-î	s-ş	t-ţ
CNN	93.51 %	99.44 %	98.39 %	97.94 %

However, at character-level, the system described in [180] outperforms all the other systems. The justification for this result can be the use in [180] of a language model together with the RNN, while our systems restore the diacritics without any additional linguistic information.

2.1.5 Conclusions and future work

In [12] we have compared 6 neural networks architectures with the objective of obtaining automatic diacritics restoration systems, applied to Romanian language. All the models were trained using only parallel input-output pairs of texts, with and without diacritics. As input to the sequence-to-sequence architectures we used the character-level one-hot encoding. However, it is a common practice in NLP to encode the words or characters using multidimensional embeddings obtained from large amounts of text data. These embeddings would allow the network to have an initial estimate of the characters' function in a language. So as future work, we intend to substitute the one-hot encoding with letter or word embeddings, and also to include additional linguistic or semantic information.

In our experiments presented in [12] we split the data into trigrams, both for training and for testing. Each network received a diacritic-stripped trigram and predicted the entire corresponding sequence with diacritics. We intend to experiment with other N-gram, allowing the network to capture more context. One other mean of improving the results is to predict the diacritics only for the sequence-ending word, considering all previous words to be correctly typed.

In addition, we are planning to investigate other types of fully convolutional neural networks, based on the dilated convolutions combined with attention mechanisms, architectures largely used in the Machine Translation and the Speech Synthesis fields, but unexplored in the ADR domain.

2.2 Automatic Romanian lemmatization

This subchapter describes a deep learning sequence-to-sequence approach to improve the task of automatic Romanian lemmatization. We have introduced the methods in our original research paper [11].

2.2.1 Motivation

Recent works in the deep learning field focus on using as few input features as possible, offering an end-to-end strategy fed only with (labelled) data needed for the task. For instance, in the text-to-speech scenario [187], [188], to create synthetic voices, only pairs of written text and corresponding audio are necessary and it is the system's responsibility to learn a mapping between the lexical and the acoustic features.

TABLE 2.6: Literature results for Romanian Lemmatization.

Reference	Dataset	Context	Architecture	Accuracy
Boroş [200]	DEX	POS	Perceptron	94
Chakrabarty et al. [201]	Romanian-RRT	Semantic embedd.	BiLSTM	94.32
Boroş [202]	Romanian-RRT	POS+WE	BiLSTM + LSTM	94.79
Yildiz et al. [203]	ro-rrt	BiLSTM+ vector repres.	LSTM + BiLSTM	96.54
Chrupala et al. [204]	MULTEXT-EAST	POS +Lexical feats	Search alg. + Classifiers	97.78
Qi et al. [205]	ro-rrt	LSTM + dict.	LSTM + Attention	97.95
Kanerva et al.[206]	Romanian-RRT	POS taggs	BiLSTM + LSTM	98.25
Straka et al. [207]	Romanian-RRT	BERT+WE+CWE	LSTM	98.59 (F1-score)
Dumitrescu et al. [208]	SIGMORPHON 2018	N/A	BiLSTM + LSTM	88 (reinflection accuracy)

Although it seems that no additional textual or lexical data is needed, the text processing tasks, such as phonetic transcription [47], lemmatization [189], [190] or part of speech (POS) tagging [191] can improve the quality of the synthesised speech.

The study addresses the *lemmatization*, as this intends to be a preliminary text processing step in the task of creating synthetic voices with emotional expressivity. Romanian is a low resourced language considering the lack of emotional labelled corpora, thus it is a challenge to develop a method to automatically label the text based on the emotional content level. A range of studies addresses the connection between the word's lemma and emotions within a text. In [192] the lemma is used to measure the level of emotion in political discourse, but the method can be extended to other domains. The authors of [193] and [194] propose lemma-based approaches to analyze the texts for preventing a suicide behaviour. In [195] lemma is used to create an emotion lexicon. Apart from detecting emotions in texts, lemma is efficient to reduce the complexity of the vocabulary in the fields like topic identification [196], text summarisation [197], word search [198], machine translation [199] or keyword spotting [198].

In the experiments presented in [11] three types of Deep Neural Networks architectures have been compared. The Long Short-Term Memory (LSTM) cells and Convolutional Neural Networks (CNN) with Attention layer have been chosen to train 24 systems with three types of input-output data: simple pairs composed of word and corresponding lemma, tuples of words enriched with morphological information and the corresponding lemma and, finally, trigrams (sequence of three words, with or without morphological description) and their corresponding lemmas. In this last scenario, lemmas were predicted either for the entire sequence of three words or only for the word in the middle of the trigram. The analyzed data is part of two Romanian corpora: the Romanian Explicative Dictionary (DEX)[4] and the CoRoLa corpus [185].

2.2.2 Related work

In recent years a large number of research papers have analysed the lemmatization problem using the deep learning strategies. Table 2.6 and the following related work discussion summarizes the different studies in the Romanian lemmatization field. However, due to the different datasets, data versioning and experimental details, the results are not directly comparable.

In [200] Boroş implements a framework based on a perceptron-like algorithm with a margin-dependent learning rate (MIRA [209]) and solves NLP tasks such as syllabification, lemmatization, phonetic transcription and lexical stress prediction

[4]https://dexonline.ro/

applied on Romanian language. For lemmatization the overall **accuracy** was equal to **94%** (initially, the accuracy was computed separated for every POS class).

Chakrabarty et al. [201] use two successive BiLSTM structures to perform a context sensitive language independent lemmatization. The first BiLSTM network extracts the character level dependencies, while the second network learns contextual information for the given word. For Romanian an **accuracy** of **94.32%** was obtained.

The NLP-Cube[5] framework described in [202] performs sentence splitting, tokenization, lemmatization, tagging and parsing for 82 languages. For the lemmatization task the system is composed of a bidirectional LSTM encoder and a simple LSTM decoder leading to a **94.79% accuracy** when applied to Romanian language.

In [203] Yildiz et al. present *Morpheus*, a lemmatizer and morphological tagger. The tools follows the encoder-decoder architecture, using LSTM cells and dedicated decoder for each task. The lemmatizer used an extra bidirectional LSTM to encapsulate the word's context. Unlike most lemmatizers which directly predict the characters of the word's lemma, *Morpheus* also outputs the minimum edit operations between the word and its lemma. For Romanian language, *Morpheus* achieves an **accuracy** of **97.88%** for edit operations and **96.54%** when predicting characters. The Romanian-RRT dataset[6] was used.

Chrupala et al. [204] describe Morfette, a modular system for morphological tagging and lemmatization, based on searching algorithms, the shortest sequence of instructions (shortest edit script - SES [210]) and Maximum Entropy classifiers. The two modules can be used together to improve the input information (one's input uses the other's output) or separately. The lemmatizer's input consists of word-lemma pairs enriched with lexical features (prefixes and suffixes of a certain length, predicted POS, spelling pattern). Morfette is analysed on three morphologically rich languages: Polish, Spanish and Romanian. For the Romanian language the lemmatizer was trained on the MULTEXT-EAST[7] dataset obtaining an **accuracy** of **97.78%**.

Qi et al. [205] present *Stanza*[8], a Python multilingual NLP tool, processing 66 languages. To predict the lemma, *Stanza* uses a pair of a dictionary-based and a sequence-to-sequence lemmatizers with LSTM cells and attention mechanism. For the Romanian language, an accuracy of **97.95%** was obtained for the Romanian-RRT dataset.

Kanerva et al. [206] use a two BiLSTM layered encoder enriched with learned character and POS tag embeddings and a decoder composed of two unidirectional LSTM layers with an input feeding attention on top of the encoder's output. The system was trained for 52 different languages. The **accuracy** of the Romanian lemmatizer was **98.25%**.

In [207] Straka et al. compare the contribution of the three conceptualized embeddings (BERT[211], Flair[212] and ElMo[213]) within a LSTM based system enriched with word embeddings (WE) and character-level word embeddings (CWE). The analyzed tasks were POS tagging, lemmatization and dependency parsing for a number of 54 languages. For the particular task of Roamnian lemmatization, the Romanian-RRT dataset was chosen. The highest **F1-score** of **98.59%** was achieved when both the BERT and the Flair embeddings were used.

[5] https://github.com/adobe/NLP-Cube

[6] https://universaldependencies.org/treebanks/ro_rrt/index.html

[7] http://nl.ijs.si/ME/

[8] https://github.com/stanfordnlp/stanza

TABLE 2.7: The following family of words preserves the stem, while the lemma differs.

Word	Lemma	Stem
copilașul = infant	copilaș	
copilandru = youth	copilandru	
copilărie = childhood	copilărie	copil = child
copilări = to childhood	copilări	
copiliță = little girl	copiliță	
copilărește = childishly	copilăresc	

In [208] Dumitrescu et. al describe an universal morphological reinflection system based on an attention-free encoder-decoder neural architecture with a bidirectional LSTM for encoding the input sequence and a uni-directional LSTM for decoding and producing the output. This architecture is evaluated on the SIGMORPHON 2018 [214] dataset, with data from 86 languages. For the Romanian language the **accuracy** of the **reinflection** task was **88%**.

2.2.3 Lemmatization - theoretical background

Lemmatization is the process of determining the word's dictionary form, called lemma. In the linguistic fields, through lemmatization, all flexional forms of a word are grouped together to be analysed as a single entity.

The lemmatization is language dependent and adheres to certain rules. In Romanian, the lemma of a noun the masculine singular nominative, while a verb's lemma is the infinitive form.

Noun *copilandre (plural,feminine)* → copilandru (singular, masculine) = youth
Verb *merg = (I) go, mergeam = (I) went, mersesem = (I) had gone* → (a) merge (infinitive) = (to) go

In contrast to stemming, which returns the part of the word that never changes even when different forms of the word are used (the stem), lemmatization depends on the word's meaning or context and on the morphology of the word (Table 2.7).

Although the stemming algorithms are faster and easier to be applied in word searching applications, they have lower accuracy in the case of homonyms (words spelled the same but with different meaning). For instance, the word *ouă* (*En: eggs)* has the same stem *ou* (*En: egg*), in both of the examples below, regardless the part of speech:

Noun Am cumpărat patru **ouă**. (*En: I bought four eggs.*)
Verb Găinile **ouă**. (*En: The hens lay eggs*)

while the lemma differs based on the word's meaning and part of speech:

Noun ouă → ou (*En: egg*)
Verb ouă → (a) oua (*En: to lay eggs)*

Ambiguous words

Based on semantical and on morphological contexts, the lemmatization methods can be grouped in two major categories:

- *context-aware methods* - the system knows information regarding the context in which the word appears (sentence context - for meaning, POS tag- for morphological information, etc.)

TABLE 2.8: The distribution of samples for each POS category for DEX dataset (left) and CoRoLa dataset (right).

POS - DEX dataset	Abreviation	Percentage
Noun	n	41.28
Adjective	a	28.93
Verb	v	28.00
Unique form	u	1.11
Invariant	i	0.57
Pronoun	p	0.10

POS - CoRoLa dataset	Abbreviation	Percentage
Noun	n	50.08
Verb	v	26.77
Adjective	a	19.00
Adverb	r	2.27
Pronoun	p	0.54
Apposition	s	0.32
Numeral	m	0.30
Conjunction	c	0.15
Hyphen	@	0.18
Abbreviation	y	0.17
Determiner	d	0.16
Article	t	0.05
Particle	q	0.01

- *individual word-based methods* - the system predicts the lemma knowing only the given word, without any additional information

The advantage of the first approach is the higher accuracy rate when predicting lemma for ambiguous words, as the system can benefit from the contextual information provided. The second strategy can be improved by listing all the possible lemmas of the given words. In this way, if the predicted lemma belongs to the word's lemmas list, then it will be considered correct, as the system's total unawareness of the context was taken into consideration.

Being a rich inflectional and morphological language, the Romanian ambiguous words form a consistent category which is challenging for the lemmatization task. The ambiguity can appear either between different POS classes (as an adverb, *poate* = *maybe* has the associated lemma "poate", while, as verb *poate* = *(he) can* with "putea" as lemma) or within the same POS class (the plural noun *torturi* means both *cakes* and *tortures*, depending on the pronunciation; thus it has two lemmas - *tort* and *tortură* respectively).

2.2.4 Experimental setup

To train the systems presented in [11], we used two different datasets. The first dataset is the Romanian Explicative Dictionary[9](ID: **DEX**) which contains 1.158.194 word forms, each associated with its lemma and part of speech tag. The words are clustered in six major categories based on the part of speech: nouns, adjectives, verbs, pronouns, invariables (adverbs, proper names) and unique forms (interjections, archaic words, Latin names). Table 2.8 contains the distribution of samples for each POS category.

The second dataset is the CoRoLa[10] corpus [185] which contains texts from different functional styles: belletristic, scientific, publicistic, official. In this work the belletristic subset was chosen. It contains 51043 sentences with approx. 1 million tokens and 63.194 unique words. Each token belongs to one of the thirteen POS categories: noun, verb, adjective, adverb, pronoun, apposition, numeral, conjunction, hyphen, abbreviation, determiner, article and particle.

[9]https://dexonline.ro/
[10]http://CoRoLa.racai.ro/

TABLE 2.9: Datasets description

Dataset	Number of samples		
	Total	**Training**	**Testing**
DEX	1.158.194	926.556	231.638
Corolla Belletristic - trigrams	753.490	602.792	150.698
Corolla Belletristic - one word	76.417	61.134	15.283

Several pre-processing steps were performed and include the following operations:

- converting text to lowercase
- striping the digits and punctuation
- splitting sentences into words (for the CoRoLa dataset)
- creating pairs of input (the word) - target (corresponding lemma) sequences
- appending a start-marker ("\t") and an end-marker ("\n") to the target word

Both datasets were subsequently split into disjoint training (80%) and testing (20%) sets (Table 2.9), each of them being individually shuffled before the splitting. For the experiments evaluated in this study, the datasets were split at lemma level, ensuring that the same lemma do not appear both in train and test.

Input Data

The neural networks were fed with pairs of (word-lemma). The input data is one-hot encoded, thus the words become bidimensional MxN sparse matrixes, where M is equal to the longest word's length and N is the set of characters that forms all the sequences (number of letters which forms the dataset's alphabet).

First, the systems were forced to learn as much information as possible only from simple pairs of (word-correspondent lemma) (*Input type: word*). Then, the part of speech (POS) for every single word from the DEX dataset was appended (*Input type: word + POS*). During the training, the input consists of the word and the paired POS and the system predicts the corresponding lemma: (word|POS $\rightarrow$ lemma).

Example: (strugurelui| n $\rightarrow$ strugure)
En. grape's|n (n=noun) $\rightarrow$ grape

In the same idea of adding relevant contextual information, for each word, a context was added, by using a window size of three neighboring words (*Input type: trigrams*) thus obtaining trigrams. For this scenario, as the DEX dataset contains only isolated words, only the belletristic subset of the CoRoLa corpus could be used. An example of an input-output processed sentence is given below.

Dar nimeni nu venise în urma ei. (En. But no one had come after her.)

$\Downarrow$

Input sequence	**Target sequence**
dar nimeni nu	dar nimeni nu
nimeni nu venise	nimeni nu veni
nu venise în	nu veni în
venise în urma	veni în urma
în urma ei.	în urma el

For the trigrams scenarios, two strategies have been tried. First the systems predicted the lemmas for the entire sequence of three words. By analysing the results, it was observed, in most of the cases, the lemma was correctly predicted for only two out of the three words. Thus, the systems were trained to predict the lemma only for the word from the middle (*Input type: trigrams middle-lemma*).

Architectures

For the sequence-to-sequence lemmatization task three different neural network architectures were tested. All the systems were trained for 300 epochs. The code was implemented in Python using the functional Application Programming Interface (API) of Keras toolkit[11] with a TensorFlow backend[12] and run on a system with 4 GPUs (Nvidia GeForce RTX 2080 Ti, 11GB memory).

In the first experiments, both the encoder and the decoder contain a single LSTM layer. Based on initial tests, the batch size was set to 512 and the latent dimension of the hidden layers to 256. Figures 2.5a illustrates the system's architecture. Aiming to improve the system accuracy, the encoder and the decoder were enriched with one or two additional LSTM layers resulting a stacked LSTM hierarchy described in Figure 2.5b. Based on initial tests, the system was trained using a batch size of 256 and a latent dimension of 128.

In a last scenario a convolutional architecture with attention module was implemented. Both the encoder and the decoder are composed of 3 convolutional layers with 128 feature map and use the ReLU activation function. The decoder is followed by an attention structure with a softmax activation. The decoder's output is passed through 2 additional convolutional layers with a softmax activation. A final dense layer provides the system's output. Figure 2.5c describes the network's architecture. A batch size of 512 was chosen after preliminary tests.

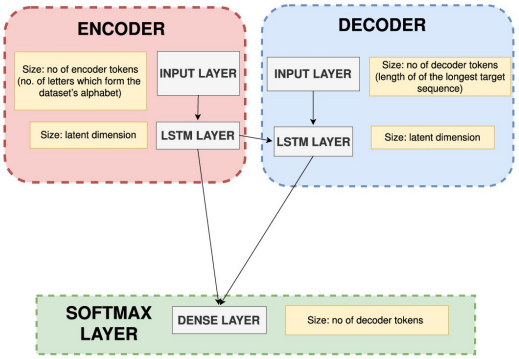
2.2.5 Results and Discussions

Each architecture described in Figure 2.5 from [11] is trained using each dataset individually, with and without additional POS annotation. For the CoRoLa dataset, the trigram scenario was also analysed, thus, it resulted 18 systems trained using data from CoRoLa (see Table 2.11c) and 6 systems trained over the DEX dataset (see Table 2.11a). All the 24 systems were evaluated with the *classification accuracy metrics*, which is expressed as the ratio between the correct predicted items out of the total samples. The accuracy was measured at different levels: trigram (CoRoLa subset), word and character level (for both datasets).

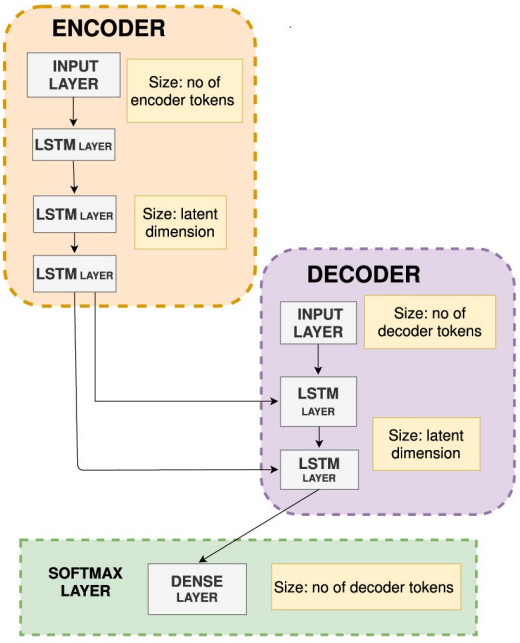
Ambiguous words. In order to solve the ambiguity problem of words with multiple lemmas, a dictionary of accepted lemmas was built for each dataset. More precisely, we paired each word with a set of lemmas, as illustrated in Table 2.10. During evaluation, if the predicted lemma belongs to the given word's lemma dictionary then it was considered to be correct. The lemma dictionary is necessary even when POS context is added as ambiguity can exist within the same POS class when no additional morphological information is given (genre, case or verb tense), as illustrated in Table 2.10c.

The results for the DEX dataset are described in Table 2.11a. The one layer LSTM based architecture achieved the highest accuracy rate at both word and character

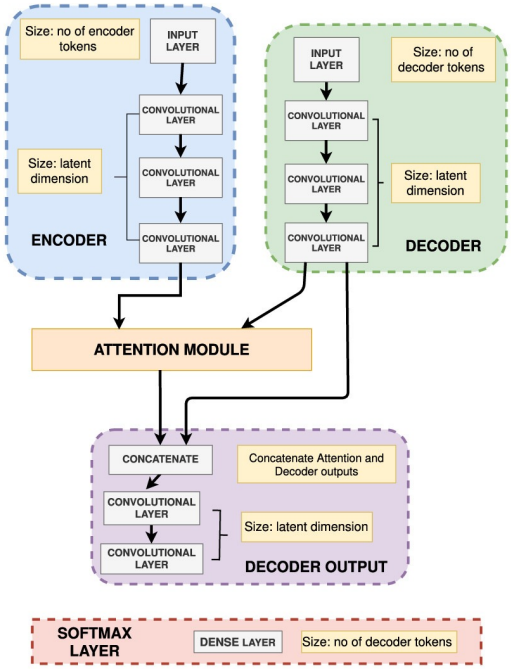
[11] https://keras.io/
[12] https://www.tensorflow.org/



(A) **The LSTM model architecture**. Both the encoder and the decoder are composed of one LSTM layer. The size of the encoder's input layer is given by the length of the dataset's alphabet. The size of the decoder's input layer is equal to the length of the longest target sequence. The output of the decoder is passed to a dense layer with a softmax activation.



(B) **The LSTM stacked model architecture**. The encoder is composed of three LSTM layers. The output of the third layer of the encoder is passed to both LSTM layers of the decoder. The result is fed to a dense layer with a softmax activation.



(C) **The CNN model architecture**. The encoder and the decoder contain an input layer and three convolutional layers. The outputs from both the encoder and the decoder are passed to an attention module. The obtained output is concatenated again with the decoder's output and fed to two convolutional layers. A fully connected layer with a softmax activation is applied to compute the final output of the network.

FIGURE 2.5: Sequence to sequence model architectures for the lemmatization task

TABLE 2.10: Illustrating the dictionary of accepted lemmas

Word	Meaning	Lemma
Torturi	Birthday cakes (Noun. pl.)	tort
Torturi	Torments (Noun pl.)	tortură

(A) same POS

Word	Meaning	Lemma
Nouă	Somenthing new (Adj. fem)	nou
Nouă	Nine (Numeral)	nouă

(B) different POS

Word	Dictionary of lemmas
nouă	nou nouă
torturi	tort tortură

(C) Dataset entry

level. When additional POS information was provided, the system's accuracy increased by 3.39% at word level and by 2.14% at character level.

Unlike the DEX dataset, for the CoRoLa belletristic subset, the system using the CNN layers [11] performs better at word and trigrams level (Tables 2.11b and 2.11c). At character level, the stacked LSTM network achieves the best results.

One explanation may be the amount of training data: the DEX dataset contains over 900.000 samples compared to only 60.000 samples in the CoRoLa subset. Even if only a small amount of data is available, the CNNs are faster as they are using attention layers and kernels to simulate the context and the recurrence of the data.

On the other hand, the LSTM architecture presented in [11] not only fails to predict the correct lemmas, but it also outputs more lemmas than necessary. For instance, in 13% of the trigrams cases, the LSTM based system predicts the appropriate lemmas for the first one or two words from the input sequence while the rest of the output is series of arbitrary lemmas. This means that the end-sequence marker ("\t") is not correctly predicted. The POS additional information helps all the networks to learn the context from the data, thus the accuracy rate increases by almost 5%.

2.2.6 Conclusions and future work

In [11] we analysed 24 systems based on deep neural networks in the context of the Romanian lemmatization. The systems were trained on labelled pairs of words and the corresponding lemmas, using at most the part-of-speech tag as morphological information. The input data was one-hot encoded and then passed through the encoder-decoder-based architectures, within a sequence-to-sequence approach. The scope of this study was to use as few lexical input information as possible, as the analysed language offers few corpora with completed annotated texts. Apart from fine tuning the network's hyperparameters during training or enriching the input text's metadata, humans can not interfere to improve the learning process in this end-to-end scenario.

As future work, inspired by other studies in the lemmatization task, more types of encoding, such as word embeddings or contextualized embeddings (BERT), will be investigated. In order to leverage the learning process, in [11] the input data encapsulated the lexical and semantic context using n-grams with a sliding window of three neighboring words. Other words window sizes (of five, seven, etc.) will be investigated and their input within the systems will be analysed. Besides the CNN and the LSTM layers, other types of neural networks will also be explored, such as the bidirectional LSTMs, the gated recurrent units (GRU), or only attention based architectures, which are already frequently applied in other text processing tasks.

TABLE 2.11: Network parameters and accuracy for each dataset. Best results are marked in bold. At word level, the columns *with ambig. lemma* and *without ambig. lemma* refer to the same target data but check the dictionary created for the words with multiple lemmas.

(A) The **DEX** dataset.

No.	Input type	Architecture	Latent dimension	Batch size	Accuracy		
					word level		char level
					with ambig. lemma	without ambig. lemma	
1	word	LSTM	256	512	88.20	**95.93**	**97.29**
2		LSTM_stacked	128	256	88.30	94.64	97.17
3		CNN	128	512	90.36	95.83	92.82
4	word + POS	LSTM	256	512	98.10	**99.32**	**99.43**
5		LSTM_stacked	128	256	97.05	98.07	99.12
6		CNN	128	512	97.39	98.36	98.40

(B) The **CoRoLa** dataset with **one word** as input.

No.	Input type	Architecture	Latent dimension	Batch size	Accuracy		
					word level		char level
					with ambig. lemma	without ambig. lemma	
7	word	LSTM	256	512	68.03	75.00	88.38
8		LSTM_stacked	128	256	72.08	79.64	**90.68**
9		CNN	128	512	78.51	**86.07**	84.17
10	word + POS	LSTM	128	512	76.77	77.36	89.76
11		LSTM_stacked	128	256	86.55	87.22	**95.05**
12		CNN	128	512	90.72	**91.35**	94.23

(C) The **CoRoLa** dataset with **trigrams** as input

No.	Input type	Architecture	Latent dimension	Batch size	Accuracy				
					trigram level		word level		char level
					with ambig.	with ambig.	with ambig. lemma	without ambig. lemma	
13	trigrams (middle-lemma)	LSTM	256	512	88.69	92.49	97.17	98.58	96.97
14		CNN	128	512	89.56	94.15	98.01	**99.69**	94.82
15		stacked_LSTM	128	256	75.79	79.76	92.12	93.77	91.70
16	trigrams (middle-lemma)	LSTM	256	512			93.86	95.21	93.28
17		CNN	128	1024	N/A	N/A	97.07	**98.32**	**96.74**
18		stacked_LSTM	128	256			96.61	95.28	94.86
19	trigrams + POS	LSTM	256	512	62.48	62.84	85.15	85.33	85.05
20		CNN	128	512	95.86	96.49	98.87	**99.09**	97.54
21		stacked_LSTM	128	256	93.31	93.88	97.71	97.91	**97.62**
22	trigrams + POS (middle-lemma)	LSTM	256	512			98.07	98.07	98.43
23		CNN	128	1024	N/A	N/A	98.52	98.52	98.11
24		stacked_LSTM	128	256			98.83	98.83	**98.78**

Although the transformer architectures gained popularity in solving NLP tasks for English, Mandarin or other rich resourced languages, we have to analyze the impact on the training process of the limited amount of data if we want to apply these systems to Romanian language.

2.3 Automatic Romanian Part of Speech tagging

In this subchapter we address the issue of automatic Part of Speech tagging for Romanian words using LSTM networks. The method was introduced in the original research paper [13].

2.3.1 Motivation

Besides diacritics restoration and lemmatization, another important task in the NLP field is the part of speech (POS) tagging. It means to determine the part of speech of a given word, often enriched with morphological or syntactical information.

Depending on the annotation level, in the Romanian language we can discriminate three types of tagsets (as exemplified in Table 2.12).

Input	Copilăria
RPOS	N
MSD	Ncfsry
CTAG	NSRY

TABLE 2.12: The tagsets illustrated for the word *Copilăria* (en. Childhood)

- RPOS - The simplest one is the root POS (RPOS), which refers to identify only the part of speech of the word (noun, adjective, verb, adverb, preposition, conjunction, numeral, article, interjection, pronoun) and can be extracted from the dictionary entry of the word's lemma.
- MSD - The Morpho-Syntactic Descriptions (MSD) [215] contains more grammatical information, depending on the part of speech of the word. For example, if we analyse a noun, the MSD tagset offers information about the type (common or proper), number (singular or plural), gender (feminine, masculine or neuter), case (nominative, genitive, accusative, dative, vocative, direct, oblique), definiteness and clitic.
- CTAG [216] (C-tagset)- contains maximum 3 additional information to RPOS, preserving only the ambiguous characteristics from the MSD tagset. For instance, if we have to analyze a noun, the case and the number are sufficient to describe the word, as the other descriptors (gender, type, definiteness or clitic) can be recoverable form the word's form.

There are two major difficulties when we want to determine the POS of a word. The first one refers to the homographs, the words which share the same written form but have different meanings in different contexts. The second problem refers to rich inflectional languages, which is the case of Romanian: declination and inflections of the word are not regular. Thus, to determine the POS, hand-crafted rules or contextual information are needed

2.3.2 Related work

The POS tagging is essential in tasks like machine translation, textual information extraction, speech recognition or speech synthesis. In order to solve the automatic POS tagging, several tools of text processing have been developed. One first attempt consists on using rules-based or probabilistic methods (Maximum Entropy Classifiers, Hidden Markov Models, Bayesian Networks or Conditional Random Fields), which lead to unsatisfactory results[217] when applied to the rich inflectional languages like Romanian. Thus, the use of the machine learning techniques came as a solution to improve the POS taggers accuracy.

Numerous studies analysed the Romanian POS tagging problem. The results are summarized in Table 2.13 and described in [13].

In [216] Tufis et. al combine a language model (build with a tiered tagging with C-tagset) with a post-processor based on probabilistic methods and reconstruct the MSD tag with an accuracy of 98.39%.

To the best of our knowledge, the first attempt of using neural network in the task of POS tagging is described in [217]. Boroş et. al implement a feed forward neural network combined with genetic algorithms to automatic determine the MSD tagsets of Romanian words. The reported accuracy was 98.19%.

The manual rule construction problem in the task of POS tagging was analysed in [218] and [219]. The authors combined statistical models and rules-based system that classifies the tagging errors.

The BALIE multilanguage system described in [220] uses machine learning techniques and the WEKA framework to predict the POS tags. For the Romanian language, an accuracy of 95.30% was reported.

Using a Naive Bayes model with a word database, Teodorescu et. al [221] obtained an accuracy of 96.12%.

Authors	Method	Accuracy	Tagset
Tufis & Mason [216]	Probabilistic	98.39%	MSD
Boros & Dumitrescu [217]	Deep Neural Networks	98.19%	MSD
Simionescu [219]	Probabilistic & Rule-based	97.03%	MSD
Teodorescu et al. [221]	Probabilistic	96.12%	Root POS
Frunza et al. [220]	Machine Learning	95.30%	Root POS

TABLE 2.13: POS-tagging accuracy results for Romanian reported in the literature

Having these studies as a base for the Romanian POS tagging, in [13] we analysed the use of neural network in predicting the word's POS. Precisely, inspired by other works [222], [223] which applied the LSTM in the POS tagging for foreign languages, we investigated the use of this type of network for Romanian. These previous research studies demonstrated that LSTM not only is applicable for low resourced languages, but it also is a proper candidate for determining the correct POS when extended tagsets exist. In [13] we compared the LSTM networks with a sequence-to-sequence architecture, also based on LSTM cells.

2.3.3 Experimental setup

Systems and architectures

In [13] we analysed the use of **LSTM networks** for the task of predicting the POS of a given word. As described in the section 1.3, the advantage of this type of architecture lays in the possibility of using the information from the previous step of

learning, which is essential in the text processing tasks, as the words and the letters are interconnected. In [13] the encoded input data is passed through a LSTM layer and the result is processed by two stacked dense layers, as illustrated in Figure 2.7a. The second dense layer is the output layer and has as many nodes as the number of possible POS tags. The latent dimension of the LSTM layer was chosen based on initial tests and is between 64 and 1024.

Besides the LSTM architecture, a **sequence-to-sequence model** was implemented for the task of predicted the MSD tag. Both the encoder and the decoder are composed of LSTM layers. The system's architecture is illustrated in Figure 2.7b. The sequence to sequence architecture's flow is illustrated in Figure 2.6 and is described in section 1.3.

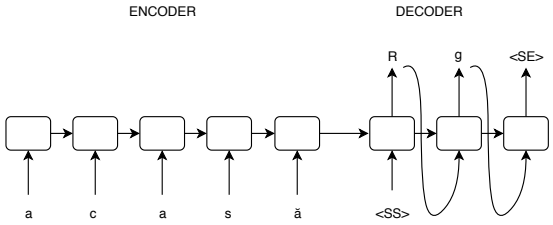


FIGURE 2.6: Sequence-to-sequence model example: the system predict the CTAG for word *acasă*

The systems learns from pairs of input-target data, both obtained form initial texts, divided in sequence of words with a sliding window of 3 to 5 words. An input example is given below, where # and @ are the start and the end characters of the target sequence:

Input sequence: 'Flori pentru mama'
Target sequence: '#NSN S NSY@'

All the architecures and experiments were implemented using Python 3.7 using Keras library[13] with Tensorflow backend[14].

POS tagsets

In [13] all the experiments were run for the Romanian language. We analysed the all three tagsets: RPOS, MSD and CTAG, described in Tables 2.14a and 2.14b.

Tagset	No. of tags used
Basic	13
MSD	334
CTAG	89

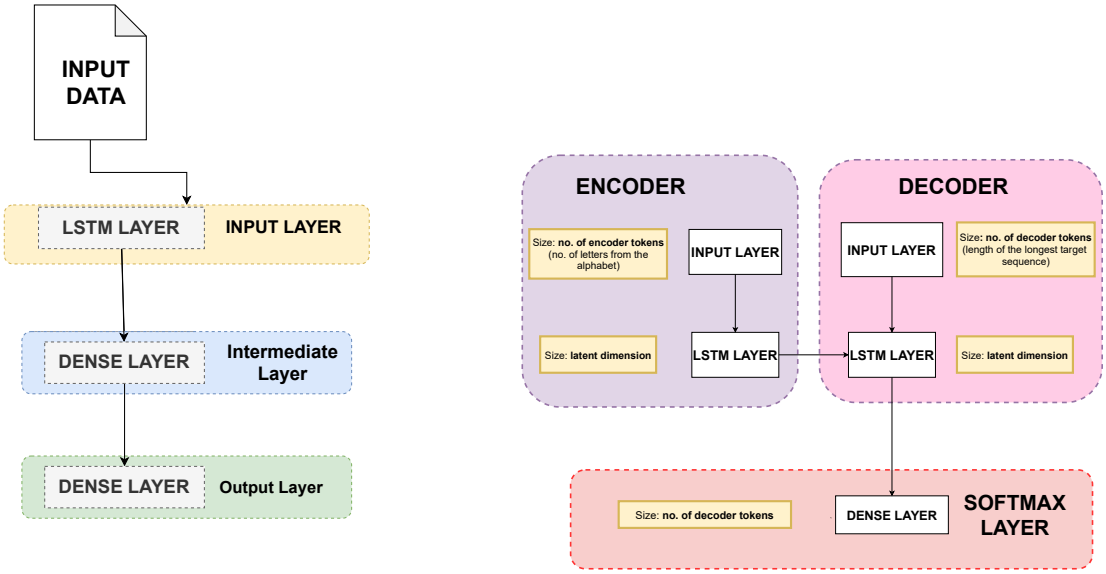
(A) Number of tags per tagset

Afpms		**Ncfpry**	
A	Adjective	N	noun
f	qualificative	c	common
p	positive	f	feminine
m	masculine	p	plural
s	singular	r	nominative
		y	definite

(B) MSD tag examples for an adjective (*plin* - en. full) and a noun (*mişcările* - en. moves)

TABLE 2.14: Illustrating the POS tagsets

[13]https://keras.io/
[14]https://www.tensorflow.org/



(A) LSTM with dense layers model (B) LSTM sequence-to-sequence model

FIGURE 2.7: Systems architectures for the task of POS tagging

Datasets

The systems presented in [13] were trained using three different datasets. The majority of the experiments were performed using the the Simionescu's dataset [218] (**WPT**[15]), developed based on the DexOnline database[16] and Wikipedia[17]. For the RPOS prediction, we trained the systems using the first letter from the MSD tagset provided in **WPT**. For the words with multiple tags, we analysed two scenarios: firstly, we considered as a correct output any POS tag of the word (systems marked with an asterisk in Table 2.16), then we have looked only at the tag of the first occurrence of the word.

We also used the basic DexOnline dataset (**DEX**), which pairs each word with the root POS and a word frequency. We took into account only words with positive frequency.

For the CTAG and the MSD prediction tasks we trained the systems with the CoRoLa (**CoRoLa**[18]) dataset, as it contains full texts, thus providing linguistic and contextual information. We used the entire dataset, with five different styles: juridical, scientific, belletristic, memorialistic and publicistic.

For the training and testing steps, all three datsets all randomly splitted, using 20% of data for testing. All three datasets are summarised in Table 2.15.

[15]http://nlptools.infoiasi.ro/WebPosTagger
[16]https://dexonline.ro/
[17]https://ro.wikipedia.org/
[18]https://corola.racai.ro/

TABLE 2.15: Number of training and test samples per dataset

Dataset	Total samples	No. of training samples	No. of test samples
WPT	1,715,881	897,328	224,331
DEX	1,994,412	936,611	234,152
CoRoLa	3,075,165	2,460,132	615,033

Input data encodings

For the experiments conducted in [13] we chose to use only the one hot encoding(**OHE**) and the letter encoding(**LE**) styles. We did not opt for the word embeddings, as we were interested in predicting the POS tags based only on the orthographic form of the word, without encapsulating the context. For the Letter embedding style, we used the Gensim library[19] to create a letter embedding of order 30, based on Romanian Wikipedia database.

2.3.4 Results and discussions

In [13] the performance of the systems was measured using the accuracy metric (as ratio of correct predicted output over the total number of samples). We trained the networks over various epochs (from 25 to 100) with different batch sizes (from 256 to 1024) and latent dimensions of the LSTM layers (from 64 to 1024). The parameters' values for which the systems obtained the best results, as well as the corresponding accuracy, are highlighted in Table 2.16.

Unlike the System ID 2, which looks only at the RPOS of the first occurrence in the dataset of the word, the System ID 1 uses all the RPOS of the word and achieves higher accuracy (increased with 4.33%). The explanation is the context has a major contribution in the correct prediction of the RPOS: the same word can have different POS in different contexts. With no information regarding the meaning of the word, it becomes difficult and ambiguous to determine the appropriate POS.

The systems trained to predict the RPOS obtained better accuracy than the ones trained for the MSD or the CTAG tasks. The cause is the number of MSD possible tags (over 330) which the systems have to learn and predict, compared to only 13 RPOS tags. The results obtained in [13] by the System ID 6 are comparable with the ones from [216], although the latter used context information.

Regarding the input data embedding, we observed that changing the embedding type did not impact the learning process. One explanation may be the small size of the input datasets, as the layer embedding may need more data in order to learn a proper representation of it. Another remark is the letter position within the word is not discriminating when identifying the word's POS, thus the letter embedding used in the System ID 3 and the System ID 4 did not increased the accuracy.

2.3.5 Conclusions and future work

For the task of POS prediction of the Romanian words, in [13] we compared two types of architectures: the LSTM-based networks and the sequence-to-sequence architecture based on LSTM layers. When predicting the MDS and the CTAG tagsets, the sequence-to-sequence approach obtained better results than the simple LSTM-based architectures. Different types of encoding the input data have been tested,

[19]https://radimrehurek.com/gensim/

TABLE 2.16: Network parameters and accuracy results

System ID	Dataset	Tag	Network type	Character encoding	Latent dimension	Batch size	Epochs	Accuracy
1	WPT	RPOS	LSTM + Dense (*)	OHE	256	512	50	**99.18%**
2	WPT	RPOS	LSTM + Dense	OHE	256	512	50	94.85%
3	WPT	RPOS	LSTM + Dense	LE	256	256	25	54.80%
4	WPT	RPOS	seq2seq LSTM	LE	256	256	25	94.99%
5	WPT	RPOS	seq2seq + Embedding layer	OHE	256	256	20	93.88%
6	DEX	RPOS	LSTM + Dense	OHE	256	512	50	94%
7	WPT	MSD	seq2seq LSTM (*)	OHE	512	1024	50	98.25%
8	WPT	MSD	seq2seq LSTM	OHE	512	1024	50	75.28%
9	WPT	MSD	seq2seq + Embedding layer	OHE	256	512	50	76.62%
10	CoRoLa	CTAG	seq2seq LSTM	OHE	256	512	100	97.15%

with no major impact for the learning process. As future work, we intend to analyse other types of neural networks (convolutional, attention mechanisms) especially for predicting the extended MSD and the CTAG tagsets. As regarding the context information, adding linguistic features as lemma and lexical stress may improve the systems' performance. Moreover, we should consider using the contextual word embeddings such as BERT.

Chapter 3

Medical text data processing

In this subchapter we discuss methods and algorithms from machine learning applied to a different domain: medical data.

3.1 Topic modelling for identifying medical diagnostic

First, in [2] we applied a topic modelling approach in order to help physician to find a diagnostic by extracting meaningful information from patients' medical records.

3.1.1 Motivation

When it comes to evaluate patients' health, medical doctors analyse different aspects of the person's life, in the so called anamnesis process. Previous diagnostics, family antecedents of a certain illness, different symptoms declared by the patient, together with the personal background and lifestyle contribute to establish an appropriate diagnostic, thus to prescript an adequate medical treatment plan.

Recent advances in the machine learning field with respect to automatically processing written text may ease the work of medical physicians. This leads to a more accurate medical process, with fewer human errors caused by the professional fatigue, the stress of working against the clock with other awaiting patients, and so on and so forth. Machine learning algorithms are already applied in different aspects of the medical process:

- Classification algorithms - to label the patients' disease [224], [225], [226], [227], [228].
- Clustering - grouping together similar medical cases in order to study patterns of the diseases' evolution. The information used can be both written texts or medical images [229], [230], [231], [232].
- Topic modelling - extracting meaningful information from the written medical observations, as clusters of words which are close in meaning, to discover hidden diagnostics repeatedly occurring within a certain collection of observation [233].
- Recommendations - to automatically recommend medical treatment [234], [235], [236].
- Anomaly detection - detecting the outliers within patients medical situation, in order to determine if special treatments, observations or actions are needed [237], [238], [239].

- Predictions - make prognosis based on the available data and similar trends in the patients' health condition [240], detect the absenteeism in future appointments [241], predict healthcare costs [242].
- Automation - parts of the medical process (data entry, medical appointments, inventory managements, etc.) can be automatically done using the machine learning algorithms (a review of 100 papers in the field of automated machine learning applied in medical care can be read in [243]).
- Ranking - to rank the medical databases, by putting the relevant content first [244], [245]

For the experiments introduced in [2], we have focused on topic modelling, or extracting meaningful information from the written texts, by classifying or grouping together the texts within a certain subject or theme, based on the contained main idea. A set of written medical observations of 102 patients was separately grouped as an attempt of automatic diagnostic prediction, based on the topic modelling techniques from machine learning. The texts are written in English and consists in clinical observation.

3.1.2 Related work

In recent years, a large number of research studies analysed the application of the topic modelling algorithms in the field of medical data.

In [246] Bhattacharya et al. applied Latent Dirichlet Allocation to identify the patterns of associated co-occuring conditions. They analysed over 13 000 patients with diseases in kidney function. Instead of using the words as the basic discrete unit for the algorithm, they used the diagnosed conditions in SNOMED codes [1]. In terms of qualitative evaluation, the authors of the study analysed the medical relevance of the results, by verifying the medical literature if the most probable conditions within each and every topic are indeed certified to occurs related to the tracked disease. In term of quantitative evaluation, the results were measured using the tightness (each topic can be expressed using a small number of conditions) and the distinctiveness (how well the topics are separated one of each other).

In [247] the topic modelling is used to create an automated method for suggesting the similarities within patients and applicable diagnostics. The authors search for similarities between two systems: the hospital information system (diseases diagnosed by the medical doctor - disease corpus) and the laboratory results assigned to a certain set of autonomous patients (the patients corpus). The study showed that this method can bring additional insight over future diseases the patients may develop in the future, by correlating different patients' symptoms hidden in the patients' clinical history.

Lin et al. propose in [248] a top-down binary hierarchical topic model (biHTM) for biomedical literature. Their heuristic method consists in applying a LDA model and adaptively (with few hyperparameters) processing the subtrees of the hierarchy. The method is applied to a bibliographical dataset of biomedical information (articles from medicine, nursing, pharmacy, dentistry, veterinary medicine, and healthcare). The authors proved that the biHTM can quickly learn topic hierarchy without

[1] NIH U.S. National Library of Medicine `https://www.nlm.nih.gov/healthit/snomedct/index.html`

using latent variables. This method was compared with hierarchical LDA and Hierarchical Latent Tree Analysis (HLTA) in terms of efficiency, topic quality and interpretability and proved to be suitable to process large amount of biomedical data (25 million abstracts from 5639 selected medical publications).

3.1.3 Experimental setup and results

For the experiments run in [2] we used the medical records taken by a medical physician. The set contains 102 instances, each representing a patient with the clinical observation, the current and past treatments and the patient's response to the treatment. Thus, as type of data, we worked with text (the clinical observation and the prescribed treatment, both in English) and numbers (patient's response to treatment encrypted from 1 = non-responsive to 5 = very responsive). In order to comply with GDPR policies, all the patients' personal information (names, addresses) have been suppressed by the physician before giving us access to the data.

Few pre-processing steps were required to prepare the dataset for machine learning algorithms. To generate the relevant textual features, we used word's frequencies, as defined below:

- Term Frequency (**TF**) counts the frequency of a word w in a document d, as a ratio between the number of w occurrences in d divided by the total number of words in d. We have to mention that it weights the words only based on the number of occurrences and does not contain any semantic meaning.
- Inverse Document Frequency (**IDF**) measures the amount of information provided by a given word across the document. IDF is the logarithmic scaled inverse ratio of the number of documents that contain the word and the total number of documents.
- Term Frequency-Inverse Document Frequency (**TF-IDF**) normalizes the document term matrix. It is the product of TF and IDF. A word with high TF-IDF has many occurrences in the given documents and must be absent from the other documents. In this case, the word must be a signature word.

To model the topics present in the analysed texts, we used the Latent Dirichlet Allocation (LDA) and the Latent Semantic Indexing. The LDA algorithm is a Bayesian hierarchical probabilistic generative model which assumes that each document is a discrete distribution over multiple topics and each topic separately is seen as a discrete distribution over words, seen as tokens. In order to assign the topics to each documents, LDA first assumes a number of k topics within the researched documents, topics which are distributed across each document, by assigning a topic to each word. Then for each word w within a certain document we supposed the chosen topic is wrong, but the topics selected for the all other words are correctly selected. The new topic is assigned to the word w in a probabilistic way, based on two criteria: the existing subjects within the considered document and the number of times the analysed word is associated with a certain topic across all the documents in the dataset. We repeated this process for a number a times for each word, until we reached a stable state in which the topics allocation does not change any further. The distribution of the topics within the documents is determined from the topic allocations.

For the present topic modelling system[2] the first step was to classify the texts using the TF-IDF. The obtained model was fit using 80% of the data, while the predictions were made for the rest of 20% of the dataset. In order to apply the LDA for

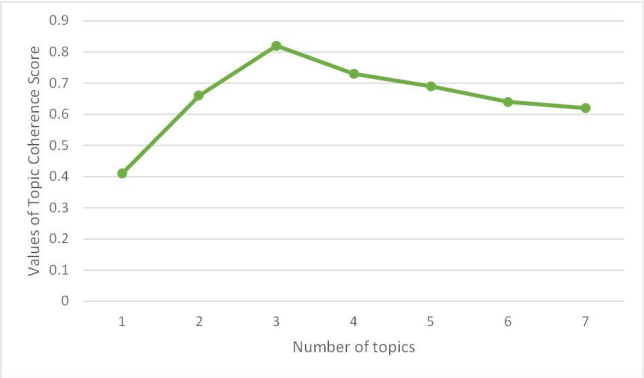


FIGURE 3.1: Values of topic coherence score for different number of topics [2]

topic modelling, we needed an input corpus and a dictionary. We used the Natural Language Toolkit (NLTK)[2] from Python, as it contains most of the algorithms and the functions needed to process the unstructured text: sentence and word tokenization, removing stop words, stemming, lemmatization, POS tagging. With the help of NLTK library we matched each word with an unique ID. Then, we mapped each word ID with the word's frequency to obtain the desired corpus.

As LDA is an unsupervised method, we did not know apriori the number of topics to pass to the algorithm. Thus, we analysed different number of topics as input and we compared the results using the topic coherence score. The values of the topic coherence score for different numbers of topics are illustrated in Figure 3.1. We can observe that for a number of topics bigger than 3, the curve of topic coherence measure begins to decrease. Thus, 3 topics offer us significant insight of data. In Table 3.1 we extracted the most relevant words for each of the 3 chosen topics.

Topic No.	The most relevant words from each topic
Topic 1	0.2 * ulcer + 0.2 * abdominal pain + 0.1 * migraine + 0.1 * fatigue + 0.1 * vomiting
Topic 2	0.2 * jaundice + 0.2 * fatigue + 0.1 * wight loss + 0.1 *itching + 0.1 * nausea
Topic 3	0.2 * headache + 0.2 * fever + 0.1 * nausea + 0.1 * photophobia + 0.1 * sleepiness

TABLE 3.1: Words relevant from each topic

3.1.4 Conclusions and future work

In the experiments described in [2] we applied machine learning techniques in the field of topic modelling to help the medical physicians in the diagnostic process. Thus, using the Latent Dirichlet Allocation models combined with the text processing tools we clustered 102 medical records into three topics, based on the most relevant words within the documents. This approach is intended to assist the physicians in the process of analysing the patients' medical condition in order to decide the disease and the treatment which fits best.

As future work, we intend to extend this approach to medical records written in the Romanian language, as well as to a bigger database. Furthermore, we intend

[2]https://www.nltk.org/

to analyse different approaches for the topic modelling field, in order to overcome the limitations of the LDA algorithm, such as: questionable efficiency for short texts [249] or disregarding co-occurrence relation over the studied documents [250]. We intend to combine the LDA algorithm with the deep learning techniques, in an hybrid approach, as already applied in [251]

3.2 Personal communication styles analysis

Secondly, in [14] we compared six machine learning algorithms to classify one person's communication style based on psychology questionnaires.

3.2.1 Motivation and related work

In the era of Big Data when written and recorded audio data are available almost everywhere (from the social networks to the official registered databases) it becomes imperative to address the issue of automatically gaining insights from the collected data either by:

- interpreting the questionnaires responses (to determine different traits, communication styles, psychological personality, future trends in shopping or marketing, etc.),
- predicting or forecasting future events (diagnostics, illness's evolution or remission, stress levels related to contextual situations, suicidal intention, vulnerable categories of people in certain contexts, etc.)

Researchers identified this need and analyzed different approaches and methodologies both to facilitate the use of the collected data and to extract meaningful information among the big amount of data. The ability of handling significant amount of data transform machine learning algorithms into a good candidate for the above mentioned applications.

In [252] Dinga et al. apply the penalized logistic regression over a set of over 800 patients to predict the depression course. The study intends to find the best set of predictors out of clinical, psychological, or biological variables, based on the inventory of depressive symptomatology. The dataset is balanced in terms of the patients already being diagnosed or not with depression in the last two years. 81 features are analysed as the patients were monitored over two years. Subjects were grouped according to the illness' presence and to its trajectory: chronic, remission or gradual improvement. The results presented in [252] state an accuracy of 62% when predicting the illness' remission and an accuracy of 69% when it comes to predict the presence of a major depressive disorder.

Study [253] review the scientific literature for the applications of the personality data. They started from detecting the personality type from the psychology point of view (predict individuals' Big Five personality traits, using a wide range of written data, including informal social networks texts and reactions, musical preferences, spending records, language samples) and reached the unsupervised machine learning techniques to analyse other psychological aspects within the digital data. Secondly, the study directs its attention to the works which apply methodological questions in order to measure different social and demographic aspects such as predicting life outcomes, measuring tasks performances or even to self-reporting data.

Thirdly, it is investigated the use of machine learning algorithms applied on personality data to create recommender systems used in the marketing and retailing areas. Last but not least, the review paper analysed the principal issues(overfitting, underfitting, unbalanced data, etc.) that may occur during the use of the machine learning algorithms within the mentioned areas together with the possible solutions. A set of common measures used to interpret the results is also described. The review contains more than **130** research papers published between **2005-2020.**

Research paper [254] compose a comprehensive review of the principals machine learning methods applied to predict suicidal intentions among different categories of people. The authors present the different types of questionnaires designed to detect as early as possible the intention of suicide among the vulnerable categories of people. Furthermore, valuable information can be extracted by automatically processing the content published by the monitored subjects on different social networks by applying Natural Language Processing techniques (N-gram features, knowledge-based features, syntactic features, context features, term frequency-inverse document frequency matrices for messages,word embeddings, topic modelling, etc.). An increased attention must be paid to the affective characteristics, as they can easily discriminate between those with suicidal thoughts and healthy subjects. Deep learning methods (Multilayer perceptron, CNN, RNNs, LSTMs and attention based neural networks) are analysed across more than **100** research papers published between **2010-2020** dealing the suicidal ideation.

3.2.2 Experimental setup and results

In [14] we analysed six machine learning algorithms for the task of classification. More precisely, we used the data obtained from answering of a questionnaire for determining the communication style and try to link one person's stress level with the style of communication.

Our dataset contains 220 instances with more than 60 variables. As type of data, we worked with a discrete attribute of the nominal type (gender of the person), a numerical attribute (age), a discrete ordinal type (the level of stress) and 60 binary attributes (true or false). We chosen the questionnaire proposed by Marcus et al. in [255] which classifies a person's communication style into one out of the four communication styles: non-assertive, manipulator, aggressive and assertive by answering a set of 60 questions. Additionally, for the experiments described in [14] each person was asked to measure its stress level as low, medium or high, as the purpose of our study was to analyse the correlation between the stress level and the communication style.

We passed the data to six classification algorithms: Decision Tree Based Model, Support Vector Machine, Random Forest, Classification based on instances (k-NN), Naive Bayes and Logistic Regression. We evaluated the learning processing by applying the cross-validation technique. We consequently divided the dataset into k subsets and repeatedly trained the systems using $k-1$ subsets for learning and the last k subset for validating. For each learning iteration, one different subsets was left outside for validation only.

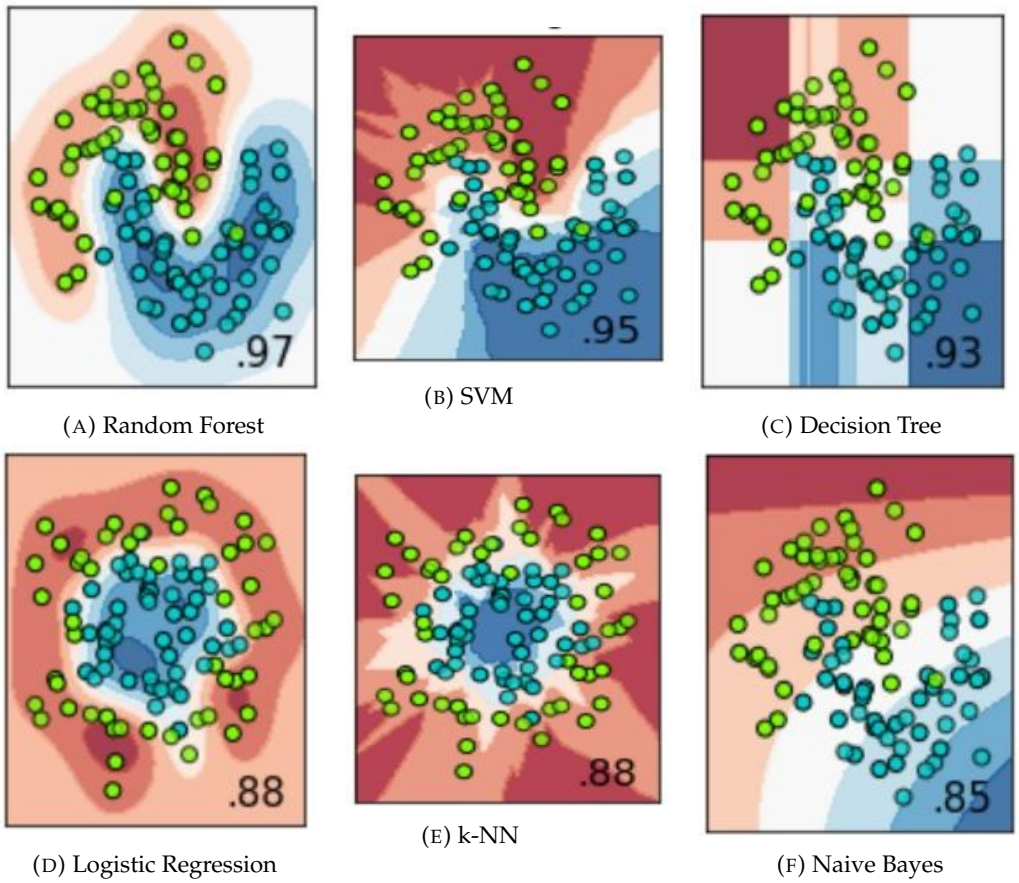
We evaluated these algorithms in terms of accuracy, precision, sensitivity, and specificity. In terms of classifier metrics, the results obtained by our six learners fall within the limits accepted in the literature. If we analyse the results obtained by the accuracy metric, we can conclude that the Random Forest classifier best performs, obtaining an accuracy of **97%**, while the Naive Bayes obtained the poorest results,

only **85%**. The results of the other three classification metrics are synthesized in Table 3.2.

Metrics Classifier Models	Naive Bayes	k-NN	Logistic Regression	Decision Tree	SVM	Random Forest
Accuracy	0.85	0.88	0.88	0.93	0.95	**0.97**
Precision	0.81.	0.79	0.81	0.78	**0.82**	0.81
Sensitivity	0.75	0.73	0.77	**0.78**	0.74	0.77
Specificity	0.68	**0.72**	0.62	0.63	0.62	0.63

TABLE 3.2: Communication styles: Results obtained by each of the six classifiers

Data dispersion for each classifier is illustrated in Figure 3.2. We can observe that the characteristics of each algorithm influence the data dispersion delimitations. Consequently, the limits are clearly illustrated for the Random Forest (Figure 3.2a) and the Decision Trees (Figure 3.2c), while Logistic Regression (Figure 3.2d) and k-NN (Figure 3.2e) do not provide clear boundaries.



(A) Random Forest (B) SVM (C) Decision Tree

(D) Logistic Regression (E) k-NN (F) Naive Bayes

FIGURE 3.2: Communication styles: Data dispersion after applying the classification models [14]

3.2.3 Conclusions and future work

In the experiments described in [14] we analysed the correlation between the stress level and the communication style over a group of 220 persons, by combining the

machine learning with the cognitive psychology. We applied six learning classification algorithms (Decision Tree Based Model, Support Vector Machine, Random Forest, Classification based on instances (k-NN), Naive Bayes and Logistic Regression) on a handcrafted dataset, created from the responses received at the communication styles questionnaire described in [255]. Once we will chose the best learner according to the studied context, we will apply the system to new unclassified instances, unseen during the learning process.

As future work, we intend to increase the size of the dataset in order to eliminate the biases in data. Furthermore, we will focus on analysing newer classification techniques with a better performance in terms of accuracy metrics.

Part II

Solutions for Romanian Speech Synthesis problems

Chapter 4

Theoretical insights into Text-to-Speech (TTS) systems

In this chapter we present the background knowledge and the state-of-the-art in the field of Text-to-Speech systems, using Machine Learning methods. We focus on Expressive TTS and Speech Synthesis for low resourced languages. The information collected in this chapter facilitates the research published in [9], [10].

4.1 TTS beginnings

Text-to-speech (TTS), also known as Speech Synthesis, represents a topic of interest for research due to the wide variety of applications in the industry. As TTS aims to create intelligible and natural speech by synthesising a given text, it requires knowledge from various disciplines: linguistics, acoustic, signal processing, machine learning. The first attempts of building artificial speech consists on mechanical TTS systems. In the 12th century, names as Albertus Magnus (1198–1280) and Roger Bacon (1214–1294) were linked to the legendaries "braven heads", an automaton meant to predict the future by answering to "yes or no" questions. Six centuries later, a Hungarian scientist Wolfgang von Kempelen (1734-1804) used bagpipes, springs and resonance boxes in order to produce simple words or short sentences [256]. In the 1930s the Nokia Bell Labs developed the vocoder [257], a device meant to easier transmit telephone conversations over long distances by reducing the bandwidth. The idea was to split the input signal (the human speech) into multiple bands and to keep only those necessary to recompose intelligible speech. Once the computers developed, the TTS technology includes articulatory synthesis, formant speech, concatenative speech, statistical parametric speech synthesis and neural speech synthesis. The following paragraphs will shortly describe the main types of TTS systems mentioned above [105].

4.2 TTS classification

4.2.1 Articulatory Synthesis

[258], [259] simulates the human articulator (lips, tongue, glottis or moving vocal tract) to produce the synthesised speech. Intuitively, this is the best TTS systems in terms of effectiveness, as it mimics the way the human body works. However, in practice, modelling these articulator behavior is difficult and challenging, as data for articulator simulator is hard to collect. The quality of speech obtained through articulatory synthesis is worse than the one obtained through formant or concatenative synthesis.

4.2.2 Formant Synthesis

[260], [261], [262] uses a set of rules - composed by the linguists, in order to best simulate the formant structure and spectral properties of the text - to build a source-filter model. An acoustic model and a synthesize module are used to produce the speech. The advantages of Formant Synthesis consist on highly intelligible output speech, moderate computational resources and the independence upon a pre-recorded human speech corpus, unlike the concatenative speech synthesis. The main disadvantages are the lack of naturalness and the artifacts present in the output speech. Moreover, the set rules are difficult to formulate.

4.2.3 Concatenative Speech

[263], [264], [265], [266], [267] consists on concatenating different pre-recorded sound pieces of the desired text in order to obtain the required output. The pre-existent database, usually recorded by a professional actor/speaker, contains samples of different sounds (vowels and consonants), syllables or even whole sentences. During the inference phase, the systems searches for units which best matching the input texts and concatenates them in order to obtain the desired output. The main benefit lays in the high intelligible synthesised speech, which retain the properties (timbre, accent, etc.) as close to the original recorded voice as possible. However, to cover all the possible combinations of sounds requires large amount of recorded database, difficult to obtain, especially for scarcely spoken languages - in contrast to English, Chinese or Hindi languages. Another disadvantage lays in the characteristics of the synthesised speech: the output voice is less natural and scarcely emotional, as it consist in concatenated speech units, which can lead to a noticeable transition from one sound to another.

4.2.4 Statistical Parametric Speech Synthesis - SPSS

meant to overcome the disadvantages of concatenative speech synthesis [268], [269], [270], [271], [272]. Instead of generating the waveform from existing recording speech units, SPSS predicts acoustics parameters and reconstruct the desired output using different algorithms [273], [274], [275]. A SPSS is composed by three components [105]:

- text processing module - implies text normalization, text segmentation at different granularity (phrase, word) or grapheme to phoneme conversion in order to extract linguistic features (phonemes, POS tags, etc) from the given input text.
- acoustic model - is trained with pairs of linguistic features and the corresponding acoustic characteristics (including fundamental frequency, spectrum, etc) to learn the behaviour of as many possible combinations of sounds as possible.
- vocoder - is used in both analysis and synthesis. First, the vocoder extracts (from the input audio files) the audio features needed to train the acoustic model. Second, during the synthesis, the vocoder uses the audio features predicted by the acoustic model and synthesizes the desired speech.

FIGURE 4.1: The 3 components of a TTS system

Among the advantages of the SPSS systems we can mention the following: the naturalness of the output voice, the low recorded databases compared with the concatenative synthesis systems and the adjustability, as the acoustic or the linguistics parameters can be modified in order to obtain the desired output. However, the SPSS systems have their lacunae. For instance, the synthesised sound is still metallic and sometimes hard to understand (in term of intelligibility) because of the artifacts like noises or buzziness.

With the appearance and the development of neural networks, as an attempt to overcome the SPSS drawbacks, ANN have been successfully included within the SPSS synthesis. Recent studies proved that replacing the HMM model with the Recurrent Neural Networks [276], [277], [278] or the Deep Neural Networks [278], [279], [280] within the SPSS systems lead to a better quality of the synthesised speech. Furthermore, studies as [281] proposed extracting the acoustic features directly from the phonetic content, not involving the linguistic features, as a first step in developing the end-to-end speech synthesis systems.

4.2.5 Neural Speech synthesis

[105] The next intuitive step was to develop the TTS systems based on neural networks to First WaveNet [282] and DeepVoice 1/2 [283], [284] replaced the main three components illustrated in Figure 4.1 with their correspondents based on the neural networks. With the rise of the end-to-end systems, the first attempts as Tacotron 1/2 [3], [4], [285], DeepVoice3 [286] or FastSpeech [287] came to simplify both the text analysis module (by directly processing the phonemes or characters sequences of the input text) and the acoustic module (by processing the waveform of the input wav file). ClariNet [288], FastSpeech2 [289] and EATS [290] came as a fully end-to-end alternative to predict the waveform directly from the input text.

A more comprehensive list of TTS systems is illustrated in Figure 4.2.

4.3 TTS systems for low resourced languages

As already described in the previous classification, the quality of a synthesised voice is highly dependent on the quantity of data used for training. Consequently, only widely spoken languages among which English, French, German, Russian, Hindi or Mandarin benefit of support for a wide range of applications from business to social good. In order to overcome the low resourced languages low coverage many research studies developed strategies to build TTS systems, among which we mention [105]:

4.3.1 Cross-lingual transfer

suppose pre-training the TTS system with the available amount from a rich resourced language and then, fine tune the output voice to the characteristics of the desired low resourced language [324], [325], [326]. However, may occur differences between the phonemes or the linguistics rules form one language to another. In [326] the authors

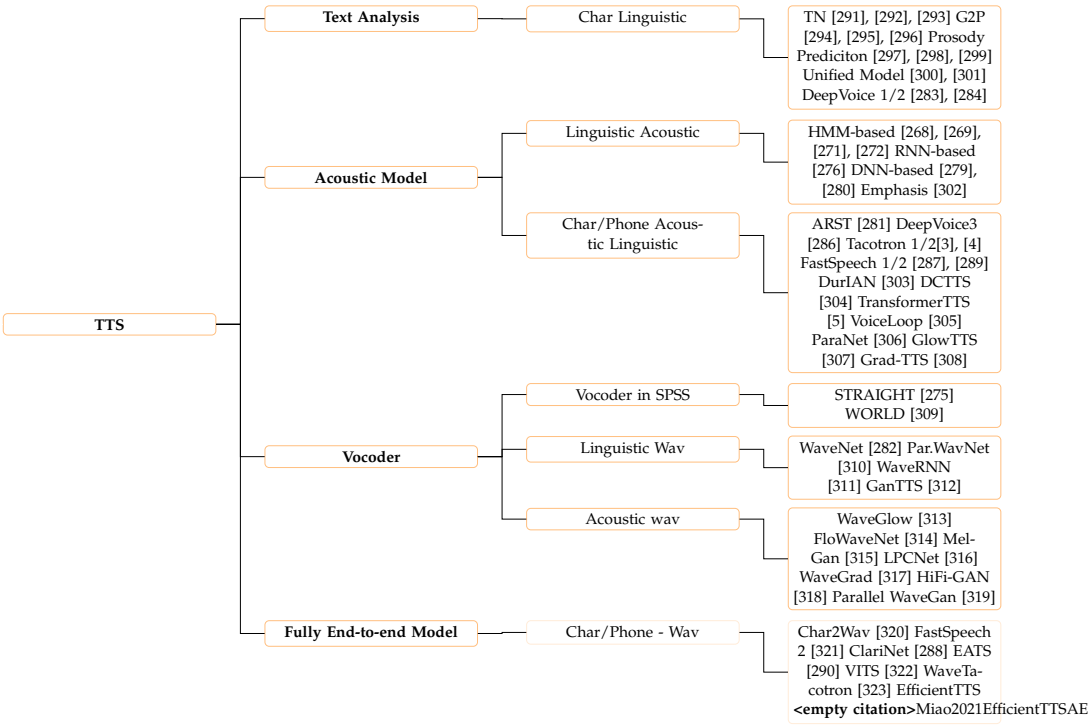


FIGURE 4.2: A TTS classification with examples [105]

propose a mapping between the linguistic symbols of the input and the target languages. LRSpeech [325] propose a Tranformer-based architecture to build TTS and voice recognition systems for scarcely represented languages, such as Lithuanian.

4.3.2 Cross-speaker transfer

When little data from a single speaker is available, the synthesised voice can be improved by transferring the knowledge from a better represented speaker. Techniques like voice conversion [327], [328], [329], speaker adaptation [330], [331] or voice cloning [332] may leverage the quality of the target speaker synthesised voice.

4.3.3 Self-supervised Learning

The TTS systems learn form the available input data, most commonly structured as pairs of the audio files and the corresponding written/linguistic information. For low resourced languages such labelled data is difficult to obtain. Inspired by the human's way of learning through everyday experiences, a self-supervised learning approach seem to improve the TTS Systems' language understanding and speech generation. Studies as [333], [334] come to enrich the TTS text encoder using pretrained BERT models [211] while the speech decoder can be trained together with a voice conversion task [335].

Figure 4.3 summarizes the above described approaches for TTS low-resourced languages.

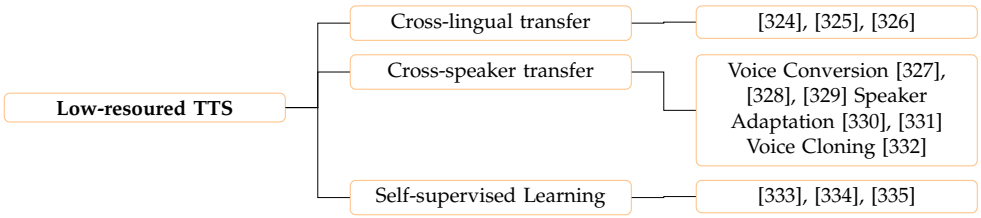


FIGURE 4.3: Low-resourced TTS approaches [105]

4.4 Expressive TTS

Beside the intelligibility and the naturalness, the expressivity of the synthesised speech is another characteristic which gained interest in the recent studies in the TTS field. The same text can have different interpretations in terms of sarcasm or emotions. Usually, the synthesised voice has a linear timbre and fails to convey the author's feelings. The expressivity of the generated voice can be influenced by the timbre, the prosody or the speech style. In order to increase the synthesised voice's expressivity, researchers enriched the input dataset with information regarding the speaker's style, timbre, accent, etc or with specifications about the text or about the speech prosody (rhythm, intonation, style, etc). Other approaches follow an unlabeled training by extracting as much information as possible from the variation of the input data which is then disentangled during training to obtain a more expressive voice. Figure 4.4 gathers the main approaches together with the significant studies.

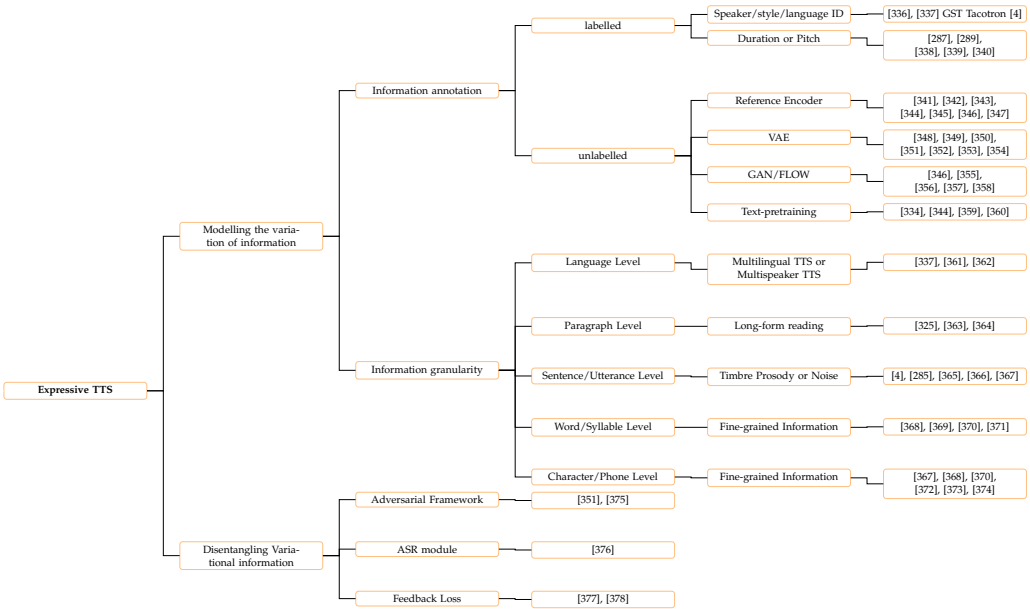


FIGURE 4.4: Expressive TTS approaches [105]

The written text is also analysed at different granularities in order to extract relevant information in terms of expressivity. For instance, at language level [337], [361], [362] we can use a language ID during the training of the multilingual TTS systems to differentiate between them. At paragraph level [325], [363], [364], the systems try to learn the links between the sentences and the words over the entire text in a long-term reading approach. At utterance level [4], [285], [365], [366], [367], an embedded vector is learnt form every sentence in order to retain information about the

timbre, the prosody or the style. The finest grained levels (word level and syllable level [368], [369], [370], [371] or even phoneme or character level [367], [368], [370], [372], [373], [374]) come to complete the learning by extracting specialized information (duration, pitch or prosody) hardly observed at a rougher level.

Beside modelling the variation information, recent studies focus on disentangling, controlling or transferring the information to improve the TTS expressivity. The datasets with mixed multiple styles, speakers or prosody information may cumber an accurate learning. Thus disentangling the variation information leverage the control and the transfer of the knowledge of interest. For instance, the adversarial framework is applied to disentangle either speaker information from noisy sounds [346] or one speaking style from another [351], [375]. In some cases, the TTS systems may disregard the style information embedded in the input data and provide a speech without the desired expressivity. Thus studies [377], [378] tries to enforce the TTS learning using a loss feedback to ensure a desired style or emotion achievement. Other approaches [376] use an automatic speech recognition module (ASR) to adjust the correlations between the training and the inference data.

4.5 Evaluation methods

As the synthesised speech is meant to be used by people within different applications, there is a need to evaluate the quality of the TTS systems. Common trend combine the subjective evaluation methods (listening tests) with the objective evaluation techniques. In the following paragraph we will sketch this two approaches together with the main advantages and disadvantages.

4.5.1 Subjective evaluation - Listening tests

The subjective evaluation of the synthesised speech consists in applying a listening test (a questionnaire about listeners' preferences for an apriori chosen set of audio samples) to a specially selected group of people. The listening tests can be coarsely classified into relatively preference based tasks and into ranking assigning tasks.

ABX preference

The first category contains pairs of reference - test audio files (stimuli) and the listeners have to express their preference for one of them. In a multi-system comparison TTS scenario, the stimuli pairs can be obtained by randomly combining two of the analysed systems, as in [9].

Mean Opinion Score (MOS)

The ranking assigning listening tests involve evaluating a set of stimuli (no reference stimuli is needed) on a certain scale, in terms of naturalness, expressivity and intelligibility, as applied in [10]. Usually the chosen scale range from 1 - *"Unsatisfactory"* to 5 - *"Excellent"*, as recommended in [379] by the IEEE Subcommittee on Subjective Methods and in [380] by ITU. Then the MOS (Mean opinion score) is computed as the arithmetic mean over single ratings performed by human subjects for a given

stimulus in a subjective quality evaluation test 4.1:

$$MOS = \frac{\sum_{n=1}^{N} R_n}{N} \tag{4.1}$$

where R are the individual ratings for a given stimulus by N listeners.

MuSHRA setup

The MUltiple Stimuli with Hidden Reference and Anchor (MuSHRA)[1] listening test setup compare the audio quality of several test conditions with the intermediate impairments to a high quality reference. Using a 1 to 100 scale, the listeners rate the TTS systems to be tested relatively to a reference stimuli. Although very similar with the MOS methodology, the MUSHRA tests present all the test conditions at the same time for each and every reference sample, as illustrated in Figure 4.5. In the research paper [10] we evaluated our TTS systems using the MUSHRA listening test.

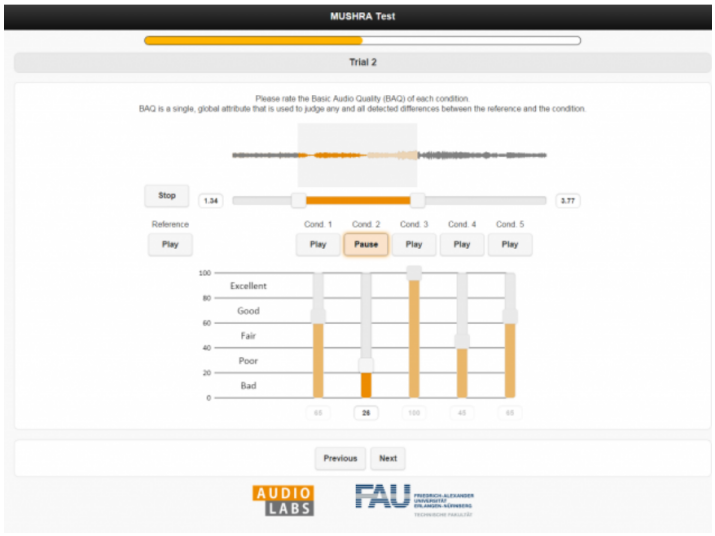


FIGURE 4.5: Caption from a MUSHRA test in AudioLab implementation

4.5.1.1 Advantages

The main advantage of the listening tests is the directly link with the human perception of the audio samples. While the objective evaluation methods avoid the subjectivity by scientifically computing and evaluating different distortion measures, they may not accurately reflect the way the human will perceive the synthesised samples. Thus better scored audio samples may sound poorer to the public and viceversa.

4.5.1.2 Disadvantages

As listening tests suppose evaluating audio samples based on the perception of the assessors, selecting professional listeners is an expensive process in terms of time and human resource. Moreover, the items of the questionnaire require a high level

[1] ITU-R Recommendation BS.1534-1

of attention and concentration as a listening test usually lasts 20 minutes. Thus collecting all the results is time consuming, unlike the objective evaluation methods. Moreover, biases may occur (some listeners may misrate different tasks of the test) which have to be dropped of from the final evaluation.

4.5.2 Objective evaluation - Distortion measures

The objective evaluation methods consists in ranking the TTS system's output naturalness based on different scores. The main flow of an objective evaluation consists in segmenting the audio signal in speech frames (of 10-30 milliseconds) followed by the analysis of a distortion measure computed between a target and a reference audio. To be noted that the distortion measure do not act completely like a distance, as some measures are not symmetric (in [381] authors suggest that symmetry should not pay an important role in distortion measures) or have negative values (log spectral distance measure). Different studies [382], [383], [384] survey the existent objective measures, however we will focus only on those used in our experiments.

Mel Cepstral Distortion (MCD)

In [9] we used Mel Cepstral Distortion (MCD) measure as an objective evaluation. In speech processing tasks we commonly analyze the waveform split into the multidimensional coefficients, seen as vectors, at uniform spaced intervals, called *frames*. The MCD is defined over such two cepstral coefficient vectors as:

$$MCD(c^{tgt}, c^{ref}) = \frac{1}{T}\sum_{t=0}^{T-1}\sqrt{\sum_{d=1}^{D}(c_d^{tgt}(t) - c_d^{ref}(t))^2} \tag{4.2}$$

where c^{tgt} and c^{ref} are the target and the reference cepstral vectors, respectively, T is the total number of frames, and D is the cepstral dimension, t is time or frame index.

The smaller the MCD between the synthesized and the natural mel cepstral sequences, the closer the synthetic speech is to reproducing the natural speech.

Mel Spectrogram Distortion (MSD)

However, in end-to-end systems, the cepstrum is not used to parametrise the waveform. The Mel spectrogram is used instead. The Mel Spectrogram Distortion (MSD) is similarly defined over two Mel spectrogram coefficients vectors as:

$$MSD(s^t, s^r) = \frac{10\sqrt{2}}{ln10}\frac{1}{T}\sum_{t=0}^{T=1}\sqrt{\sum_{d=1}^{D}(s_d^t(t) - s_d^r(t))^2} \tag{4.3}$$

where s^t and s^r are the target and the reference Mel spectrogram vectors, respectively; T is the total number of frames, and D is the number of Mel bins. The 0^{th} coefficient (the energy) is discarded. To align the synthesised and the natural sequences, a Dynamic Time Warping (DTW) algorithm was used. The smaller the MSD between synthesized and the natural mel spectogram sequences, the closer the synthetic speech is to reproducing the natural speech.

The experiments described in [10] are evaluated based on MSD score.

4.5.2.1 Advantages

By far the main advantage of the objective evaluation methods is the short time needed to obtain a measurable result. Thus the entire process of selecting the listeners and creating a listening test setup is replaced with computations based on a well known formula, already implemented in many programming languages[2].

4.5.2.2 Disadvantages

One disadvantage of this evaluation approach is that we always need the natural-voice sample corresponding to the evaluated synthesised utterance in order to accurate compute the distortion. Furthermore, the objective evaluation is not always related with the human perception on the audio sample. Thus, only the objective evaluation by itself may fail to accurately evaluate an output correlated with the synthetic voice's application purpose. In these cases, the objective results are interpreted in correlation with the ones obtained after the listening tests.

[2]MCD score Python implementation - https://github.com/MattShannon/mcd

Chapter 5

Enhancing the Romanian TTS Systems

In this chapter we apply machine learning in the field of speech synthesis. All the experiments and the corresponding results were elaborated in the original research papers [9], [10]. The following chapter is based on these publications.

5.1 Can synthesised speech data improve the speech expressivity?

In this subchapter we analysed if adding synthesised speech data to train the deep learning Text-to-Speech systems can improve the expressivity of the resulted synthesised voice. Detailed experiments have been published in [10].

5.1.1 Motivation

In latest research, the naturalness of text-to-speech systems has grown due to the use of the deep learning models. However, the expressivity of the synthesized voices (which is dependent on the existence of expressive corpora) remains a field of interest, especially for the low resourced languages. For the largely spread languages such as English or Mandarin extended corpora with expressive speech are frequently released, while for the other languages obtaining such a corpus is a challenge, as this is not part of the research community's interest.

In the absence of a large corpus enriched with expressivity, the researchers choose different methods to improve the TTS expressivity. Regardless the TTS paradigm chosen, to improve the naturalness of the synthesised voice the common approach is training a voice using data from multiple speakers and fine-tuning the obtained parameters toward a specific speaker, called *target speaker*. In this case, fewer data for the data speaker is necessarily, as the systems was able to learn an internal representation of the speech characteristics.

When we intend to improve the expressivity, the methods are chosen with respect to the TTS's paradigm. For the statistical parametric TTS, we need to adapt the duration and the fundamental frequency (F_0) models, as illustrated in [385]. The recent end-to-end systems, based on the sequence-to-sequence learning, do not contain a specific representation of the prosody, as the systems learn it using different latent or observed attributes. In [3] the Tacotron architecture is upgraded with an embedding layer responsible with learning the prosody contained in an audio reference. These learned prosodic features are passed to the network during the synthesis step. Studies of [4] and [285] introduce a new module (called Global Style

Token layer) to specifically learn different prosody characteristics present in the input data. Variational Autoencoders are applied in order to learn different emotions from speech, in an unsupervised manner [386] or to explain the characteristics not present in the input data: accent, recording conditions [387].

However, all the results mentioned above use manual adjustments or audio reference in the inference step. In [388] the features of the target speaker's style are learned through a hidden layer augmentation strategy, by adding new neurons to learn the desired style characteristics.

Romanian is part of the languages with limited data in the field of expressivity, both for voices or datasets. RSS [389] is a starter dataset containing for hours of high quality speech of a single speaker. SWARA [390] extends the RSS with new 16 speakers, leading to the one of the largest Romanian speech corpus with parallel data (includes 21 hours of speech). To the best of our knowledge, there are no other freely available Romanian datasets suitable for Speech synthesis. However, both RSS and SWARA contain only monotone speech, as the recorded texts have been chosen from the Romanian local newspapers. Having in mind all the previous mentioned studies, in our original research paper [10] we introduced a new expressive Romanian speech corpus MARA. We also analysed to what extend the synthesised speech data can improve the expressivity or the prosody transfer within a TTS system.

5.1.2 MARA dataset

MARA corpus started from an audiobook, read by a professional female actor, which was kindly granted to us by "Cartea Sonoră" [1]. It presents the story of a Romanian female called Mara, describing the struggles and the way people lived in Transylvania in the 19th century, during the Austrian-Hungarian occupation. The book entitled "Mara" was written by Ioan Slavici and published in 1906.

As the audiobook only contains the recorded audio files, one for each chapter, further processing steps were required, both at audio and at text levels.

At first, we manually segmented the audio files into smaller files, following the speaker's phrase break pauses. This step is needed due to the fact that the sequence-to-sequence architectures have great difficulties in learning from long utterances. Thus we obtained 8150 audio files with an average length of 5 seconds, corresponding to approximately 12 words. The entire dataset contains 11 hours of speech, sampled at 44kHz and 16bps.

As no text was provided, the next natural step was to manually divide the novel's written content into chunks following the corresponding audio files. The obtained text was annotated through the RACAI Relate Platform[2], resulting 8150 text files in CONLLU format[3] with high-level linguistic information such as text normalisation, phonetic transcription, syllabification, lexical stress assignment, lemma extraction, part-of-speech tagging, chunking and dependency parsing. To prepare the text for the TTS system input, a HMM based aligner was train to provide the phone-level boundaries.Then each phoneme is associated with the linguistic metadata using the HTS label format files [4].

All these processing steps are described in our original research paper [10]. As this work was part of the *SINTERO* research project [5] we have to mention that I, in

[1] https://www.youtube.com/c/CarteaSonoraCartiAudio/featured
[2] https://relate.racai.ro
[3] https://universaldependencies.org/format.html
[4] https://wiki.inf.ed.ac.uk/twiki/pub/CSTR/F0parametrisation/hts_lab_format.pdf
[5] `https://speech.utcluj.ro/sintero/`

particular, was responsible for processing half of the dataset.

The segmented corpus along with the annotations is available on `http://speech.utcluj.ro/marasc/` and utilizes a CC-by-NC-ND 4.0 licence.[6]

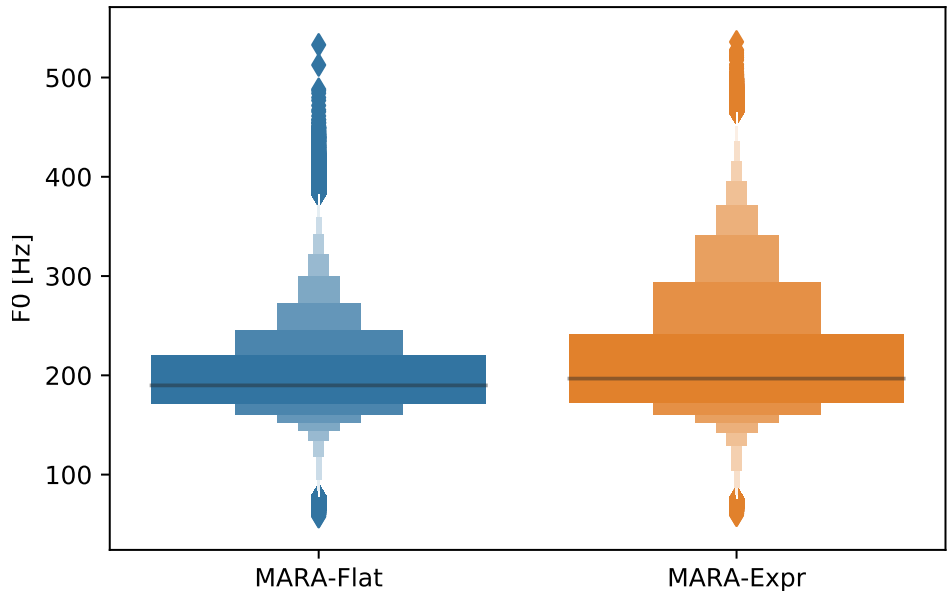
5.1.3 Experimental setup

Datasets

Due to the fact that the dataset contains audio files with different levels of expressivity, we divided the data in two parts:

- **MARA-Flat** - contains the audio files with a narrator's like intonation, translated as a F_0 mean value within 100kHz of the corpus value and a F_0 standard deviation smaller than 50kHz.
- **MARA-Expr** - contains the rest of the dataset, mostly with dialogue parts or characters voices played by the reading actor.

In figure 5.1 one can observe that the MARA-Expr subset has a much wider domain for the F_0 values. The dataset splitting is balanced, as each obtained dataset contains around 5 hours of data.



]

FIGURE 5.1: MARA Corpus: Letter-value plots of the F_0 values in the MARA-Flat and MARA-Expr subsets [10]

Obtaining synthesised expressive data

The following experiments were published in our original research [10], aiming to analyse the impact of the synthesized speech data to the overall TTS expressivity. We trained the TTS systems using only the narrative subset **MARA-Flat**. From the **MARA-Expr** audio samples, we extracted the phones duration and the F_0 contour which, combined with the spectral parameters generated by the TTS systems for the same utterances, generated the synthetic waveform. The obtained synthesised data

[6] `https://creativecommons.org/licenses/by-nc-nd/4.0/`

was used as an attempt to enrich the prosody when no expressivity is present in data. Two different TTS architectures were analysed: Hidden Markov Model systems and Deep Neural Networks. For the HMM based systems we selected the HTS implementation [391], while the deep learning based TTS systems were trained with the Merlin toolkit [47].

To overcome the artefacts of the HTS synthesised data, we applied a postfiltering procedure. Consequently, the output of the HTS system trained with the entire MARA-Flat dataset was paired with the corresponding natural audio files in a voice-conversion manner. This voice-conversion architecture is meant to direct the original input towards the target voice, thus correcting the HTS systems' artefacts. Figure 5.2 illustrates the overview of the above described processes.

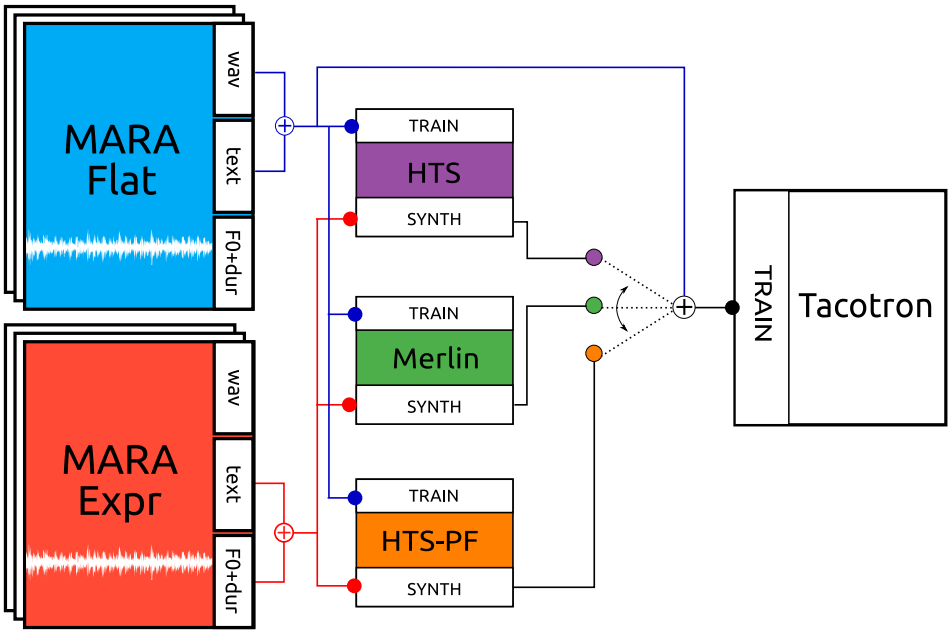


FIGURE 5.2: MARA Corpus: Block diagram of the end-to-end systems' training process using synthesised expressive speech data [10]

The end-to-end synthesis systems

The data obtained within the previous paragraph is fed to an end-to-end speech synthesis system. For the deep learning based architectures used in the experiments form [10], we chose the Mozilla open source implementation[7] of the Tacotron system [3].

To assure a clean start data (with no artefact obtained during synthesis), we firstly train Tacotron for 500 epochs using only the MARA-FLAT dataset. The obtained parameters and weights were applied to initialize all the other systems used during the [10] experiments. To get the baseline systems(ID: **TAC:FLAT**), we trained this system for another 500 more epochs. The topline is considered to be the systems trained with all data from the MARA corpus, both narrative (flat) and expressive (ID: **TAC:ALL**). The systems analysed during our expressivity experiments were trained as follow: the pretrained 500 epochs Tacotron Flat system was trained for another

[7] https://github.com/mozilla/TTS

TABLE 5.1: MARA Corpus: End-to-end synthesis systems' description

No.	System id	Expressive data
1	**Tac:FLAT**	None
2	**Tac:ALL**	Natural data
3	**Tac:Merlin**	Merlin synthesised data
4	**Tac:HTS**	HTS synthesised data
5	**Tac:HTS-PF**	HTS with post-filter synthesised data

500 more epochs using Merlin, HTS and HTS-postfiltered respectively synthesised data as input. Table 5.1 comprises the systems used.

5.1.4 Evaluation and results

The results obtained from our experiments on the expressivity task are described in [10]. We used both subjective and objective evaluation methods.

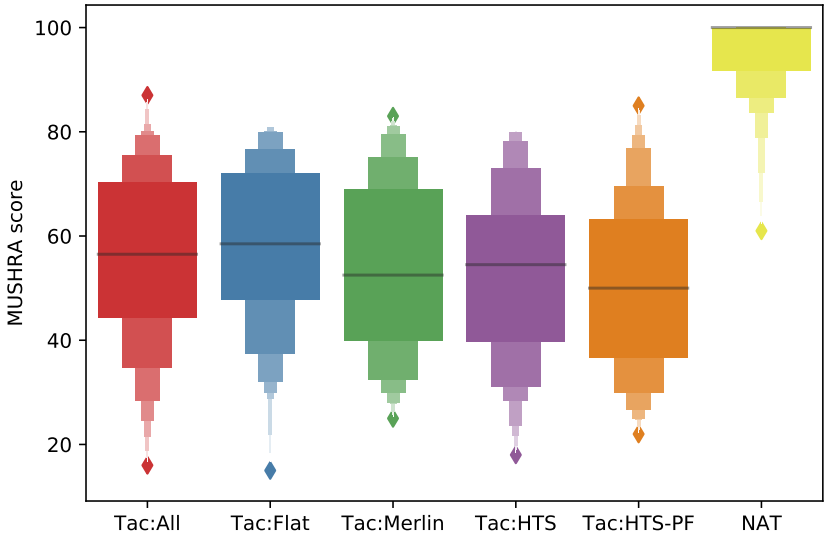
Subjective measure - Listening tests

In [10] we conducted a listening test based on the MUlti Stimulus test with Hidden Reference and Anchor (MuSHRA) methodology [8]. The test included two sections: the *naturalness* and the *expressivity*. In the naturalness section, the natural sample was presented to the listeners as reference. In the expressivity section, we did not want to influence the judgement of the listener, so that the natural sample was not clearly marked as reference, but was listed among the evaluated systems. In both sections, the lower anchor was set to a sample generated by the original HTS system. For each sample, 7 stimuli are presented to the listener side-by-side on the same screen, representing the 5 evaluated systems plus the natural and the original HTS samples. Each listener rated 10 screens and could playback the samples as many times as they wished. The average length of the utterances is 15 seconds. Audio samples from the listening test are available here: `http://speech.utcluj.ro/sped2021_mara/`.

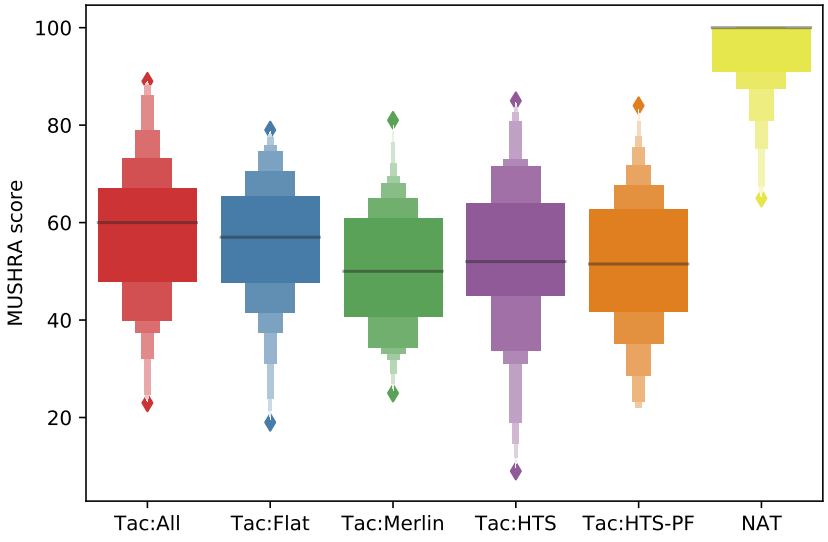
The results of the listening MuSHRA tests are illustrated in Figure 5.3 and Figure 5.4. As shown is Figure 5.3(a) the **Tac:Flat** system achieved best result in the naturalness part of the listening tests. One explanation may lay in the fact that the system learns better the speech spectral characteristics, as the F_0 variations range is not too wide. The worst result were obtained by the **Tac:HTS-PF** system because of the postfiltering system's metallic effect which has been propagated within the end-to-end process. Although only feedforward layers are used, the **Tac:Merlin** system obtained lower results than the **Tac:HTS** one, both at the naturalness and at the expressivity level. This can be explained by the phone-level alignment's accuracy which is below the threshold at which the feed-forward network can still compensate for its effects.

For the expressivity section, the **Tac:All** system obtained the best results. This was expected, as the system uses the audio data from the entire MARA Corpus. The poorest results were obtained by the **Tac:Merlin** system. The obtained values might be influenced by the fact that the listeners are not speech experts and they inevitable associate the expressivity with the speech naturalness.

[8] ITU-R Recommendation BS.1534-1



(a)



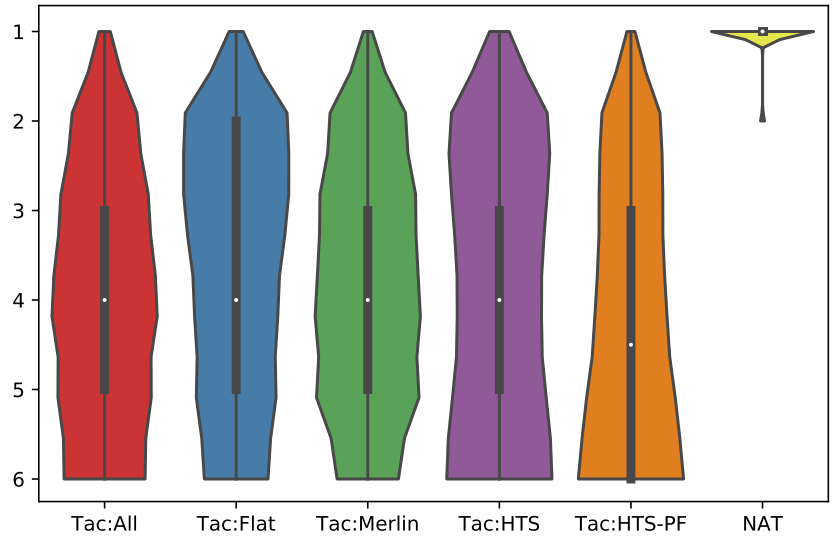
(b)

FIGURE 5.3: MARA Corpus: Letter-value plot of MuSHRA scores for the (a) **naturalness** and (b) **expressivity** section [10].

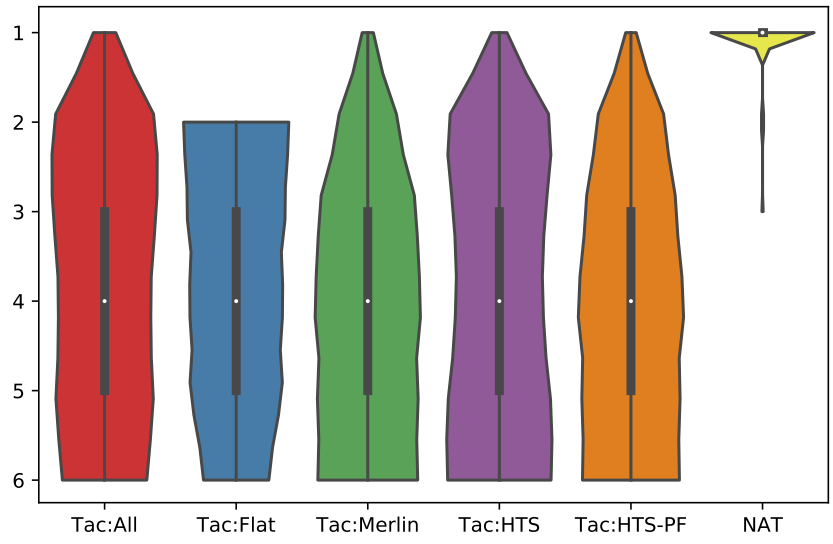
We also analysed the results of MuSHRA Listening test from the perspective of inter-systems ranking gave by the listeners. These results are illustrate din Figure 5.4 are correlated with the results obtained from the absolute MuSHRA scores described in Figure 5.3.

Objective evaluation - MSD measure

For the objective evaluation section, in [10] we chose 50 samples from the **MARA-Expr** subset, which were not present in the training set. The MSD scores were computed and illustrated in Figure 5.5. As we can notice, the **Tac:All** system obtained the highest mean. One explanation may lay in the fact that this set contains all the natural samples from the entire MARA dataset, which led to a higher prosodic variation. Furthermore, the lowest results obtained by the **Tac:Flat** system sustain the



(a)



(b)

FIGURE 5.4: MARA Corpus: Violin plot of MuSHRA rankings in the (a) **naturalness** and (b) **expressivity** sections[10].

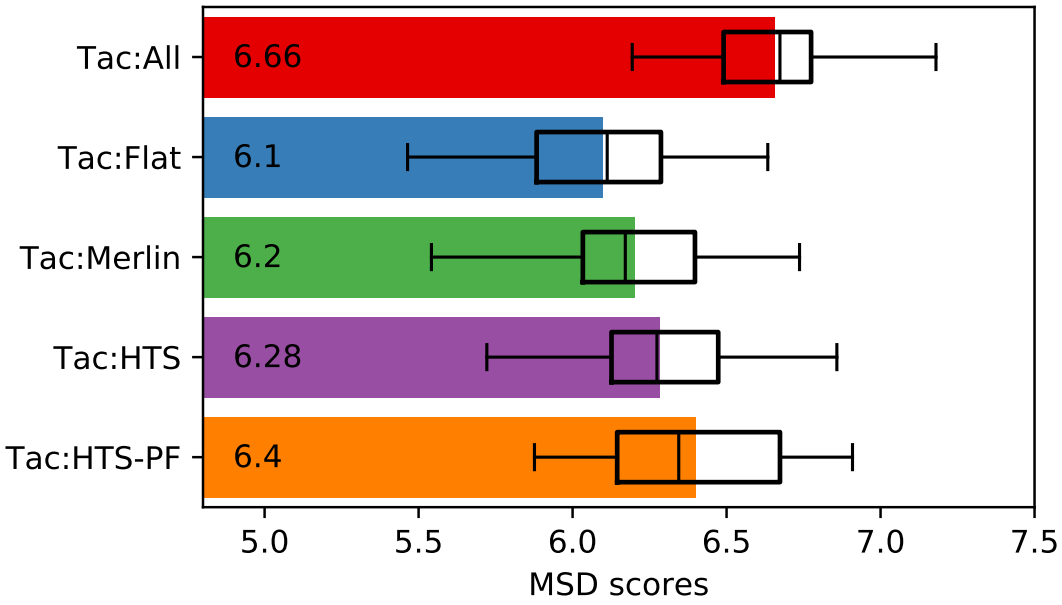


FIGURE 5.5: MARA Corpus: MSD scores across 50 testing samples. The horizontal bars represent the mean MSD values with boxplots overlapped [10].

contribution of the prosodic variation to the synthesizing process. Apart from these two opposed results, all the systems which used only parts from the MARA dataset obtained similar MSD scores. Thus we can conclude that the overall quality of the output is not influenced by adding synthesised data.

5.1.5 Interpretations, conclusions and future work

Having in mind all the previous interpretations, we can conclude that no statistically significant differences were found between the systems' objective ratings. Moreover, the listening tests showed that although substituting the natural samples with synthesised copies of them in the training data of an end-to-end TTS system the network is capable of averaging out the spectral artefacts of these samples. Thus, the naturalness and the expressivity of the output voice is only minimally affected. As future work, starting from the experiments described in [10], we intend to analyse other methods in order to obtain quality expressive synthesise data. Furthermore, we take into account the possibility of the inter-gender prosody transfer.

5.2 Using Postfiltering to enhance the quality of TTS systems with limited data

In this subchapter we address the problem of synthesised speech when limited data is available. The experiments are described in the research paper [9].

5.2.1 Motivation

Neural network based text-to speech systems achieve Mean Opinion Scores (MOS) close to the natural speech, as in the case of Tacotron2 [392]. The key of a qualitative synthesised voice is the quantity of the natural data used for training. Recent studies

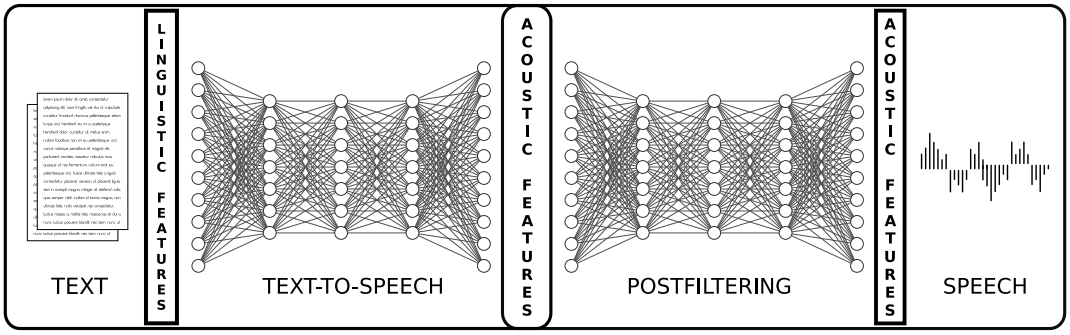


FIGURE 5.6: The postfiltering process.

developed TTS systems using generous training data, over 20 hours of recordings in the majority of cases [3], [286], [310], [393], [394], [395], [396], [397]. For scarcely spoken languages it is difficult to obtain large datasets to train qualitative TTS. The most common approaches to overcome this disadvantage consists in fine-tuning or in adapting the pre-trained model's parameters using data from the target speaker or language [398], [399], [400] or in appending speaker or language embeddings to the acoustic/linguistic features to help the model to learn discriminative features from the training dataset [395], [399], [401]

Having in mind the previous published research papers, we evaluated the postfiltering approach to improve the synthesised voice. Consequently, we trained a TTS system (using various amount of data from different Romanian speakers) and we pass the resulted voice to a postfiltering network to overcome the limited data. We obtained 20 systems which were objectively analysed. Furthermore, we selected 7 systems for a listening test, subjectively analysed by native Romanian speakers.

5.2.2 Experimental setup

As the scope of this study was to determine a postfiltering method to improve the quality of the synthesised voice even when limited training data is available, we structured the experiments into two steps:

1. train a DNN TTS with different amount of data
2. apply a postfiltering neural network to enhance the trained output

Figure 5.6 illustrates the flow's overview.

Many recent studies choose to train end-to-end TTS systems, obtaining high quality output. As already discussed, this approach requires generous amount of input data, which can be challenging to obtain, especially for the languages such as Romanian. Thus for the experiment descried in [9] we decided to train a statistical parametric network, using the setup described in [47]. Our approach consists in multiple steps. First, we prepossessed the input text to obtain the HTS format labels files, as introduced in [391]. As all experiments were applied to the Romanian language, thus the obtained linguistic features were derived using a Romanian TTS front-end described in [389][9]. These lexical features were paired with the corresponding audio files and fed to a DNN TTS network. Secondly, the synthetic and the natural feature vectors were aligned using the Dynamic Time wrapping [402]. The resulted aligned pairs were fine tuned using the postfiltering network.

[9]available online at `www.romanianTTS.com`

5.2.3 Datasets

As described in [9] all the experiments were run over a subset from the SWARA [8] Romanian speech corpus [10]. 8 female speakers (*BAS, CAU, EME, DCS, DDM, HTM, PMM* and *SAM*) were selected out of all the 17 speakers contained in corpus. Two additional female voices (*MAR* and *BEA*) were recorded for testing purposes, in similar recording conditions and using the same prompts as for the SWARA corpus. We have to mention none of the recorded speakers were professional speakers nor actors. The audio data was manually segmented at utterance level and sampled at 48Hz and 16bps.

As this work was part of the *SINTERO* research project [11], we have to mention that I, in particular, was responsible for recording my own voice for testing (*MAR* voice) and for processing the resulting audio files. Moreover I also dealt with the systems which were trained based on the *MAR* voice and analysed their results obtained within the experiments from [9].

5.2.4 TTS systems

The TTS systems are based on the Blizzard Challenge 2017 Merlin [47] setup. We extracted the acoustic features with WORLD vocoder resulting 59 plus the 0^{th} Mel generalised coefficients, 5 band aperiodicity coefficients and a fundamental frequency (F_0), enriched with delta and delta-delta values. The neural network is composed of 6 fully connected layers with 1024 nodes and *tanh* activation function. Other neural configurations (4, 5 or 6 layers with 256, 1024 neurons per layer or with bottleneck 1024-512-256-512-1024) with different activation functions (tanh or ReLu) were tested within the feed-forward approach.

The amount of input training data varies from 50 utterances (approx. 5 minutes), 100 utterances (approx. 10 minutes) up to 500 utterances (approx. 50 minutes) and consists in pairs of linguistic and acoustic features. These training systems are referred to as **M***. The training utterances are arbitrary chosen from the Romanian newspapers, thus they are not phonetically balanced nor filtered. In an attempt to enrich the dataset, we trained two more systems with doubled input data, by simply adding twice the initial data (ID: **Db**). This approach was also analysed for the postfiltering setup. We entitled the resulted TTS as **M*Db_P*Db**.

The postfiltering neural networks were based on voice conversion technique (targeting an initial voice to sound like a desired voice) and speaker adaptation (an eigen voice - trained over mixed data from multiple speakers - is targeted to the acoustic features of a certain speaker).

5.2.5 Evaluation and results

The results obtained from our expressivity experiments [9] were evaluated both with the subjective and the objective methods.

Subjective measure - Listening tests

Although the objective evaluation methods are preferred due to unbiasness, they do not truly correlate to the perceptual evaluation of the synthesised speech. Thus, we conducted listening tests to analyse the TTS output. We selected 7 systems for

[10] The corpus is available online at speech.utcluj.ro/swarasc/

[11] `https://speech.utcluj.ro/sintero/`

TABLE 5.2: Synthesis systems' description [9]

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

both MAR and BEA voices recorded for testing purposes. We created one separate listening test for each voice which were evaluated by 20 Romanian native speakers.

The listening tests contain 4 sections:

1. Naturalness - evaluated with a 5 MOS scale (Mean Opinion Score) from 1 = Unnatural to 5 = Natural.

2. Speaker-similarity - evaluated with a 5 MOS scale (Mean Opinion Score) from 1 = Not similar to 5 = Very similar.

3. Intelligibility - evaluated with WER (Word Error Rate).

4. ABX naturalness - listener has to decide which audio output sounds more natural from random pairs of systems.

Figure 5.7 describe the results obtained during the listening tests. In (a) and (b) the 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.

The System G is considered to be the baseline, as it uses most of data for training. However, we were more interest in the systems using as little data as possible (5 or 10 minutes) as the scope of this study [9] was to analyse the impact of postfiltering methods to the synthesize speech. From the listening test, we can conclude that the postfiltering increased the systems' speaker similarity and their naturalness, as observed from the results obtained by system C, for the both speakers (*BEA* and *MAR*). When the input data was doubled, the naturalness of the output speech increased, although no visible improvements were obtained for the speaker similarity criteria. Moreover, the intelligibility decreased for all the systems after the postfiltering, with little gain when the data was doubled. The systems trained with data from multiple speakers obtained better results at the speaker similarity sections, compared with the speaker dependent systems.

Objective measure - MCD

All the 7 selected systems were objectively evaluated using the average Mel Cepstral Distortion. For both voices selected for testing (*BEA* and *MAR*), we synthesised 50 utterances not present in the training set and we compute the MCD score for all the 7 selected systems. Results are illustrated in Figure 5.7. As we anticipated, the systems **M500** and **M50** obtained the best and the worst results respectively. If we analyse the results obtained by the postfiltering systems, we can observe that the MCD scores decreased with 5% to even 7.5% for the ***M500_PF500*** system. Artificially doubling the data increased the systems' quality. Doubling the data for both training and postfiltering led to a decrease of 10% for the MCD values. However, doubling the data only for the postfiltering step slightly changed the results. If data for multiple speakers is available, speaker adaptation technique proved to be a solution: systems ***SPKA100_E100, SPKA100_E500, SPKA500_E500*** obtained one of the lowest MCD scores. In spite of these results, when we used multi speaker data only for the postfiltering step, the system obtains results comparable only with the speaker dependent filter.

5.2.6 Conclusions and future work

The results obtained during the experiments described in [13] proved that postfiltering and artificially doubling the data improved the quality of synthesised speech. Moreover, the two techniques can be jointly used when insufficient training data is available. The postfiltering results can be explained by the fact that as it only learns a mapping of vectors sampled from similar feature spaces, it actually learns where the TTS system failed with respect to the natural samples, not to the lexical input. Artificially doubling the data has effect at the DNN setup level. The training uses batches of data, which are not sequentially selected. Doubling the data leads to more samples to learn from, which can improve the output.

When we listen to the synthesised samples we observed that the postfilter corrected many of the voiced/unvoiced decision errors of the TTS system. Moreover the buzzines was also reduced. Despite all this, the postfilter systems' output sounds more metallic which lead to an undesired decrease of the intelligibility. The overcome of this side effect is one of the focus of our next studies.

As future work, beside training more different TTS network architectures, we intend to study other vocoders or to add more features to the postfilter network's input, like lexical or speaker embeddings. For the multispeaker scenario, we intend to analyse the weights tuning for the target speaker. Moreover, we did not address the male voices, which can offer a different context and behaviour.

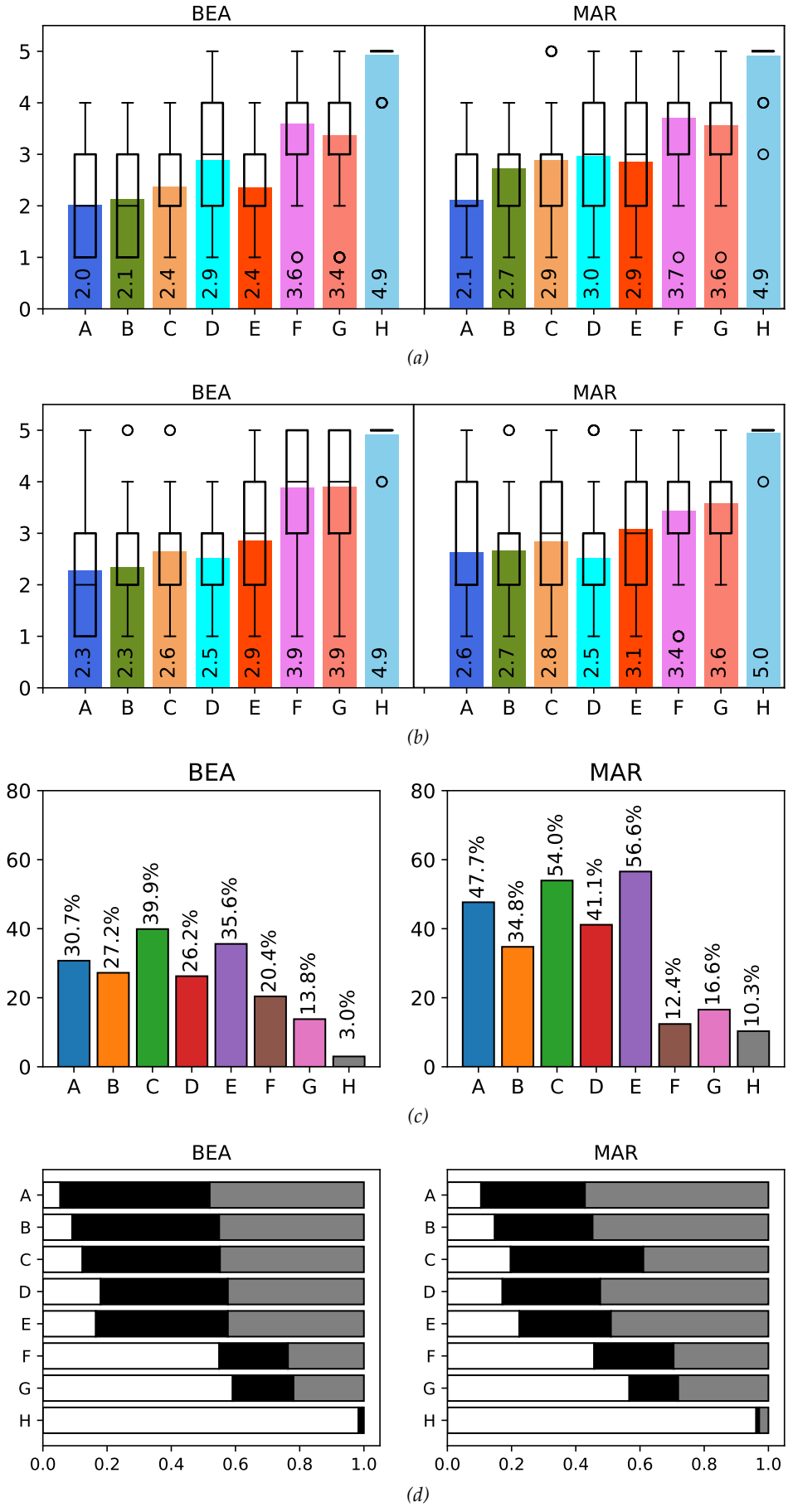
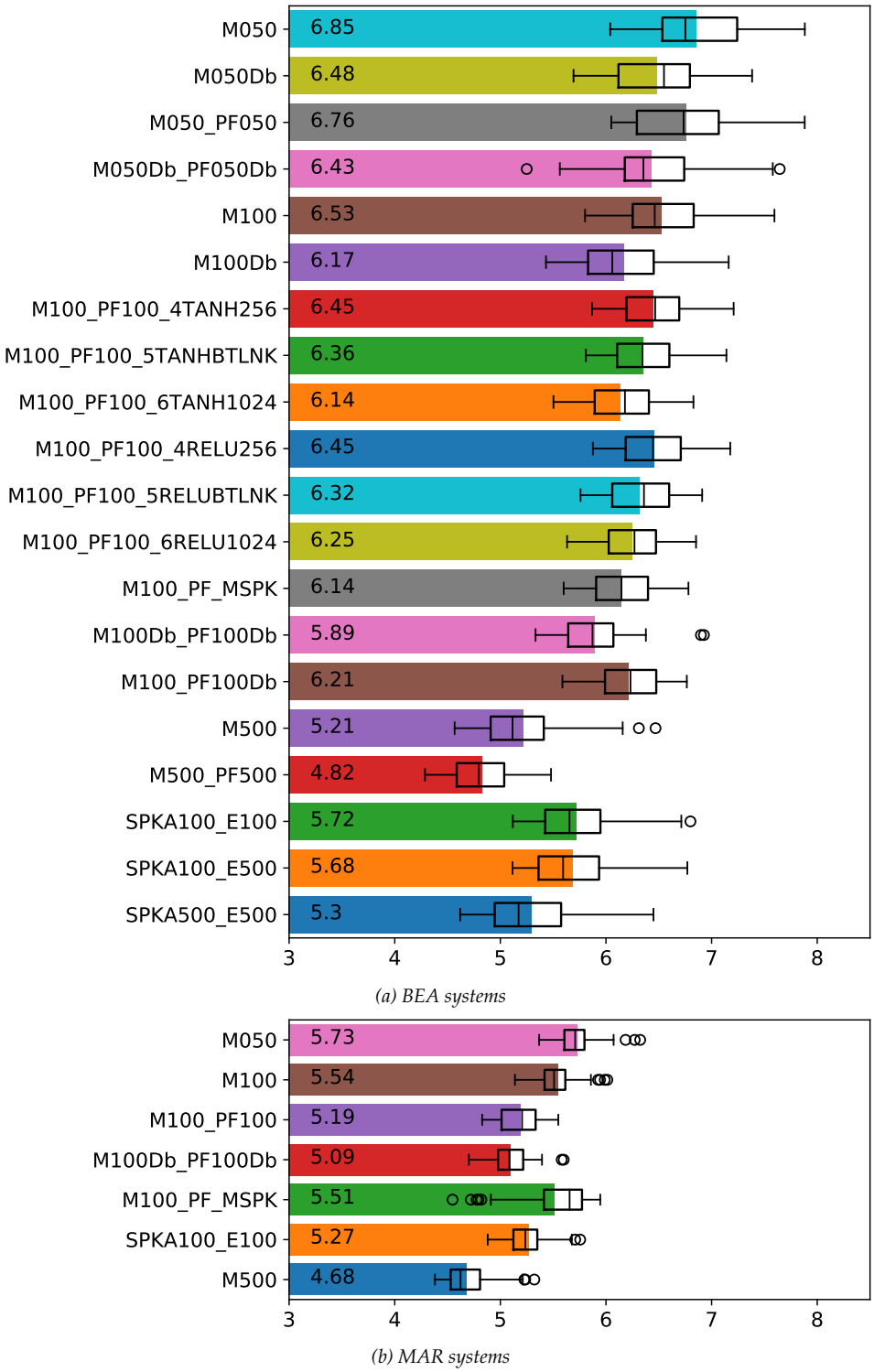


FIGURE 5.7: Listening test results for speakers **BEA** and **MAR**: (a) Naturalness MOS scores, (b) Speaker similarity MOS scores, (c) Intelligibility WER, and (d) ABX preference. [9]



(a) BEA systems

(b) MAR systems

FIGURE 5.8: Average Mel Cepstral Distortion for the (a) **BEA** and (b) **MAR** systems. Horizontal bars represent the mean MCD values, and are overlapped with boxplots. [9]

Chapter 6

Conclusions

This book gathers together machine learning based solution for problems from text processing and speech synthesis.

For the Natural Language Processing (NLP) part, we focused on two directions. On the one hand, we applied several machine learning models to automatize the process of extracting relevant information from the medical records. We analysed both the supervised (text classification [14]) and the unsupervised (document clustering, topic modelling [2]) learning techniques. The experiments were run for written English. On the other hand, we addressed aspects from the text annotation field by applying neural networks based solutions in tasks like automatic diacritics restoration [12], automatic lemmatization [11] or POS tagging [13]. These latest experiments were run for texts written in Romanian.

In our experiments from [11], [12], [13] we trained our deep learning systems in a supervised manner, with labelled pairs of words and their corresponding annotations (lemma, diacritized form or POS tag, depending on the researched task). The input text was encoded and passed to a encoder-decoder architecture. As **future work**, we will focus on analysing different types of neural networks, such as bidirectional LSTMs, GRU, or only attention based architectures (transformers), which are already frequently applied in other text processing tasks. As already stated in Section 2.2.6, even if transformer based networks successfully solves NLP tasks in widely used languages (English, Mandarin), we have to analyse the impact on the training process of the limited amount of data if we want to apply these systems to the Romanian language. Furthermore, inspired by other studies in the field of text processing, we intend to explore more types of encoding for the input text, such as word embeddings or conceptualized embeddings (BERT). Nevertheless, enriching the input text with more context information may lead to better results in predicting the desired annotation. However, we have to mention that the scope of studies [11], [12], [13] was to automatically process the input text using the minimum context knowledge, due to the lack of large annotated corpora in Romanian.

Beside automatically annotating texts written in Romanian, we applied the machine learning NLP algorithms to ease the work of the medical physicians. We focused our work on two main directions: determine patients' medical diagnostic through topic modelling techniques [2] and interpreting the psychological questionnaires' results [14]. At first, we created a dataset consisting of the personal records of a family doctor, gathered during the consultations. The dataset contains 102 instances, consisting in written text, more precisely the clinical observations and the prescribed treatment, both in English, and numerical data for the patient's response to treatment encrypted from 1 = non-responsive to 5 = very responsive. Another direction was working with personality data, by combining cognitive psychology and machine learning. We analysed over 200 instances with more than 60 variables, as

each participant at the study was asked to answer to a 60 questions communication style questionnaire together with measuring the personal stress level (low, medium, high). For the obtained dataset and the current task, we trained and tested 6 machine learning classifiers.

As future work for the NLP tasks using medical data, described in [2] and [14] we intend to analyse the impact and the efficacy of other types of machine learning and deep learning algorithms. Secondly, it would be of interest to study medical data written in the Romanian language, having in mind not only the language particularities (diacritics, spelling, etc), but also the challenge of gathering the input data, as Romanian is a scarcely represented language. Moreover, we can apply the systems developed in [11], [12], [13] to automate the text's annotation and to correct its undiacritised written form.

For the Text to Speech Synthesis part, the aim was on increasing the quality and the expressivity of the synthesised voice. We analysed different neural networks architectures using Romanian texts as input. Our approaches are novel in relation to the Romanian Speech Synthesis field and have been published in research articles within conferences proceedings [9] [10].

In our experiments from [9] we researched the potential of postfiltering techniques to enhances the quality of TTS systems with low resourced data input available. We split our approach in two parts: first we trained the TTS systems with different amounts of data, then we applied a postfiltering neural network to enhance the trained output. The amount of the input training data varies from 50 utterances (approx. 5 minutes), 100 utterances (approx. 10 minutes) up to 500 utterances (approx. 50 minutes) and consists in pairs of linguistic and acoustic features. Moreover, we trained two systems with doubled input data, by simply adding twice the initial data, in order to analyse if the quality of the output is influenced rather by the physical amount of data than by the data content. The postfiltering neural networks were based on voice conversion technique (targeting an initial voice to sound like a desired voice) and speaker adaptation (an eigen voice - trained over mixed data from multiple speakers - is targeted to the acoustic features of a certain speaker). These latest experiments were run for texts and audio samples in Romanian and the results are detailed in [9]. As this work was part of the *SINTERO* research project[1] we have to mention that I, in particular, was responsible for recording my own voice for testing (*MAR* voice) and for processing the resulting audio files. Moreover I also dealt with the systems which were trained based on the MAR voice and analysed their results obtained within the experiments from [9].

Research paper [10] analyses different ways to enrich the expressivity of the TTS systems in a low resourced emotional/expressive dataset context. With this purpose in mind, we first created an expressive dataset in Romanian, consisting in an audiobook (*Mara* - written by Ioan Slavici) which was manually segmented in smaller files, following the speaker's phrase break pauses. The written text was annotated through the RACAI Relate Platform with high-level linguistic information as described in Section 5.1.2. Starting with MARA dataset we analysed the impact of synthesized speech data to the overall TTS expressivity. We trained 5 different TTS systems with various amount of expressive data as input: None, synthesised expressive data or all natural expressive data. The impact of expressive data and all the result are described in our research work [10].

As future work for the speech synthesis part, beside training more different TTS network architectures (attention based, transformers), it is of interest to examine

[1] `https://speech.utcluj.ro/sintero/`

other vocoders or to add more features to the postfilter network (lexical or speaker embeddings) in an attempt to enrich the synthesised speech quality. For the expressive TTS systems experiments, we intend to analyse other methods in order to obtain quality expressive synthesise data. Furthermore, we take into account the possibility of the inter-gender prosody transfer.

Research work as a whole Having in mind all the experiments run for input data written in Romanian language, we plan to encapsulate all the resources and the systems described in the present volume in a tool, aiming to help others researchers within their work. More specifically, we intend to create a Romanian text-to-speech system, enriched with expressivity, which will be feed with the input written text preprocessed by the tools analysed in [11], [12], [13] and consisting on the TTS technologies introduced in [9], [10].

List of Publications

All rankings are listed according to the UEFISCDI journal classification for financing of research results[2] and CORE classification of conferences in Computer Science[3]. For each article, we considered the classification valid in the year of publication

Publications in international journals and conferences

1. [11] **Maria Nuţu**. *Deep Learning Approach for Automatic Romanian Lemmatization*. In *2021 25th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems (KES2021)*, Procedia Computer Science, Elsevier Publisher, vol. 192, pp. 49-58.

   Rank B, 4 points.

2. [14] Adriana Mihaela Coroiu, Alina Delia Călin, and **Maria Nuţu**. Communication Style - An Analysis from the Perspective of Automated Learning. In *Artificial Neural Networks and Machine Learning (ICANN)*, Cham Springer International Publishing, pp. 589–597, 2018 ISBN: 978-3-030-01418-6

   Rank B, 4 points.

3. [2] Adriana Mihaela Coroiu, Alina Delia Călin, and **Maria Nuţu**. Topic Modeling in Medical Data Analysis. Case Study Based on Medical Records Analysis. In *2019 International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, pp. 1–5, 2019.

   Rank B, 4 points.

4. [9] Beáta Lőrincz, **Maria Nuţu**, Adriana Stan and Mircea Giurgiu. An Evaluation of Postfiltering for Deep Learning Based Speech Synthesis with Limited Data. In:2020 *IEEE 10th International Conference on Intelligent Systems (IS)*, pp. 437–442, 202,

   DOI:10.1109/IS48319.2020.9199932.

   Rank C, 1 point.

5. [12] **Maria Nuţu**, Beáta Lőrincz and Adriana Stan Deep Learning for Automatic Diacritics Restoration in Romanian. In *2019 IEEE 15th International Conference on Intelligent Computer Communication and Processing (ICCP)*,IEEE Computer Society, pp. 235–240, 2019.

   Rank C, 2 points.

6. [13] Beáta Lőrincz, **Maria Nuţu**, and Adriana Stan "Romanian Part of Speech Tagging using LSTM Networks". In *2019 IEEE 15th International Confer-ence*

[2] `https://uefiscdi.gov.ro`

[3] Computing Research and Education Association of Australasia, `https://portal.core.edu.au/conf-ranks`

on Intelligent Computer Communication and Processing (ICCP), IEEE Computer Society, pp. 223–228, 2019.

Rank C, 2 points.

7. [10] Adriana Stan, Beáta Lőrincz, **Maria Nuţu** and Mircea Giurgiu, "The MARA corpus: Expressivity in end-to-end TTS systems using synthesised speech data," 2021 International Conference on Speech Technology and Human-Computer Dialogue (SpeD), Bucharest, Romania, 2021, pp. 85-90,

   Rank D, 0.5 points.

Publications Score: 17.5 **points**

Citations of the published research paper (source: Google Scholar)

- [11] **Maria Nuţu** Deep Learning Approach for Automatic Romanian Lemmatization. In *2021 25th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems (KES2021)*, Procedia Computer Science, Elsevier Publisher, vol. 192, pp. 49-58.

  Citations

  1. [403] Pratama, Angga, Raksaka Indra Alhaqq, and Yova Ruldeviyani. "Sentiment Analysis Of The Covid-19 Booster Vaccination Program as a Requirement for Homecoming During Eid Fitr In Indonesia." Journal Of Theoretical And Applied Information Technology , vol.101, No.1, ISSN: 1817-3195 (2023).
  2. [404] Mouyassir, Kawtar, Abderrahmane Fathi, and Noureddine Assad. "Elevating Aspect-Based Sentiment Analysis in the Moroccan Cosmetics Industry with Transformer-based Models." International Journal of Advanced Computer Science & Applications 15.6 (2024).
  3. [405] Yoon, Liu Jun, et al. "A Comparative Study of Lemmatization Approaches for Rojak Language." The International Conference on Data Science and Emerging Technologies. Springer Nature Singapore, 2023.
  4. [406] Kawtar, Mouyassir, et al. "Hierarchical Spatiotemporal Aspect-Based Sentiment Analysis for Chain Restaurants using Machine Learning." International Journal of Advanced Computer Science & Applications 15.3 (2024).

- [10] Adriana Stan, Beáta Lőrincz, **Maria Nuţu** and Mircea Giurgiu, "The MARA corpus: Expressivity in end-to-end TTS systems using synthesised speech data," 2021 International Conference on Speech Technology and Human-Computer Dialogue (SpeD), Bucharest, Romania, 2021, pp. 85-90,

  Citations

  1. [407] Ungureanu, D., Badeanu, M., Marica, G. C., Dascalu, M., and Tufis, D. I. (2021, October). Establishing a Baseline of Romanian Speech-to-Text Models. In 2021 International Conference on Speech Technology and Human-Computer Dialogue (SpeD) (pp. 132-138). IEEE.
  2. [408] Beáta Lőrincz, Elena Irimia, Adriana Stan, and Verginica Barbu Mititelu. "RoLEX: The development of an extended Romanian lexical dataset and its evaluation at predicting concurrent lexical information." Natural Language Engineering (2022): 1-26.

3. [409] Gasan, Carol-Luca, and Păis, Vasile. "Investigation of Romanian speech recognition improvement by incorporating Italian speech data." Linguistic resources and tools for natural language processing (2023): 235.
4. [410] Stan, Adriana, and Johannah O'Mahony. "An analysis on the effects of speaker embedding choice in non auto-regressive TTS." arXiv preprint arXiv:2307.09898 (2023).

- [9] Beáta Lőrincz, **Maria Nuţu**, Adriana Stan and Mircea Giurgiu. An Evaluation of Postfiltering for Deep Learning Based Speech Synthesis with Limited Data. In: 2020 *IEEE 10th International Conference on Intelligent Systems (IS)*, pp. 437–442, 202,

  DOI:10.1109/IS48319.2020.9199932

  Citations:

  1. [411] Eren, Eray, and Cenk Demiroglu. *Deep learning-based speaker-adaptive postfiltering with limited adaptation data for embedded text-to-speech synthesis systems*. Computer Speech & Language (2023): 101520.
  2. [412] Beáta Lőrincz. Contributions to neural speech synthesis using limited data enhanced with lexical features. In Proc. 2021 ISCA Symposium on Security and Privacy in Speech Communication (pp. 83-85).
  3. [413] Anas Fahad Khan et. al. *When Linguistics Meets Web Technologies. Recent advances in Modelling Linguistic Linked Open Data.* In The journal Semantic Web – Interoperability, Usability, Applicability, publisher IOS Press, ISSN: 1570-0844,

- [12] **Maria Nuţu**, Beáta Lőrincz and Adriana Stan. Deep Learning for Automatic Diacritics Restoration in Romanian. In *2019 IEEE 15th International Conference on Intelligent Computer Communication and Processing (ICCP)*,IEEE Computer Society, pp. 235–240, 2019.

  Citations:

  1. [414] Stankevičius, L., Lukoševičius, M., Kapočiūtė-Dzikienė, J., Briedienė, M., & Krilavičius, T. (2022). Correcting diacritics and typos with a ByT5 transformer model. Applied Sciences, 12(5), 2636.
  2. [415] Pakalniškis, L. (2022). Giliuoju mokymusi grįstas diakritinių ženklų atstatymas lietuvių kalbai (Doctoral dissertation, Kauno technologijos universitetas).
  3. [416] Stan, A., & Lőrincz, B. (2021). Generating the Voice of the Interactive Virtual Assistant. In Virtual Assistant. IntechOpen.
  4. [417] Hifny, Y. (2021, June). Recent Advances in Arabic Syntactic Diacritics Restoration. In ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 7768-7772). IEEE.
  5. [418] Náplava, J., Straka, M., & Straková, J. (2021). Diacritics Restoration using BERT with Analysis on Czech language.
  6. [419] Esmail, S., Bar, K., & Dershowitz, N. (2021). How Much Does Lookahead Matter for Disambiguation? Partial Arabic Diacritization Case Study. (Master thesis)

     Tel Aviv University, Blavatnik School of Computer Science)

7. [420] Scott, K. M., Ashby, S., & Cibin, R. (2020, September). Implementing text-to-speech tools for community radio in remote regions of Romania. In Adjunct Proceedings of the 2020 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2020 ACM International Symposium on Wearable Computers (pp. 123-126).
8. [421] Al-Thubaity, A., Alkhalifa, A., Almuhareb, A., & Alsanie, W. (2020). Arabic diacritization using bidirectional long short-term memory neural networks with conditional random fields. IEEE Access, 8, 154984-154996.
9. [422] Iordache, F., Georgescu, L., Oneață, D., & Cucu, H. (2019). Romanian Automatic Diacritics Restoration Challenge. In Proceedings of the 14th international conference Linguistic resources and tools for natural language processing (pp. 64-74).
10. [423] Ogheneruemu, Kingsley Lucky Ogheneovo. Development of Yoruba Diacritic Restoration for Under Dot and Diacritic Mark for Yoruba Text Using Deep Learning Model. MS thesis. Kwara State University (Nigeria), 2022.
11. [424] Özge, Asiye Tuba, Özge Bozal, and Umut Özge. "Diacritics correction in Turkish with context-aware sequence to sequence modeling." Turkish Journal of Electrical Engineering and Computer Sciences 30.6 (2022): 2433-2445.
12. [425] Esmail, Saeed, Kfir Bar, and Nachum Dershowitz. "How much does lookahead matter for disambiguation? partial arabic diacritization case study." Computational Linguistics 48.4 (2022): 1103-1123.
13. [417] Hifny, Yasser. "Recent advances in Arabic syntactic diacritics restoration." ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2021.

- [13] Beáta Lőrincz, **Maria Nuţu**, and Adriana Stan "Romanian Part of Speech Tagging using LSTM Networks". In *2019 IEEE 15th International Confer-ence on Intelligent Computer Communication and Processing (ICCP)*, IEEE Computer Society, pp. 223–228, 2019.

1. [426] Shafahat Sardarov. *Development and Design of Deep Learning-based Parts-of-Speech Tagging System for Azerbaijani language*, Thesis for Master of Science in Engineering in Computer Science, 2022, Khazar University, Azerbaijan
2. [427] Josipa Juričić. *Označavanje vrsta riječi pomoću neuronskih mreža*. Master thesis, University of Split, Faculty of Science. Department of Informatics, 2022.
3. [428] Harjanto, Shadifa Auliatama, and Ade Romadhony. "Question Template Extraction Using Sequence Labeling Approach." 2024 International Conference on Data Science and Its Applications (ICoDSA). IEEE, 2024.
4. [429] Aydinov, Farhad, et al. "Investigation of automatic part-of-speech tagging using CRF, HMM and LSTM on misspelled and edited texts." Proceedings of the 2022 5th artificial intelligence and cloud computing conference. 2022.

5. [430] Juričić, Josipa, and Branko Žitko. "POS-Only Tagging Using RNN for Croatian Language." International Conference on Digital Transformation in Education and Artificial Intelligence Application. Springer Nature Switzerland, 2023.

- [2] Adriana Mihaela Coroiu, Alina Delia Călin, and **Maria Nuţu**. Topic Modeling in Medical Data Analysis. Case Study Based on Medical Records Analysis. In *2019 International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, pp. 1–5, 2019.

  Citations:

  1. [431] Gupta, Aditi, and Hoor Fatima. "Topic Modeling in Healthcare: A Survey Study." NEUROQUANTOLOGY 20.11 (2022): 6214-6221.
  2. [432] Kenei, J., Opiyo, E., & Machii, J. Modeling and visualization of clinical texts to enhance meaningful and user-friendly information re-trieval. In Med. Sci. Forum, Vol. 1, February, 2022.

List of Grants

1. Grant from the Romanian Ministry of Research and Innovation, PCCDI – UEFISCDI, project number PN-III-P1-1.2-PCCDI-2017-0818/73.
Period: January 2019 to April 2021.

References

[1] A. S. Lundervold and A. Lundervold, "An overview of deep learning in medical imaging focusing on mri," *Zeitschrift für Medizinische Physik*, vol. 29, no. 2, pp. 102–127, 2019, Special Issue: Deep Learning in Medical Physics, ISSN: 0939-3889. DOI: https://doi.org/10.1016/j.zemedi.2018.11.002. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0939388918301181.

[2] A. M. Coroiu, A. D. Călin, and M. Nuţu, "Topic modeling in medical data analysis. case study based on medical records analysis," in *2019 International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, 2019, pp. 1–5. DOI: 10.23919/SOFTCOM.2019.8903900.

[3] Y. Wang *et al.*, "Tacotron: Towards End-to-End Speech Synthesis," in *Proc. Interspeech*, 2017. [Online]. Available: https://arxiv.org/abs/1703.10135.

[4] Y. Wang *et al.*, "Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis," *arXiv preprint arXiv:1803.09017*, 2018.

[5] N. Li, S. Liu, Y. Liu, S. Zhao, and M. Liu, "Neural speech synthesis with transformer network," in *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence*, Honolulu, Hawaii, USA: AAAI Press, 2019, ISBN: 978-1-57735-809-1. [Online]. Available: https://doi.org/10.1609/aaai.v33i01.33016706.

[6] K. Ito and L. Johnson, *The lj speech dataset*, https://keithito.com/LJ-Speech-Dataset/, 2017.

[7] H. Zen *et al.*, "Libritts: A corpus derived from librispeech for text-to-speech," *arXiv preprint arXiv:1904.02882*, 2019.

[8] A. Stan *et al.*, "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, Jul. 2017.

[9] B. Lőrincz, M. Nuţu, A. Stan, and M. Giurgiu, "An evaluation of postfiltering for deep learning based speech synthesis with limited data," in *2020 IEEE 10th International Conference on Intelligent Systems (IS)*, 2020, pp. 437–442. DOI: 10.1109/IS48319.2020.9199932.

[10] A. Stan, B. Lőrincz, M. Nuțu, and M. Giurgiu, "The mara corpus: Expressivity in end-to-end tts systems using synthesised speech data," in *2021 International Conference on Speech Technology and Human-Computer Dialogue (SpeD)*, 2021, pp. 85–90. DOI: 10.1109/SpeD53181.2021.9587438.

[11] M. Nuţu, "Deep learning approach for automatic romanian lemmatization," *Procedia Computer Science*, vol. 192, pp. 49–58, 2021, Knowledge-Based and Intelligent Information and Engineering Systems: Proceedings of the 25th International Conference KES2021, ISSN: 1877-0509. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050921014939.

[12] M. Nuţu, B. Lőrincz, and A. Stan, "Deep learning for automatic diacritics restoration in romanian," in *2019 IEEE 15th International Conference on Intelligent Computer Communication and Processing (ICCP)*, 2019, pp. 235–240. DOI: 10.1109/ICCP48234.2019.8959557.

[13] B. Lőrincz, M. Nuţu, and A. Stan, "Romanian part of speech tagging using lstm networks," in *2019 IEEE 15th International Conference on Intelligent Computer Communication and Processing (ICCP)*, IEEE, 2019, pp. 223–228.

[14] A. M. Coroiu, A. D. Călin, and M. Nuțu, "Communication style - an analysis from the perspective of automated learning," in *Artificial Neural Networks and Machine Learning – ICANN 2018*, V. Kůrková, Y. Manolopoulos, B. Hammer, L. Iliadis, and I. Maglogiannis, Eds., Cham: Springer International Publishing, 2018, pp. 589–597, ISBN: 978-3-030-01418-6.

[15] A. Turing, "Mathematical theory of enigma machine," *Public Record Office, London*, vol. 3, p. 150, 1940.

[16] B. J. Copeland, "Colossus: Its origins and originators," *IEEE Annals of the History of Computing*, vol. 26, no. 4, pp. 38–45, 2004.

[17] A. M. Turing, *Solvable and unsolvable problems*. Penguin Books London, 1954.

[18] A. M. Turing and J. Haugeland, "Computing machinery and intelligence," *The Turing Test: Verbal Behavior as the Hallmark of Intelligence*, pp. 29–56, 1950.

[19] R. Descartes and D. A. Cress, *Discourse on method*. Hackett Publishing, 1998.

[20] D. Diderot, *Pensées philosophiques*. Flammarion, 2007.

[21] M. J. M. Chuquicusma, S. Hussein, J. Burt, and U. Bagci, "How to fool radiologists with generative adversarial networks? a visual turing test for lung cancer diagnosis," in *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*, 2018, pp. 240–244. DOI: 10.1109/ISBI.2018.8363564.

[22] S. López-Fierro, "Emotional disorders: "if you pinch him, he will squeak". a new perspective on how machines can pass the turing test," in *2020 7th International Conference on Behavioural and Social Computing (BESC)*, 2020, pp. 1–5. DOI: 10.1109/BESC51023.2020.9348305.

[23] G. S. Vinayagam, P. Ezhilarasu, and J. Prakash, "Applications of turing machine as a string reverser for the three input characters — a review," in *2016 10th International Conference on Intelligent Systems and Control (ISCO)*, 2016, pp. 1–7. DOI: 10.1109/ISCO.2016.7726890.

[24] E. R. Vimina, "An automated reverse turing test using facial expressions," in *2009 2nd Conference on Human System Interactions*, 2009, pp. 314–317. DOI: 10.1109/HSI.2009.5090998.

[25] T.-Y. Chan, "Using a test-to-speech synthesizer to generate a reverse turing test," in *Proceedings. 15th IEEE International Conference on Tools with Artificial Intelligence*, 2003, pp. 226–232. DOI: 10.1109/TAI.2003.1250195.

[26] Z. Shao, G. Yang, and Z. Xu, "Industrial turing test: Concept and practice," in *2021 IEEE 1st International Conference on Digital Twins and Parallel Intelligence (DTPI)*, 2021, pp. 176–179. DOI: 10.1109/DTPI52967.2021.9540110.

[27] S. M. Shieber, "An analysis of the turing test," in *The Turing Test: Verbal Behavior as the Hallmark of Intelligence*. 2004, pp. 297–306.

[28] N. Chomsky, "Three models for the description of language," *IRE Transactions on information theory*, vol. 2, no. 3, pp. 113–124, 1956.

[29] C. A. Montgomery, "Linguistics and automated language processing," in *International Conference on Computational Linguistics COLING 1969: Preprint No. 41*, 1969.

[30] H. P. Edmundson, "New methods in automatic extracting," *J. ACM*, vol. 16, no. 2, pp. 264–285, Apr. 1969, ISSN: 0004-5411. DOI: 10.1145/321510.321519. [Online]. Available: https://doi.org/10.1145/321510.321519.

[31] A. Colmerauer, "An introduction to prolog iii," in *Computational Logic*, Springer, 1990, pp. 37–79.

[32] T. Winograd, "Shrdlu: A system for dialog," 1972.

[33] C. B. Asmussen and C. Møller, "Smart literature review: A practical topic modelling approach to exploratory literature review," *Journal of Big Data*, vol. 6, no. 1, pp. 1–18, 2019.

[34] C. Jacobi, W. Van Atteveldt, and K. Welbers, "Quantitative analysis of large amounts of journalistic texts using topic modelling," *Digital journalism*, vol. 4, no. 1, pp. 89–106, 2016.

[35] F. Shamrat *et al.*, "Sentiment analysis on twitter tweets about covid-19 vaccines using nlp and supervised knn classification algorithm," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 23, no. 1, pp. 463–470, 2021.

[36] A. Hasan, S. Moin, A. Karim, and S. Shamshirband, "Machine learning-based sentiment analysis for twitter accounts," *Mathematical and Computational Applications*, vol. 23, no. 1, p. 11, 2018.

[37] B. Trstenjak, S. Mikac, and D. Donko, "Knn with tf-idf based framework for text categorization," *Procedia Engineering*, vol. 69, pp. 1356–1364, 2014.

[38] B. Li, S. Yu, and Q. Lu, "An improved k-nearest neighbor algorithm for text categorization," *arXiv preprint cs/0306099*, 2003.

[39] T. Pranckevičius and V. Marcinkevičius, "Comparison of naive bayes, random forest, decision tree, support vector machines, and logistic regression classifiers for text reviews classification," *Baltic Journal of Modern Computing*, vol. 5, no. 2, p. 221, 2017.

[40] Y. LeCun, Y. Bengio, *et al.*, "Convolutional networks for images, speech, and time series," *The handbook of brain theory and neural networks*, vol. 3361, no. 10, p. 1995, 1995.

[41] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[42] K. He, X. Zhang, S. Ren, and J. Sun, *Deep residual learning for image recognition*, 2015. DOI: 10.48550/ARXIV.1512.03385. [Online]. Available: https://arxiv.org/abs/1512.03385.

[43] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[44] A. Vaswani *et al.*, *Attention is all you need*, 2017. DOI: 10.48550/ARXIV.1706.03762. [Online]. Available: https://arxiv.org/abs/1706.03762.

[45] I. Goodfellow *et al.*, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, Eds., vol. 27, Curran Associates, Inc., 2014.

[46] M. W. Gardner and S. Dorling, "Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences," *Atmospheric environment*, vol. 32, no. 14-15, pp. 2627–2636, 1998.

[47] Z. Wu, O. Watts, and S. King, "Merlin: An open source neural network speech synthesis system.," in *proceedings of the 9th International Speech Communication Association (ISCA) Speech Synthesis Workshop: SSW 2016*, Sunnyvale, United States: International Speech Communication Association (ISCA), 2016, pp. 202–207.

[48] H. Graves Abdel-Rahman, *Peephole long short-term memory, wikimedia commons*. [Online]. Available: `https://commons.wikimedia.org/wiki/`.

[49] S. Liao, J. Wang, R. Yu, K. Sato, and Z. Cheng, "Cnn for situations understanding based on sentiment analysis of twitter data," *Procedia computer science*, vol. 111, pp. 376–381, 2017.

[50] J. Wang, L.-C. Yu, K. R. Lai, and X. Zhang, "Dimensional sentiment analysis using a regional cnn-lstm model," in *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 2: Short papers)*, 2016, pp. 225–230.

[51] C. Dos Santos and M. Gatti, "Deep convolutional neural networks for sentiment analysis of short texts," in *Proceedings of COLING 2014, the 25th international conference on computational linguistics: technical papers*, 2014, pp. 69–78.

[52] Y. Kim, *Convolutional neural networks for sentence classification*, 2014. DOI: `10.48550/ARXIV.1408.5882`. [Online]. Available: `https://arxiv.org/abs/1408.5882`.

[53] R. Johnson and T. Zhang, "Effective use of word order for text categorization with convolutional neural networks," *arXiv preprint arXiv:1412.1058*, 2014.

[54] R. Johnson and T. Zhang, "Semi-supervised convolutional neural networks for text categorization via region embedding," *Advances in neural information processing systems*, vol. 28, 2015.

[55] T. H. Nguyen and R. Grishman, "Relation extraction: Perspective from convolutional neural networks," in *Proceedings of the 1st workshop on vector space modeling for natural language processing*, 2015, pp. 39–48.

[56] D. Zeng, K. Liu, S. Lai, G. Zhou, and J. Zhao, "Relation classification via convolutional deep neural network," in *Proceedings of COLING 2014, the 25th international conference on computational linguistics: technical papers*, 2014, pp. 2335–2344.

[57] Y. Bengio, R. Ducharme, and P. Vincent, "A neural probabilistic language model," *Advances in neural information processing systems*, vol. 13, 2000.

[58] W. De Mulder, S. Bethard, and M.-F. Moens, "A survey on the application of recurrent neural networks to statistical language modeling," *Computer Speech and Language*, vol. 30, no. 1, pp. 61–98, 2015.

[59] N.-Q. Pham, G. Kruszewski, and G. Boleda, "Convolutional neural network language models," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2016, pp. 1153–1162.

[60] M. Lin, Q. Chen, and S. Yan, "Network in network," *arXiv preprint arXiv:1312.4400*, 2013.

[61] M. Daniluk, T. Rocktäschel, J. Welbl, and S. Riedel, "Frustratingly short attention spans in neural language modeling," *arXiv preprint arXiv:1702.04521*, 2017.

[62] K. Benes, M. K. Baskar, and L. Burget, "Residual memory networks in language modeling: Improving the reputation of feed-forward networks.," in *INTERSPEECH*, 2017, pp. 284–288.

[63] Y. Kim, Y. Jernite, D. Sontag, and A. M. Rush, "Character-aware neural language models," in *Thirtieth AAAI conference on artificial intelligence*, 2016.

[64] J. Botha and P. Blunsom, "Compositional morphology for word representations and language modelling," in *International Conference on Machine Learning*, PMLR, 2014, pp. 1899–1907.

[65] R. Jozefowicz, O. Vinyals, M. Schuster, N. Shazeer, and Y. Wu, "Exploring the limits of language modeling," *arXiv preprint arXiv:1602.02410*, 2016.

[66] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[67] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," *Advances in neural information processing systems*, vol. 26, 2013.

[68] R. Socher *et al.*, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proceedings of the 2013 conference on empirical methods in natural language processing*, 2013, pp. 1631–1642.

[69] R. Socher, J. Bauer, C. D. Manning, and A. Y. Ng, "Parsing with compositional vector grammars," in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2013, pp. 455–465.

[70] T. Fujisaki, F. Jelinek, J. Cocke, E. Black, and T. Nishino, "A probabilistic parsing method for sentence disambiguation," in *Current issues in parsing technology*, Springer, 1991, pp. 139–152.

[71] F. Jelinek, J. D. Lafferty, and R. L. Mercer, "Basic methods of probabilistic context free grammars," in *Speech recognition and understanding*, Springer, 1992, pp. 345–360.

[72] P. Le and W. Zuidema, "The inside-outside recursive neural network model for dependency parsing," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 729–739.

[73] P. Stenetorp, "Transition-based dependency parsing using recursive neural networks," in *NIPS Workshop on Deep Learning*, Citeseer, 2013.

[74] D. Chen and C. D. Manning, "A fast and accurate dependency parser using neural networks," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 740–750.

[75] H. Zhou, Y. Zhang, S. Huang, and J. Chen, "A neural probabilistic structured-prediction model for transition-based dependency parsing," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2015, pp. 1213–1222.

[76] D. Weiss, C. Alberti, M. Collins, and S. Petrov, "Structured training for neural network transition-based parsing," *arXiv preprint arXiv:1506.06158*, 2015.

[77] C. Dyer, M. Ballesteros, W. Ling, A. Matthews, and N. A. Smith, "Transition-based dependency parsing with stack long short-term memory," *arXiv preprint arXiv:1505.08075*, 2015.

[78] D. Andor *et al.*, "Globally normalized transition-based neural networks," *arXiv preprint arXiv:1603.06042*, 2016.

[79] Y. Wang, W. Che, J. Guo, and T. Liu, "A neural transition-based approach for semantic dependency graph parsing," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, 2018.

[80] C. Dyer, A. Kuncoro, M. Ballesteros, and N. A. Smith, *Recurrent neural network grammars*, 2016. DOI: 10.48550/ARXIV.1602.07776. [Online]. Available: https://arxiv.org/abs/1602.07776.

[81] E. Charniak *et al.*, "Parsing as language modeling," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2016, pp. 2331–2336.

[82] D. Fried, M. Stern, and D. Klein, "Improving neural parsing by disentangling model combination and reranking effects," *arXiv preprint arXiv:1707.03058*, 2017.

[83] T. Dozat and C. D. Manning, "Simpler but more accurate semantic dependency parsing," *arXiv preprint arXiv:1807.01396*, 2018.

[84] Z. Tan, M. Wang, J. Xie, Y. Chen, and X. Shi, "Deep semantic role labeling with self-attention," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, 2018.

[85] L. Duong, H. Afshar, D. Estival, G. Pink, P. R. Cohen, and M. Johnson, "Active learning for deep semantic parsing," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2018, pp. 43–48.

[86] B. Hu, Z. Lu, H. Li, and Q. Chen, "Convolutional neural network architectures for matching natural language sentences," *Advances in neural information processing systems*, vol. 27, 2014.

[87] A. Bordes, X. Glorot, J. Weston, and Y. Bengio, "A semantic matching energy function for learning with multi-relational data," *Machine Learning*, vol. 94, no. 2, pp. 233–259, 2014.

[88] W. Yin and H. Schütze, "Convolutional neural network for paraphrase identification," in *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2015, pp. 901–911.

[89] W. Dolan, C. Quirk, C. Brockett, and B. Dolan, "Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources," 2004.

[90] H. He, K. Gimpel, and J. Lin, "Multi-perspective sentence similarity modeling with convolutional neural networks," in *Proceedings of the 2015 conference on empirical methods in natural language processing*, 2015, pp. 1576–1586.

[91] K. S. Tai, R. Socher, and C. D. Manning, "Improved semantic representations from tree-structured long short-term memory networks," *arXiv preprint arXiv:1503.00075*, 2015.

[92] H. He and J. Lin, "Pairwise word interaction modeling with deep neural networks for semantic similarity measurement," in *Proceedings of the 2016 conference of the north American chapter of the Association for Computational Linguistics: human language technologies*, 2016, pp. 937–948.

[93] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *International conference on machine learning*, PMLR, 2014, pp. 1188–1196.

[94] A. Go, R. Bhayani, and L. Huang, "Twitter sentiment classification using distant supervision," *CS224N project report, Stanford*, vol. 1, no. 12, p. 2009, 2009.

[95] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A convolutional neural network for modelling sentences," *arXiv preprint arXiv:1404.2188*, 2014.

[96] X. Li and D. Roth, "Learning question classifiers," in *COLING 2002: The 19th International Conference on Computational Linguistics*, 2002.

[97] A. Poliak, Y. Belinkov, J. Glass, and B. Van Durme, "On the evaluation of semantic phenomena in neural machine translation using natural language inference," *arXiv preprint:1804.09779*, 2018.

[98] A. Poliak, J. Naradowsky, A. Haldar, R. Rudinger, and B. Van Durme, "Hypothesis only baselines in natural language inference," *arXiv preprint:1805.01042*, 2018.

[99] J. Herzig and J. Berant, "Neural semantic parsing over multiple knowledge-bases," *arXiv preprint arXiv:1702.01569*, 2017.

[100] M.-T. Luong, R. Socher, and C. D. Manning, "Better word representations with recursive neural networks for morphology," in *Proceedings of the seventeenth conference on computational natural language learning*, 2013, pp. 104–113.

[101] Y. Belinkov, N. Durrani, F. Dalvi, H. Sajjad, and J. Glass, "What do neural machine translation models learn about morphology?" *arXiv preprint:1704.03471*, 2017.

[102] H. Morita, D. Kawahara, and S. Kurohashi, "Morphological analysis for unsegmented languages using recurrent neural network language model," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015, pp. 2292–2297.

[103] M. Dehouck and P. Denis, "A framework for understanding the role of morphology in universal dependency parsing," in *EMNLP 2018-Conference on Empirical Methods in Natural Language Processing*, 2018.

[104] D. W. Otter, J. R. Medina, and J. K. Kalita, "A survey of the usages of deep learning for natural language processing," *IEEE transactions on neural networks and learning systems*, vol. 32, no. 2, pp. 604–624, 2020.

[105] X. Tan, T. Qin, F. Soong, and T.-Y. Liu, "A survey on neural speech synthesis," *arXiv preprint:*
2106.15561, 2021.

[106] M. Yang, "A survey on few-shot learning in natural language processing," in *2021 International Conference on Artificial Intelligence and Electromechanical Automation (AIEA)*, 2021, pp. 294–297. DOI: 10.1109/AIEA53260.2021.00069.

[107] P.-S. Huang, X. He, J. Gao, L. Deng, A. Acero, and L. Heck, "Learning deep structured semantic models for web search using clickthrough data," in *Proceedings of the 22nd ACM international conference on Information and Knowledge Management*, 2013, pp. 2333–2338.

[108] Y. Shen, X. He, J. Gao, L. Deng, and G. Mesnil, "Learning semantic representations using convolutional neural networks for web search," in *Proceedings of the 23rd international conference on world wide web*, 2014, pp. 373–374.

[109] Z. Lu and H. Li, "A deep architecture for matching short texts," *Advances in neural information processing systems*, vol. 26, 2013.

[110] L. Pang, Y. Lan, J. Guo, J. Xu, S. Wan, and X. Cheng, "Text matching as image recognition," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, 2016.

[111] J. Guo, Y. Fan, Q. Ai, and W. B. Croft, "A deep relevance matching model for ad-hoc retrieval," in *Proceedings of the 25th ACM international on conference on information and knowledge management*, 2016, pp. 55–64.

[112] H. Zamani, M. Dehghani, W. B. Croft, E. Learned-Miller, and J. Kamps, "From neural re-ranking to neural ranking: Learning a sparse representation for inverted indexing," in *Proceedings of the 27th ACM international conference on information and knowledge management*, 2018, pp. 497–506.

[113] S. MacAvaney, A. Yates, A. Cohan, and N. Goharian, "Cedr: Contextualized embeddings for document ranking," in *Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval*, 2019, pp. 1101–1104.

[114] W. Wang, N. Yang, F. Wei, B. Chang, and M. Zhou, "Gated self-matching networks for reading comprehension and question answering," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2017, pp. 189–198.

[115] A. Santoro *et al.*, "A simple neural network module for relational reasoning," *Advances in neural information processing systems*, vol. 30, 2017.

[116] W. Yang *et al.*, "End-to-end open-domain question answering with bertserini," *arXiv preprint arXiv:1902.01718*, 2019.

[117] J. Hammerton, "Named entity recognition with long short-term memory," in *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003*, 2003, pp. 172–175.

[118] C. N. d. Santos and V. Guimaraes, "Boosting named entity recognition with neural character embeddings," *arXiv preprint arXiv:1505.05008*, 2015.

[119] J. P. Chiu and E. Nichols, "Named entity recognition with bidirectional lstm-cnns," *Transactions of the association for computational linguistics*, vol. 4, pp. 357–370, 2016.

[120] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, "Neural architectures for named entity recognition," *arXiv preprint arXiv:1603.01360*, 2016.

[121] A. Akbik, D. Blythe, and R. Vollgraf, "Contextual string embeddings for sequence labeling," in *Proceedings of the 27th international conference on computational linguistics*, 2018, pp. 1638–1649.

[122] Y. Chen, L. Xu, K. Liu, D. Zeng, and J. Zhao, "Event extraction via dynamic multi-pooling convolutional neural networks," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2015, pp. 167–176.

[123] T. H. Nguyen, K. Cho, and R. Grishman, "Joint event extraction via recurrent neural networks," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016, pp. 300–309.

[124] X. Liu, H. Huang, and Y. Zhang, "Open domain event extraction using neural latent variable models," *arXiv preprint arXiv:1906.06947*, 2019.

[125] S. Zheng *et al.*, "Joint entity and relation extraction based on a hybrid neural network," *Neurocomputing*, vol. 257, pp. 59–66, 2017.

[126] M. Sun, X. Li, X. Wang, M. Fan, Y. Feng, and P. Li, "Logician: A unified end-to-end neural approach for open-domain information extraction," in *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, 2018, pp. 556–564.

[127] C. Lin, T. Miller, D. Dligach, S. Bethard, and G. Savova, "A bert-based universal model for both within-and cross-sentence clinical temporal relation extraction," in *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, 2019, pp. 65–71.

[128] J. Wei, Q. Zhou, and Y. Cai, "Poet-based poetry generation: Controlling personal style with recurrent neural networks," in *2018 International Conference on Computing, Networking and Communications (ICNC)*, IEEE, 2018, pp. 156–160.

[129] J. Hopkins and D. Kiela, "Automatically generating rhythmic verse with neural networks," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2017, pp. 168–178.

[130] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, *et al.*, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[131] B. Bena and J. Kalita, "Introducing aspects of creativity in automatic poetry generation," *arXiv preprint arXiv:2002.02511*, 2020.

[132] K.-L. Lo, R. Ariss, and P. Kurz, *Gpoet-2: A gpt-2 based poem generator*, 2022. DOI: 10.48550/ARXIV.2205.08847. [Online]. Available: https://arxiv.org/abs/2205.08847.

[133] Z. Yu, J. Tan, and X. Wan, "A neural approach to pun generation," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2018, pp. 1650–1660.

[134] H. Ren and Q. Yang, "Neural joke generation," *Final Project Reports of Course CS224n*, 2017.

[135] B. Chippada and S. Saha, "Knowledge amalgam: Generating jokes and quotes together," *arXiv preprint arXiv:1806.04387*, 2018.

[136] P. Jain, P. Agrawal, A. Mishra, M. Sukhwani, A. Laha, and K. Sankaranarayanan, "Story generation from sequence of independent short descriptions," *arXiv preprint arXiv:1707.05501*, 2017.

[137] N. Peng, M. Ghazvininejad, J. May, and K. Knight, "Towards controllable story generation," in *Proceedings of the First Workshop on Storytelling*, 2018, pp. 43–49.

[138] L. Martin *et al.*, "Event representations for automated story generation with deep neural nets," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, 2018.

[139] E. Clark, Y. Ji, and N. A. Smith, "Neural text generation in stories using entity representations as context," in *Proceedings of the 2018 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long papers)*, 2018, pp. 2250–2260.

[140] J. Xu, X. Ren, Y. Zhang, Q. Zeng, X. Cai, and X. Sun, "A skeleton-based model for promoting coherence among sentences in narrative story generation," *arXiv preprint arXiv:1808.06945*, 2018.

[141] M. Drissi, O. Watkins, and J. Kalita, "Hierarchical text generation using an outline," *arXiv preprint arXiv:1810.08802*, 2018.

[142] Q. Huang, Z. Gan, A. Celikyilmaz, D. Wu, J. Wang, and X. He, "Hierarchically structured reinforcement learning for topically coherent visual story generation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 8465–8472.

[143] A. Fan, M. Lewis, and Y. Dauphin, "Hierarchical neural story generation," *arXiv preprint arXiv:1805.04833*, 2018.

[144] J. Li, M.-T. Luong, and D. Jurafsky, "A hierarchical neural autoencoder for paragraphs and documents," *arXiv preprint arXiv:1506.01057*, 2015.

[145] K. Lin, D. Li, X. He, Z. Zhang, and M.-T. Sun, "Adversarial ranking for language generation," *Advances in neural information processing systems*, vol. 30, 2017.

[146] Y. Zhang *et al.*, "Adversarial feature matching for text generation," in *International Conference on Machine Learning*, PMLR, 2017, pp. 4006–4015.

[147] L. Chen *et al.*, "Adversarial text generation via feature-mover's distance," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[148] J. Guo, S. Lu, H. Cai, W. Zhang, Y. Yu, and J. Wang, "Long text generation via adversarial training with leaked information," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, 2018.

[149] I. Serban *et al.*, "A hierarchical latent variable encoder-decoder model for generating dialogues," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31, 2017.

[150] Z. Hu, Z. Yang, X. Liang, R. Salakhutdinov, and E. P. Xing, "Toward controlled generation of text," in *International conference on machine learning*, PMLR, 2017, pp. 1587–1596.

[151] W. Wang *et al.*, "Topic-guided variational autoencoders for text generation," *arXiv preprint arXiv:1903.07137*, 2019.

[152] A. Holtzman, J. Buys, L. Du, M. Forbes, and Y. Choi, "The curious case of neural text degeneration," *arXiv preprint arXiv:1904.09751*, 2019.

[153] Y. Zhang and B. Wallace, "A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification," *arXiv preprint arXiv:1510.03820*, 2015.

[154] A. Conneau, H. Schwenk, L. Barrault, and Y. Lecun, "Very deep convolutional networks for text classification," *arXiv preprint arXiv:1606.01781*, 2016.

[155] M. Jiang *et al.*, "Text classification based on deep belief network and softmax regression," *Neural Computing and Applications*, vol. 29, no. 1, pp. 61–70, 2018.

[156] A. Adhikari, A. Ram, R. Tang, and J. Lin, "Docbert: Bert for document classification," *arXiv preprint arXiv:1904.08398*, 2019.

[157] J. Worsham and J. Kalita, "Genre identification and the compositional effect of genre in literature," in *Proceedings of the 27th international conference on computational linguistics*, 2018, pp. 1963–1973.

[158] A. M. Rush, S. Chopra, and J. Weston, "A neural attention model for abstractive sentence summarization," *arXiv preprint arXiv:1509.00685*, 2015.

[159] J. Krantz and J. Kalita, "Abstractive summarization using attentive neural techniques," *arXiv preprint arXiv:1810.08838*, 2018.

[160] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, "Convolutional sequence to sequence learning," in *International conference on machine learning*, PMLR, 2017, pp. 1243–1252.

[161] H. Zhang, J. Xu, and J. Wang, "Pretraining-based natural language generation for text summarization," *arXiv preprint arXiv:1902.09243*, 2019.

[162] D. Bahdanau, K. Cho, and Y. Bengio, *Neural machine translation by jointly learning to align and translate*, 2014. DOI: 10.48550/ARXIV.1409.0473. [Online]. Available: https://arxiv.org/abs/1409.0473.

[163] N. Kalchbrenner and P. Blunsom, "Recurrent continuous translation models," in *Proceedings of the 2013 conference on empirical methods in natural language processing*, 2013, pp. 1700–1709.

[164] K. Cho *et al.*, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[165] Y. Wu *et al.*, "Google's neural machine translation system: Bridging the gap between human and machine translation," *arXiv preprint arXiv:1609.08144*, 2016.

[166] D. Britz, A. Goldie, M.-T. Luong, and Q. Le, "Massive exploration of neural machine translation architectures," *arXiv preprint arXiv:1703.03906*, 2017.

[167] R. Sennrich *et al.*, "Nematus: A toolkit for neural machine translation," *arXiv preprint arXiv:1703.04357*, 2017.

[168] S. Ranathunga, E.-S. A. Lee, M. P. Skenduli, R. Shekhar, M. Alam, and R. Kaur, "Neural machine translation for low-resource languages: A survey," *arXiv preprint arXiv:2106.15115*, 2021.

[169] M. X. Chen *et al.*, "The best of both worlds: Combining recent advances in neural machine translation," *arXiv preprint:1804.09849*, 2018.

[170] M. Denkowski and G. Neubig, "Stronger baselines for trustable results in neural machine translation," *arXiv preprint arXiv:1706.09733*, 2017.

[171] V. N. Vapnik, "Adaptive and learning systems for signal processing communications, and control," *Statistical learning theory*, 1998.

[172] M. Peyrard, W. Zhao, S. Eger, and R. West, "Better than average: Paired evaluation of NLP systems," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Association for Computational Linguistics, 2021. DOI: 10.18653/v1/2021.acl-long.179. [Online]. Available: https://doi.org/10.18653%2Fv1%2F2021.acl-long.179.

[173] R. Iyer, M. Ostendorf, and M. Meteer, "Analyzing and predicting language model improvements," in *1997 IEEE Workshop on Automatic Speech Recognition and Understanding Proceedings*, IEEE, 1997, pp. 254–261.

[174] S. F. Chen, D. Beeferman, and R. Rosenfeld, "Evaluation metrics for language models," 1998.

[175] P. Clarkson and T. Robinson, "Improved language modelling through better language model evaluation measures," *Computer Speech and Language*, vol. 15, no. 1, pp. 39–53, 2001.

[176] D. Tufiş and A. Chiţu, "Automatic insertion of diacritics in romanian texts," in *Proceedings of the 5th International Workshop on Computational Lexicography COMPLEX*, 1999, pp. 185–194.

[177] M. Simard, "Automatic insertion of accents in french text," in *Proceedings of the Third Conference on Empirical Methods for Natural Language Processing*, 1998, pp. 27–35.

[178] R. Mihalcea and V. Nastase, "Letter level learning for language independent diacritics restoration," in *proceedings of the 6th conference on Natural language learning-Volume 20*, Association for Computational Linguistics, 2002, pp. 1–7.

[179] K.-H. Nguyen and C.-Y. Ock, "Diacritics restoration in vietnamese: Letter based vs. syllable based model," in *Pacific Rim International Conference on Artificial Intelligence*, Springer, 2010, pp. 631–636.

[180] J. Náplava, M. Straka, P. Straňák, and J. Hajic, "Diacritics restoration using neural networks," in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*, 2018.

[181] R. F. Mihalcea, "Diacritics restoration: Learning from letters versus learning from words," in *International Conference on Intelligent Text Processing and Computational Linguistics*, Springer, 2002, pp. 339–348.

[182] C. Ungurean, D. Burileanu, V. Popescu, C. Negrescu, and A. Dervis, "Automatic toration for a tts-based e-mail reader application," *UPB Scientific Bulletin, Series C*, vol. 70, no. 4, pp. 3–12, 2008.

[183] L. Petrică, H. Cucu, A. Buzo, and C. Burileanu, "A robust diacritics restoration system using unreliable raw text data," in *Spoken Language Technologies for Under-Resourced Languages*, 2014.

[184] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, 2014, pp. 3104–3112.

[185] V. B. Mititelu, E. Irimia, and D. Tufis, "Corola—the reference corpus of contemporary romanian language.," in *LREC*, 2014, pp. 1235–1239.

[186] H. Cristescu, *Romanian diacritic restoration with neural nets*. [Online]. Available: https://github.com/horiacristescu/romanian-diacritic-restoration.

[187] J. Shen *et al.*, "Natural tts synthesis by conditioning wavenet on mel spectrogram predictions," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Calgary, AB, Canada: IEEE, 2018, pp. 4779–4783.

[188] Y. Wang *et al.*, *Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis*, 2018. arXiv: 1803.09017 [cs.CL].

[189] T. Boros, S. D. Dumitrescu, and V. Pais, "Tools and resources for romanian text-to-speech and speech-to-text applications," *arXiv preprint arXiv:1802.05583*, 2018.

[190] O. Zine, A. Meziane, and M. Boudchiche, "Towards a high-quality lemma-based text-to-speech system for the arabic language," in *International Conference on Arabic Language Processing*, Springer, Fez, Morocco: Springer, Cham, 2017, pp. 53–66.

[191] M. Pouget, O. Nahorna, T. Hueber, and G. Bailly, "Adaptive latency for part-of-speech tagging in incremental text-to-speech synthesis," in *Proceedings of the 17th Annual Conference of the International Speech Communication Association (INTERSPEECH 2016): Understanding Speech Processing in Humans and Machines*, San Francisco, USA: International Speech Communication Association, 2016, pp. 2846–2850.

[192] L. Rheault, K. Beelen, C. Cochrane, and G. Hirst, "Measuring emotion in parliamentary debates with automated textual analysis," *PloS one*, vol. 11, no. 12, e0168843, 2016.

[193] B. Desmet and V. Hoste, "Emotion detection in suicide notes," *Expert Systems with Applications*, vol. 40, no. 16, pp. 6351–6358, 2013.

[194] A. Pak, D. Bernhard, P. Paroubek, and C. Grouin, "A combined approach to emotion detection in suicide notes," *Biomedical informatics insights*, vol. 5, BII–S8969, 2012.

[195] G. Badaro, H. Jundi, H. Hajj, W. El-Hajj, and N. Habash, "Arsel: A large scale arabic sentiment and emotion lexicon," *OSACT*, vol. 3, p. 26, 2018.

[196] T. El-Shishtawy and F. El-Ghannam, *A lemma based evaluator for semitic language text summarization systems*, 2014. arXiv: 1403.5596 [cs.CL].

[197] L. Skorkovská, "Application of lemmatization and summarization methods in topic identification module for large scale language modeling data filtering," in *International Conference on Text, Speech and Dialogue*, Berlin, Heidelberg: Springer, 2012, pp. 191–198.

[198] C. Manolache, H. Cucu, and C. Burileanu, "Lemma-based dynamic time warping search for keyword spotting applications in romanian," in *2019 International Conference on Speech Technology and Human-Computer Dialogue (SpeD)*, Timisoara, Romania: IEEE, 2019, pp. 1–9.

[199] E. Irimia, "Ebmt experiments for the english-romanian language pair," in *Recent Advances in Intelligent Information Systems*, Poland: Academic publishing house EXIT, Warsaw, 2009, pp. 91–102.

[200] T. Boroş, "A unified lexical processing framework based on the margin infused relaxed algorithm. a case study on the romanian language," in *Proceedings of the International Conference Recent Advances in Natural Language Processing RANLP 2013*, Hissar, Bulgaria: INCOMA Ltd. Shoumen, 2013, pp. 91–97.

[201] A. Chakrabarty, O. A. Pandit, and U. Garain, "Context sensitive lemmatization using two successive bidirectional gated recurrent networks," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vancouver, Canada: Association for Computational Linguistics, 2017, pp. 1481–1491.

[202] T. Boroş, S. D. Dumitrescu, and R. Burtica, "Nlp-cube: End-to-end raw text processing with neural networks," in *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, Brussels, Belgium: Association for Computational Linguistics, 2018, pp. 171–179.

[203] E. Yildiz and A. C. Tantuğ, "Morpheus: A neural network for jointly learning contextual lemmatization and morphological tagging," in *Proceedings of the 16th Workshop on Computational Research in Phonetics, Phonology, and Morphology*, 2019, pp. 25–34.

[204] G. Chrupała, G. Dinu, and J. Van Genabith, "Learning morphology with morfette," in *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco: European Language Resources Association (ELRA), 2008, pp. 2362–2367.

[205] P. Qi, Y. Zhang, Y. Zhang, J. Bolton, and C. D. Manning, *Stanza: A python natural language processing toolkit for many human languages*, 2020. arXiv: 2003.07082 [cs.CL].

[206] J. Kanerva, F. Ginter, and T. Salakoski, *Universal lemmatizer: A sequence to sequence model for lemmatizing universal dependencies treebanks*, 2019. arXiv: 1902.00972 [cs.CL].

[207] M. Straka, J. Straková, and J. Hajič, *Evaluating contextualized embeddings on 54 languages in pos tagging, lemmatization and dependency parsing*, 2019. arXiv: 1908.07448 [cs.CL].

[208] S. D. Dumitrescu and T. Boros, "Attention-free encoder decoder for morphological processing," in *Proceedings of the CoNLL SIGMORPHON 2018 Shared Task: Universal Morphological Reinflection, Brussels, October 31 - November 1, 2018*, M. Hulden and R. Cotterell, Eds., Association for Computational Linguistics, 2018, pp. 64–68. DOI: 10.18653/v1/k18-3007. [Online]. Available: https://doi.org/10.18653/v1/k18-3007.

[209] K. Crammer and Y. Singer, "Ultraconservative online algorithms for multiclass problems," *Journal of Machine Learning Research*, vol. 3, no. Jan, pp. 951–991, 2003.

[210] E. W. Myers, "Ano (nd) difference algorithm and its variations," *Algorithmica*, vol. 1, no. 1-4, pp. 251–266, 1986.

[211] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, Minneapolis, Minnesota: Association for Computational Linguistics, 2019, pp. 4171–4186.

[212] A. Akbik, T. Bergmann, D. Blythe, K. Rasul, S. Schweter, and R. Vollgraf, "Flair: An easy-to-use framework for state-of-the-art nlp," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, Minneapolis, Minnesota: Association for Computational Linguistics, 2019, pp. 54–59.

[213] M. E. Peters *et al.*, *Deep contextualized word representations*, 2018. arXiv: 1802.05365 [cs.CL].

[214] R. Cotterell *et al.*, "The conll–sigmorphon 2018 shared task: Universal morphological reinflection," *arXiv preprint arXiv:1810.07125*, 2018.

[215] T. Erjavec, "Multext-east morphosyntactic specifications: Version 3.0," *Supported By EU Projects Multext-East, Concede And TELRI*, 2004.

[216] D. Tufis and O. Mason, "Tagging romanian texts: A case study for qtag, a language independent probabilistic tagger," in *Proceedings of the First International Conference on Language Resources and Evaluation (LREC)*, vol. 1, 1998, p. 143.

[217] T. Boroş and S. D. Dumitrescu, "Improving the racai neural network msd tagger," in *International Conference on Engineering Applications of Neural Networks*, Springer, 2013, pp. 42–51.

[218] R. Simionescu, "Graphical grammar studio as a constraint grammar solution for part of speech tagging," in *The Conference on Linguistic Resources and Instruments for Romanian Language Processing*, vol. 152, 2011.

[219] R. Simionescu, "Hybrid pos tagger," in *Proceedings of Language Resources and Tools with Industrial Applications Workshop (Eurolan 2011 Summer School), Cluj-Napoca, Romania*, Citeseer, 2011, pp. 21–28.

[220] O. Frunza, D. Inkpen, and D. Nadeau, "A text processing tool for the romanian language," in *Proc. of the EuroLAN 2005 Workshop on Cross-Language Knowledge Induction*, Citeseer, 2005.

[221] L. R. Teodorescu, R. Boldizsar, M. Ordean, M. Duma, L. Detesan, and M. Ordean, "Part of speech tagging for romanian text-to-speech system," in *2011 13th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing*, IEEE, 2011, pp. 153–159.

[222] T. Horsmann and T. Zesch, "Do lstms really work so well for pos tagging?–a replication study," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017, pp. 727–736.

[223] B. Plank, A. Søgaard, and Y. Goldberg, "Multilingual part-of-speech tagging with bidirectional long short-term memory models and auxiliary loss," *arXiv preprint arXiv:1604.05529*, 2016.

[224] A. Kalantari, A. Kamsin, S. Shamshirband, A. Gani, H. Alinejad-Rokny, and A. T. Chrono-poulos, "Computational intelligence approaches for classification of medical data: State-of-the-art, future challenges and research directions," *Neurocomputing*, vol. 276, pp. 2–22, 2018.

[225] S. Boyapati, S. R. Swarna, V. Dutt, and N. Vyas, "Big data approach for medical data classification: A review study," in *2020 3rd International Conference on Intelligent Sustainable Systems (ICISS)*, IEEE, 2020, pp. 762–766.

[226] R. K. Bania and A. Halder, "R-ensembler: A greedy rough set based ensemble attribute selection algorithm with knn imputation for classification of medical data," *Computer methods and programs in biomedicine*, vol. 184, p. 105 122, 2020.

[227] L.-Y. Hu, M.-W. Huang, S.-W. Ke, and C.-F. Tsai, "The distance function effect on k-nearest neighbor classification for medical datasets," *SpringerPlus*, vol. 5, no. 1, pp. 1–9, 2016.

[228] B. Remeseiro and V. Bolon-Canedo, "A review of feature selection methods in medical applications," *Computers in biology and medicine*, vol. 112, p. 103 375, 2019.

[229] H. Alashwal, M. El Halaby, J. J. Crouse, A. Abdalla, and A. A. Moustafa, "The application of unsupervised clustering methods to alzheimer's disease," *Frontiers in computational neuroscience*, vol. 13, p. 31, 2019.

[230] A. Ahmad and S. S. Khan, "Survey of state-of-the-art mixed data clustering algorithms," *Ieee Access*, vol. 7, pp. 31 883–31 902, 2019.

[231] Y. Jiang *et al.*, "A novel distributed multitask fuzzy clustering algorithm for automatic mr brain image segmentation," *Journal of medical systems*, vol. 43, no. 5, pp. 1–9, 2019.

[232] V. Zarikas, S. G. Poulopoulos, Z. Gareiou, and E. Zervas, "Clustering analysis of countries using the covid-19 cases dataset," *Data in brief*, vol. 31, p. 105 787, 2020.

[233] W. Zhao, W. Zou, and J. J. Chen, "Topic modeling for cluster analysis of large biological and medical datasets," in *BMC bioinformatics*, BioMed Central, vol. 15, 2014, pp. 1–11.

[234] F. Ali *et al.*, "A smart healthcare monitoring system for heart disease prediction based on ensemble deep learning and feature fusion," *Information Fusion*, vol. 63, pp. 208–222, 2020.

[235] I. H. Sarker, "Machine learning: Algorithms, real-world applications and research directions," *SN Computer Science*, vol. 2, no. 3, pp. 1–21, 2021.

[236] M. A. Lambay and S. P. Mohideen, "Big data analytics for healthcare recommendation systems," in *2020 International Conference on System, Computation, Automation and Networking (ICSCAN)*, IEEE, 2020, pp. 1–6.

[237] A. Z. Woldaregay, E. Årsand, T. Botsis, D. Albers, L. Mamykina, and G. Hartvigsen, "Data-driven blood glucose pattern classification and anomalies detection: Machine-learning applications in type 1 diabetes," *Journal of medical Internet research*, vol. 21, no. 5, e11030, 2019.

[238] M. Nawir, A. Amir, N. Yaakob, and O. B. Lynn, "Effective and efficient network anomaly detection system using machine learning algorithm," *Bulletin of Electrical Engineering and Informatics*, vol. 8, no. 1, pp. 46–51, 2019.

[239] Y. Bao, Z. Tang, H. Li, and Y. Zhang, "Computer vision and deep learning–based data anomaly detection method for structural health monitoring," *Structural Health Monitoring*, vol. 18, no. 2, pp. 401–421, 2019.

[240] M. B. van Egmond *et al.*, "Privacy-preserving dataset combination and lasso regression for healthcare predictions," *BMC medical informatics and decision making*, vol. 21, no. 1, pp. 1–16, 2021.

[241] L. H. Salazar, A. M. R. Fernandes, R. Dazzi, J. Raduenz, N. M. Garcia, and V. R. Q. Leithardt, "Prediction of attendance at medical appointments based on machine learning," in *2020 15th Iberian Conference on Information Systems and Technologies (CISTI)*, 2020, pp. 1–6. DOI: 10.23919/CISTI49556.2020.9140973.

[242] A. Marquardt *et al.*, "Healthscope: An interactive distributed data mining framework for scalable prediction of healthcare costs," in *2014 IEEE International Conference on Data Mining Workshop*, 2014, pp. 1227–1230. DOI: 10.1109/ICDMW.2014.45.

[243] J. Waring, C. Lindvall, and R. Umeton, "Automated machine learning: Review of the state-of-the-art and opportunities for healthcare," *Artificial Intelligence in Medicine*, vol. 104, p. 101 822, 2020, ISSN: 0933-3657. DOI: https://doi.org/10.1016/j.artmed.2020.101822. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0933365719310437.

[244] S. Dong *et al.*, "Rating hospital performance in china: Review of publicly available measures and development of a ranking system," *Journal of medical Internet research*, vol. 23, no. 6, e17095, 2021.

[245] J. Ni, H. Fei, W. Fan, and X. Zhang, "Automated medical diagnosis by ranking clusters across the symptom-disease network," in *2017 IEEE International Conference on Data Mining (ICDM)*, 2017, pp. 1009–1014. DOI: 10.1109/ICDM.2017.130.

[246] M. Bhattacharya, C. Jurkovitz, and H. Shatkay, "Identifying patterns of associated-conditions through topic models of electronic medical records," in *2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 2016, pp. 466–469. DOI: 10.1109/BIBM.2016.7822561.

[247] K. Khawaji, I. Almubark, A. Almalki, and B. Taylor, "Similarity matching for workflows in medical domain using topic modeling," in *2018 IEEE World Congress on Services (SERVICES)*, 2018, pp. 19–20. DOI: 10.1109/SERVICES.2018.00023.

[248] X. Lin, M. Liu, and J. Zhang, "A top-down binary hierarchical topic model for biomedical literature," *IEEE Access*, vol. 8, pp. 59 870–59 882, 2020. DOI: 10.1109/ACCESS.2020.2983265.

[249] M. Hajjem and C. Latiri, "Combining ir and lda topic modeling for filtering microblogs," *Procedia Computer Science*, vol. 112, pp. 761–770, 2017, Knowledge-Based and Intelligent Information and Engineering Systems: Proceedings of the 21st International Conference, KES2017, 6-8 September 2017, Marseille, France, ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2017.08.166. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050917315235.

[250] Z. Tong and H. Zhang, "A text mining research based on lda topic modelling," in *International conference on computer science, engineering and information technology*, 2016, pp. 201–210.

[251] D. Zhang, T. Luo, and D. Wang, "Learning from lda using deep neural networks," in *Natural Language Understanding and Intelligent Applications*, C.-Y. Lin, N. Xue, D. Zhao, X. Huang, and Y. Feng, Eds., Cham: Springer International Publishing, 2016, pp. 657–664, ISBN: 978-3-319-50496-4.

[252] R. Dinga *et al.*, "Predicting the naturalistic course of depression from a wide range of clinical, psychological, and biological data: A machine learning approach," *Translational psychiatry*, vol. 8, no. 1, pp. 1–11, 2018.

[253] C. Stachl *et al.*, "Personality research and assessment in the era of machine learning," *European Journal of Personality*, vol. 34, no. 5, pp. 613–631, 2020. DOI: 10.1002/per.2257. eprint: https://doi.org/10.1002/per.2257. [Online]. Available: https://doi.org/10.1002/per.2257.

[254] S. Ji, S. Pan, X. Li, E. Cambria, G. Long, and Z. Huang, "Suicidal ideation detection: A review of machine learning methods and applications," *IEEE Transactions on Computational Social Systems*, vol. 8, no. 1, pp. 214–226, 2021. DOI: 10.1109/TCSS.2020.3021467.

[255] S. Marcus, T. David, and A. Predescu, *Empatia şi relaţia profesor-elev*. Editura Academiei Republicii Socialiste Romania, 1987.

[256] W. von Kempelen, H. Füger, and J. Mansfeld, *Wolfgangs von Kempelen k.k. wirklichen Hofraths Mechanismus der menschlichen Sprache: nebst der Beschreibung seiner sprechenden Maschine*. J.V. Degen, 1791. [Online]. Available: https://books.google.ro/books?id=W75CAQAAMAAJ.

[257] H. W. Dudley, *Us 135416a, "system for the artificial production of vocal or other sounds"*, issued 1937-04-07. [Online]. Available: https://patents.google.com/patent/US2121142A/en.

[258] C. H. Shadle and R. I. Damper, "Prospects for articulatory synthesis: A position paper," in *4th ISCA Tutorial and Research Workshop (ITRW) on Speech Synthesis*, 2001.

[259] C. H. Coker, "A model of articulatory dynamics and control," *Proceedings of the IEEE*, vol. 64, pp. 452–460, 1976.

[260] J. Engel, L. Hantrakul, C. Gu, and A. Roberts, *Ddsp: Differentiable digital signal processing*, 2020. DOI: 10.48550/ARXIV.2001.04643. [Online]. Available: https://arxiv.org/abs/2001.04643.

[261] E. Battenberg *et al.*, "Location-relative attention mechanisms for robust long-form speech synthesis," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 6194–6198. DOI: 10.1109/ICASSP40776.2020.9054106.

[262] P. Seeviour, J. Holmes, and M. Judd, "Automatic generation of control signals for a parallel formant speech synthesizer," in *ICASSP'76. IEEE International Conference on Acoustics, Speech, and Signal Processing*, IEEE, vol. 1, 1976, pp. 690–693.

[263] J. Olive, "Rule synthesis of speech from dyadic units," in *ICASSP'77. IEEE International Conference on Acoustics, Speech, and Signal Processing*, IEEE, vol. 2, 1977, pp. 568–570.

[264] E. Moulines and F. Charpentier, "Pitch-synchronous waveform processing techniques for text-to-speech synthesis using diphones," *Speech communication*, vol. 9, no. 5-6, pp. 453–467, 1990.

[265] Y. Sagisaka, N. Kaiki, N. Iwahashi, and K. Mimura, "Atr μ-talk speech synthesis system.," in *ICSLP*, vol. 92, 1992, pp. 483–486.

[266] A. J. Hunt and A. W. Black, "Unit selection in a concatenative speech synthesis system using a large speech database," in *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*, IEEE, vol. 1, 1996, pp. 373–376.

[267] A. Black, P. Taylor, R. Caley, and R. Clark, *The festival speech synthesis system*, 1998.

[268] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Simultaneous modeling of spectrum, pitch and duration in hmm-based speech synthesis," in *Sixth European Conference on Speech Communication and Technology*, 1999.

[269] T. Yoshimura, "Simultaneous modeling of phonetic and prosodic parameters, and characteristic conversion for hmm-based text-to-speech systems," *PhD diss, Nagoya Institute of Technology*, 2002.

[270] A. W. Black, H. Zen, and K. Tokuda, "Statistical parametric speech synthesis," in *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07*, IEEE, vol. 4, 2007, pp. IV–1229.

[271] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura, "Speech parameter generation algorithms for hmm-based speech synthesis," in *2000 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No. 00CH37100)*, IEEE, vol. 3, 2000, pp. 1315–1318.

[272] K. Tokuda, Y. Nankaku, T. Toda, H. Zen, J. Yamagishi, and K. Oura, "Speech synthesis based on hidden markov models," *Proceedings of the IEEE*, vol. 101, no. 5, pp. 1234–1252, 2013.

[273] S. Imai, "Cepstral analysis synthesis on the mel frequency scale," in *ICASSP'83. IEEE International Conference on Acoustics, Speech, and Signal Processing*, IEEE, vol. 8, 1983, pp. 93–96.

[274] S. Imai, K. Sumita, and C. Furuichi, "Mel log spectrum approximation (mlsa) filter for speech synthesis," *Electronics and Communications in Japan (Part I: Communications)*, vol. 66, no. 2, pp. 10–18, 1983.

[275] H. Kawahara, "Straight, exploitation of the other aspect of vocoder: Perceptually isomorphic decomposition of speech sounds," *Acoustical science and technology*, vol. 27, no. 6, pp. 349–353, 2006.

[276] H. Zen, "Acoustic modeling in statistical parametric speech synthesis-from hmm to lstm-rnn," 2015.

[277] H. Zen, "Statistical parametric speech synthesis: From hmm to lstm-rnn," 2015.

[278] Y. Fan, Y. Qian, F.-L. Xie, and F. K. Soong, "Tts synthesis with bidirectional lstm based recurrent neural networks," in *Fifteenth annual conference of the international speech communication association*, 2014.

[279] H. Ze, A. Senior, and M. Schuster, "Statistical parametric speech synthesis using deep neural networks," in *2013 ieee international conference on acoustics, speech and signal processing*, IEEE, 2013, pp. 7962–7966.

[280] Y. Qian, Y. Fan, W. Hu, and F. K. Soong, "On the training aspects of deep neural network (dnn) for parametric tts synthesis," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2014, pp. 3829–3833.

[281] W. Wang, S. Xu, B. Xu, *et al.*, "First step towards end-to-end parametric tts synthesis: Generating spectral parameters with neural attention.," in *Interspeech*, 2016, pp. 2243–2247.

[282] A. v. d. Oord *et al.*, "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[283] A. Gibiansky *et al.*, "Deep voice 2: Multi-speaker neural text-to-speech," *Advances in neural information processing systems*, vol. 30, 2017.

[284] S. Ö. Arık *et al.*, "Deep voice: Real-time neural text-to-speech," in *International Conference on Machine Learning*, PMLR, 2017, pp. 195–204.

[285] D. Stanton, Y. Wang, and R. Skerry-Ryan, "Predicting expressive speaking style from text in end-to-end speech synthesis," in *2018 IEEE Spoken Language Technology Workshop (SLT)*, IEEE, 2018, pp. 595–602.

[286] W. Ping *et al.*, "Deep Voice 3: 2000-Speaker Neural Text-to-Speech," *CoRR*, vol. abs/1710.07654, 2017. arXiv: 1710.07654. [Online]. Available: http://arxiv.org/abs/1710.07654.

[287] Y. Ren *et al.*, "Fastspeech: Fast, robust and controllable text-to-speech," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[288] W. Ping, K. Peng, and J. Chen, "Clarinet: Parallel wave generation in end-to-end text-to-speech," *arXiv preprint arXiv:1807.07281*, 2018.

[289] Y. Ren *et al.*, "Fastspeech 2: Fast and high-quality end-to-end text-to-speech," *arXiv preprint arXiv:2006.04558*, 2020.

[290] J. Donahue, S. Dieleman, M. Bińkowski, E. Elsen, and K. Simonyan, "End-to-end adversarial text-to-speech," *arXiv preprint arXiv:2006.03575*, 2020.

[291] C. Mansfield, M. Sun, Y. Liu, A. Gandhe, and B. Hoffmeister, "Neural text normalization with subword units," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Industry Papers)*, 2019, pp. 190–196.

[292] R. Sproat and N. Jaitly, "Rnn approaches to text normalization: A challenge," *arXiv preprint arXiv:1611.00068*, 2016.

[293] H. Zhang *et al.*, "Neural models of text normalization for speech applications," *Computational Linguistics*, vol. 45, no. 2, pp. 293–337, 2019.

[294] K. Yao and G. Zweig, "Sequence-to-sequence neural net models for grapheme-to-phoneme conversion," *arXiv preprint arXiv:1506.00196*, 2015.

[295] H. Sun *et al.*, "Token-level ensemble distillation for grapheme-to-phoneme conversion," *arXiv preprint arXiv:1904.03446*, 2019.

[296] H. Sun *et al.*, "Knowledge distillation from bert in pre-training and fine-tuning for polyphone disambiguation," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, IEEE, 2019, pp. 168–175.

[297] V. K. R. Sridhar, S. Bangalore, and S. Narayanan, "Exploiting acoustic and syntactic features for prosody labeling in a maximum entropy framework," in *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, 2007, pp. 1–8.

[298] Y. Qian, Z. Wu, X. Ma, and F. Soong, "Automatic prosody prediction and detection with conditional random field (crf) models," in *2010 7th International Symposium on Chinese Spoken Language Processing*, IEEE, 2010, pp. 135–138.

[299] J. H. Jeon and Y. Liu, "Automatic prosodic events detection using syllable-based acoustic and syntactic features," in *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, IEEE, 2009, pp. 4565–4568.

[300] J. Pan *et al.*, "A unified sequence-to-sequence front-end model for mandarin text-to-speech synthesis," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2020, pp. 6689–6693.

[301] Y. Zhang, L. Deng, and Y. Wang, "Unified mandarin tts front-end based on distilled bert model," *arXiv preprint arXiv:2012.15404*, 2020.

[302] H. Li, Y. Kang, and Z. Wang, "Emphasis: An emotional phoneme-based acoustic model for speech synthesis system," *arXiv preprint arXiv:1806.09276*, 2018.

[303] C. Yu *et al.*, "Durian: Duration informed attention network for multimodal synthesis," *arXiv preprint arXiv:1909.01700*, 2019.

[304] H. Tachibana, K. Uenoyama, and S. Aihara, "Efficiently trainable text-to-speech system based on deep convolutional networks with guided attention," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2018, pp. 4784–4788.

[305] Y. Taigman, L. Wolf, A. Polyak, and E. Nachmani, "Voiceloop: Voice fitting and synthesis via a phonological loop," *arXiv preprint arXiv:1707.06588*, 2017.

[306] K. Peng, W. Ping, Z. Song, and K. Zhao, "Non-autoregressive neural text-to-speech," in *International conference on machine learning*, PMLR, 2020, pp. 7586–7598.

[307] J. Kim, S. Kim, J. Kong, and S. Yoon, "Glow-tts: A generative flow for text-to-speech via monotonic alignment search," *Advances in Neural Information Processing Systems*, vol. 33, pp. 8067–8077, 2020.

[308] V. Popov, I. Vovk, V. Gogoryan, T. Sadekova, and M. Kudinov, "Grad-tts: A diffusion probabilistic model for text-to-speech," in *International Conference on Machine Learning*, PMLR, 2021, pp. 8599–8608.

[309] M. Morise, F. Yokomori, and K. Ozawa, "World: A vocoder-based high-quality speech synthesis system for real-time applications," *IEICE TRANSACTIONS on Information and Systems*, vol. 99, no. 7, pp. 1877–1884, 2016.

[310] A. van den Oord *et al.*, "Parallel WaveNet: Fast High-Fidelity Speech Synthesis," Google Deepmind, Tech. Rep., 2017. [Online]. Available: `https://arxiv.org/abs/1711.10433`.

[311] N. Kalchbrenner *et al.*, "Efficient neural audio synthesis," in *International Conference on Machine Learning*, PMLR, 2018, pp. 2410–2419.

[312] M. Bińkowski *et al.*, "High fidelity speech synthesis with adversarial networks," *arXiv preprint arXiv:1909.11646*, 2019.

[313] R. Prenger, R. Valle, and B. Catanzaro, *Waveglow: A flow-based generative network for speech synthesis*, 2018. DOI: `10.48550/ARXIV.1811.00002`. [Online]. Available: `https://arxiv.org/abs/1811.00002`.

[314] S. Kim, S.-g. Lee, J. Song, J. Kim, and S. Yoon, *Flowavenet : A generative flow for raw audio*, 2018. DOI: `10.48550/ARXIV.1811.02155`. [Online]. Available: `https://arxiv.org/abs/1811.02155`.

[315] K. Kumar *et al.*, *Melgan: Generative adversarial networks for conditional waveform synthesis*, 2019. DOI: `10.48550/ARXIV.1910.06711`. [Online]. Available: `https://arxiv.org/abs/1910.06711`.

[316] J.-M. Valin and J. Skoglund, *Lpcnet: Improving neural speech synthesis through linear prediction*, 2018. DOI: 10.48550/ARXIV.1810.11846. [Online]. Available: https://arxiv.org/abs/1810.11846.

[317] N. Chen, Y. Zhang, H. Zen, R. J. Weiss, M. Norouzi, and W. Chan, "Wavegrad: Estimating gradients for waveform generation," *ArXiv*, vol. abs/2009.00713, 2021.

[318] J. Kong, J. Kim, and J. Bae, "Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis," *ArXiv*, vol. abs/2010.05646, 2020.

[319] A. van den Oord *et al.*, "Parallel wavenet: Fast high-fidelity speech synthesis," in *ICML*, 2018.

[320] J. M. R. Sotelo *et al.*, "Char2wav: End-to-end speech synthesis," in *ICLR*, 2017.

[321] Y. Ren *et al.*, "Fastspeech 2: Fast and high-quality end-to-end text-to-speech," *ArXiv*, vol. abs/2006.04558, 2021.

[322] J. Kim, J. Kong, and J. Son, "Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech," *ArXiv*, vol. abs/2106.06103, 2021.

[323] R. J. Weiss, R. J. Skerry-Ryan, E. Battenberg, S. Mariooryad, and D. P. Kingma, "Wave-tacotron: Spectrogram-free end-to-end text-to-speech synthesis," *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5679–5683, 2021.

[324] Z. Byambadorj, R. Nishimura, A. Ayush, K. Ohta, and N. Kitaoka, "Multi-speaker tts system for low-resource language using cross-lingual transfer learning and data augmentation," in *2021 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 2021, pp. 849–853.

[325] J. Xu *et al.*, "Lrspeech: Extremely low-resource speech synthesis and recognition," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '20, Virtual Event, CA, USA: Association for Computing Machinery, 2020, pp. 2802–2812, ISBN: 9781450379984. DOI: 10.1145/3394486.3403331. [Online]. Available: https://doi.org/10.1145/3394486.3403331.

[326] T. Tu, Y.-J. Chen, C.-c. Yeh, and H.-Y. Lee, "End-to-end text-to-speech for low-resource languages by cross-lingual transfer learning," *arXiv preprint arXiv:1904.06508*, 2019.

[327] S. H. Mohammadi and A. Kain, "An overview of voice conversion systems," *Speech Communication*, vol. 88, pp. 65–82, 2017.

[328] T. Toda *et al.*, "The voice conversion challenge 2016.," in *Interspeech*, 2016, pp. 1632–1636.

[329] B. Sisman, J. Yamagishi, S. King, and H. Li, "An overview of voice conversion and its challenges: From statistical modeling to deep learning," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 132–157, 2020.

[330] I. Himawan, S. Aryal, I. Ouyang, S. Kang, P. Lanchantin, and S. King, "Speaker adaptation of a multilingual acoustic model for cross-language synthesis," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2020, pp. 7629–7633.

[331] Z. Meng, J. Li, and Y. Gong, "Adversarial speaker adaptation," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2019, pp. 5721–5725.

[332] Q. Xie *et al.*, "The multi-speaker multi-style voice cloning challenge 2021," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2021, pp. 8613–8617.

[333] Y.-A. Chung, Y. Wang, W.-N. Hsu, Y. Zhang, and R. Skerry-Ryan, "Semi-super-vised training for improving data efficiency in end-to-end speech synthesis," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2019, pp. 6940–6944.

[334] W. Fang, Y.-A. Chung, and J. Glass, "Towards transfer learning for end-to-end speech synthesis from deep pre-trained language models," *arXiv preprint:1906.07307*, 2019.

[335] M. Zhang, X. Wang, F. Fang, H. Li, and J. Yamagishi, "Joint training framework for text-to-speech and voice conversion using multi-source tacotron and wavenet," *arXiv preprint arXiv: 1903.12389*, 2019.

[336] Y. Zhang *et al.*, *Multilingual speech synthesis and cross-language voice cloning*, US Patent A pp. 16/855,042, Dec. 2020.

[337] T. Nekvinda and O. Dušek, "One model, many languages: Meta-learning for multilingual text-to-speech," *arXiv preprint arXiv:2008.00768*, 2020.

[338] T. Kenter, V. Wan, C.-A. Chan, R. Clark, and J. Vit, "Chive: Varying prosody in speech synthesis with a linguistically driven dynamic hierarchical conditional variational network," in *International Conference on Machine Learning*, PMLR, 2019, pp. 3331–3340.

[339] A. Łańcucki, "Fastpitch: Parallel text-to-speech with pitch prediction," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2021, pp. 6588–6592.

[340] M. Morrison, Z. Jin, J. Salamon, N. J. Bryan, and G. J. Mysore, "Controllable Neural Prosody Synthesis," in *Proc. Interspeech 2020*, 2020, pp. 4437–4441. DOI: 10.21437/Interspeech.2020-2918.

[341] S. Ö. Arik, J. Chen, K. Peng, W. Ping, and Y. Zhou, "Neural voice cloning with a few samples," *ArXiv*, vol. abs/1802.06006, 2018.

[342] S. Choi, S. Han, D. Kim, and S. Ha, "Attentron: Few-shot text-to-speech utilizing attention-based variable-length embedding," *ArXiv*, vol. abs/2005.08484, 2020.

[343] L. Chen, Y. Deng, X. Wang, F. K. Soong, and L. He, "Speech bert embedding for improving prosody in neural tts," *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6563–6567, 2021.

[344] Y. Xiao, L. He, H. Ming, and F. K. Soong, "Improving prosody with linguistic and bert derived features in multi-speaker based mandarin chinese neural tts," *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6704–6708, 2020.

[345] Y. Jia *et al.*, "Transfer learning from speaker verification to multispeaker text-to-speech synthesis," in *NeurIPS*, 2018.

[346] S. Ma, D. J. McDuff, and Y. Song, "Neural tts stylization with adversarial and collaborative games," in *ICLR*, 2019.

[347] R. J. Skerry-Ryan *et al.*, "Towards end-to-end prosody transfer for expressive speech synthesis with tacotron," *ArXiv*, vol. abs/1803.09047, 2018.

[348] I. Elias *et al.*, "Parallel tacotron: Non-autoregressive and controllable tts," *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5709–5713, 2021.

[349] K. Akuzawa, Y. Iwasawa, and Y. Matsuo, "Expressive speech synthesis via modeling expressions with variational autoencoder," *ArXiv*, vol. abs/1804.02135, 2018.

[350] W.-N. Hsu *et al.*, "Hierarchical generative modeling for controllable speech synthesis," *ArXiv*, vol. abs/1810.07217, 2019.

[351] W.-N. Hsu *et al.*, "Disentangling correlated speaker and noise for speech synthesis via data augmentation and adversarial factorization," *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5901–5905, 2019.

[352] G. Sun *et al.*, "Generating diverse and natural text-to-speech samples using a quantized fine-grained vae and autoregressive prosody prior," *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6699–6703, 2020.

[353] G. Sun, Y. Zhang, R. J. Weiss, Y. Cao, H. Zen, and Y. Wu, "Fully-hierarchical fine-grained prosody modeling for interpretable speech synthesis," *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6264–6268, 2020.

[354] Y.-J. Zhang, S. Pan, L. He, and Z. Ling, "Learning latent representations for style control and transfer in end-to-end speech synthesis," *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6945–6949, 2019.

[355] M. Jeong, H. Kim, S. J. Cheon, B. J. Choi, and N. S. Kim, "Diff-tts: A denoising diffusion model for text-to-speech," in *Interspeech*, 2021.

[356] J. Kim, S. Kim, J. Kong, and S. Yoon, "Glow-tts: A generative flow for text-to-speech via monotonic alignment search," *ArXiv*, vol. abs/2005.11129, 2020.

[357] S.-H. Lee, H.-W. Yoon, H.-R. Noh, J.-H. Kim, and S.-W. Lee, "Multi-spectrogan: High-diversity and high-fidelity spectrogram generation with adversarial style combination for speech synthesis," in *AAAI*, 2021.

[358] R. Valle, K. J. Shih, R. J. Prenger, and B. Catanzaro, "Flowtron: An autoregressive flow-based generative network for text-to-speech synthesis," *ArXiv*, vol. abs/2005.05957, 2021.

[359] Y. Jia, H. Zen, J. Shen, Y. Zhang, and Y. Wu, "Png bert: Augmented bert on phonemes and graphemes for neural tts," in *Interspeech*, 2021.

[360] T. Hayashi, S. Watanabe, T. Toda, K. Takeda, S. Toshniwal, and K. Livescu, "Pre-trained text embeddings for enhanced text-to-speech synthesis," in *INTERSPEECH*, 2019.

[361] Y. Zhang *et al.*, "Learning to speak fluently in a foreign language: Multilingual speech synthesis and cross-language voice cloning," *arXiv preprint arXiv:1907.04448*, 2019.

[362] M. Chen *et al.*, "Multispeech: Multi-speaker text-to-speech with transformer," *arXiv preprint arXiv:2006.04664*, 2020.

[363] A. Aubin, A. Cervone, O. Watts, and S. King, "Improving speech synthesis with discourse relations.," in *Interspeech*, 2019, pp. 4470–4474.

[364] X. Wang, H. Ming, L. He, and F. K. Soong, "S-transformer: Segment-transformer for robust neural speech synthesis," *arXiv preprint arXiv:2011.08480*, 2020.

[365] R. Skerry-Ryan *et al.*, "Towards end-to-end prosody transfer for expressive speech synthesis with tacotron," in *international conference on machine learning*, PMLR, 2018, pp. 4693–4702.

[366] R. Liu, B. Sisman, G. Gao, and H. Li, "Expressive tts training with frame and style reconstruction loss," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 1806–1818, 2021.

[367] M. Chen *et al.*, "Adaspeech: Adaptive text-to-speech for custom voice," *arXiv preprint arXiv:2103.00993*, 2021.

[368] C.-M. Chien and H.-y. Lee, "Hierarchical prosody modeling for non-autoregressive speech synthesis," in *2021 IEEE Spoken Language Technology Workshop (SLT)*, IEEE, 2021, pp. 446–453.

[369] Y. Hono *et al.*, "Hierarchical multi-grained generative model for expressive speech synthesis," *arXiv preprint arXiv:2009.08474*, 2020.

[370] G. Sun, Y. Zhang, R. J. Weiss, Y. Cao, H. Zen, and Y. Wu, "Fully-hierarchical fine-grained prosody modeling for interpretable speech synthesis," in *ICASSP 2020-2020 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, IEEE, 2020, pp. 6264–6268.

[371] A. Talman, A. Suni, H. Celikkanat, S. Kakouros, J. Tiedemann, and M. Vainio, "Predicting prosodic prominence from text with pre-trained contextualized word representations," *arXiv preprint arXiv:1908.02262*, 2019.

[372] G. Sun *et al.*, "Generating diverse and natural text-to-speech samples using a quantized fine-grained vae and autoregressive prosody prior," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2020, pp. 6699–6703.

[373] Y. Lei, S. Yang, and L. Xie, "Fine-grained emotion strength transfer, control and prediction for emotional speech synthesis," in *2021 IEEE Spoken Language Technology Workshop (SLT)*, IEEE, 2021, pp. 423–430.

[374] Z. Zeng, J. Wang, N. Cheng, and J. Xiao, "Prosody learning mechanism for speech synthesis system without text length limit," *arXiv preprint arXiv:2008.05656*, 2020.

[375] C. Zhang *et al.*, "Denoispeech: Denoising text-to-speech with frame-level noise modeling," *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7063–7067, 2021.

[376] D.-R. Liu, C.-Y. Yang, S.-L. Wu, and H.-y. Lee, "Improving unsupervised style transfer in end-to-end speech synthesis with end-to-end speech recognition," *2018 IEEE Spoken Language Technology Workshop (SLT)*, pp. 640–647, 2018.

[377] T. Li, S. Yang, L. Xue, and L. Xie, "Controllable emotion transfer for end-to-end speech synthesis," *2021 12th International Symposium on Chinese Spoken Language Processing (ISCSLP)*, pp. 1–5, 2021.

[378] M. Whitehill, S. Ma, D. J. McDuff, and Y. Song, "Multi-reference neural tts stylization with adversarial cycle consistency," in *INTERSPEECH*, 2020.

[379] I. S. on Subjective Methods, "Ieee recommended practice for speech quality measurements," *IEEE Transactions on Audio and Electroacoustics*, vol. 17, pp. 225–246, 1969.

[380] International Telecommunication Union - Telecommunication Sector, *Recommendation, Subjective performance assessment of telephone band and wideband digital codecs, note = p.830 (1998)*.

[381] R. Viswanathan, J. Makhoul, and W. Russell, "Towards perceptually consistent measures of spectral distance," in *ICASSP*, 1976.

[382] P. Wagner *et al.*, "Speech synthesis evaluation—state-of-the-art assessment and suggestion for a novel research program," in *Proceedings of the 10th Speech Synthesis Workshop (SSW10)*, 2019.

[383] C. Valentini-Botinhao, J. Yamagishi, and S. King, "Evaluation of objective measures for intelligibility prediction of hmm-based synthetic speech in noise," in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2011, pp. 5112–5115. DOI: 10.1109/ICASSP.2011.5947507.

[384] K. Kokkinakis and P. C. Loizou, "Evaluation of objective measures for quality assessment of reverberant speech," in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2011, pp. 2420–2423. DOI: 10.1109/ICASSP.2011.5946972.

[385] J. Lorenzo-Trueba, R. Barra-Chicote, R. San-Segundo, J. Ferreiros, J. Yamagishi, and J. M. Montero, "Emotion transplantation through adaptation in HMM-based speech synthesis," *Computer Speech and Language*, vol. 34, no. 1, pp. 292–307, 2015, ISSN: 0885-2308. DOI: https://doi.org/10.1016/j.csl.2015.03.008.

[386] G. E. Henter, X. Wang, and J. Yamagishi, "Deep encoder-decoder models for unsupervised learning of controllable speech synthesis," *ArXiv*, vol. abs/1807.11470, 2018.

[387] W.-N. Hsu *et al.*, "Hierarchical Generative Modeling for Controllable Speech Synthesis," in *arXiv*, 2018. [Online]. Available: https://arxiv.org/abs/1810.07217.

[388] J. Parker, Y. Stylianou, and R. Cipolla, "Adaptation of an expressive single speaker deep neural network speech synthesis system," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2018, pp. 5309–5313.

[389] 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, ISSN: 0167-6393. DOI: 10.1016/j.specom.2010.12.002.

[390] A. Stan *et al.*, "The swara speech corpus: A large parallel romanian read speech dataset," in *2017 International Conference on Speech Technology and Human-Computer Dialogue (SpeD)*, IEEE, 2017, pp. 1–6.

[391] K. Tokuda, H. Zen, and A. Black, "An HMM-Based Speech Synthesis System Applied To English," in *Proc. of SSW*, Oct. 2002, pp. 227–230, ISBN: 0-7803-7395-2. DOI: 10.1109/WSS.2002.1224415.

[392] J. Shen *et al.*, *Natural tts synthesis by conditioning wavenet on mel spectrogram predictions*, 2018. arXiv: 1712.05884 [cs.CL].

[393] S. D. Aäron van den Oord *et al.*, "WaveNet: A Generative Model for Raw Audio," in *arXiv*, 2016. [Online]. Available: https://arxiv.org/abs/1609.03499.

[394] S. Ö. Arik *et al.*, "Deep Voice: Real-time Neural Text-to-Speech," *CoRR*, 2017. arXiv: 1702.07825. [Online]. Available: http://arxiv.org/abs/1702.07825.

[395] S. Ö. Arik *et al.*, "Deep Voice 2: Multi-Speaker Neural Text-to-Speech," *CoRR*, vol. abs/1705.08947, 2017. arXiv: 1705.08947. [Online]. Available: http://arxiv.org/abs/1705.08947.

[396] S. Mehri *et al.*, "SampleRNN: An Unconditional End-to-End Neural Audio Generation Model," *CoRR*, vol. abs/1612.07837, 2016. arXiv: 1612.07837. [Online]. Available: http://arxiv.org/abs/1612.07837.

[397] J. Sotelo *et al.*, "Char2Wav: End-to-end speech synthesis," in *International Conference on Learning Representations (Workshop Track)*, Apr. 2017.

[398] Y. Fan, Y. Qian, F. Soong, and L. He, "Speaker and language factorization in DNN-based TTS synthesis," in *Proc. of ICASSP*, Mar. 2016, pp. 5540–5544. DOI: 10.1109/ICASSP.2016.7472737.

[399] 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. arXiv: 1802.06006. [Online]. Available: http://arxiv.org/abs/1802.06006.

[400] Z. Huang, H. Lu, M. Lei, and Z. Yan, "Linear networks based speaker adaptation for speech synthesis," *arXiv e-prints*, Mar. 2018. arXiv: 1803.02445.

[401] B. Li and H. Zen, "Multi-Language Multi-Speaker Acoustic Modeling for LSTM-RNN based Statistical Parametric Speech Synthesis," in *Proc. of Interspeech*, 2016.

[402] 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, ISSN: 1088-467X.

[403] A. Pratama, R. I. Alhaqq, and Y. Ruldeviyani, "Sentiment analysis of the covid-19 booster vaccination program as a requirement for homecoming during eid fitr in indonesia," *Journal of Theoretical and Applied Information Technology*, vol. 101, no. 1, 2023.

[404] K. Mouyassir, A. Fathi, and N. Assad, "Elevating aspect-based sentiment analysis in the moroccan cosmetics industry with transformer-based models.," *International Journal of Advanced Computer Science & Applications*, vol. 15, no. 6, 2024.

[405] L. J. Yoon, X. Y. Tan, K. Y. Lim, C. W. Tan, L. E. Cheng, and J. Tan, "A comparative study of lemmatization approaches for rojak language," in *The International Conference on Data Science and Emerging Technologies*, Springer, 2023, pp. 3–16.

[406] M. Kawtar, A. Fathi, N. Assad, and A. Kartit, "Hierarchical spatiotemporal aspect-based sentiment analysis for chain restaurants using machine learning.," *International Journal of Advanced Computer Science & Applications*, vol. 15, no. 3, 2024.

[407] D. Ungureanu, M. Badeanu, G.-C. Marica, M. Dascalu, and D. I. Tufis, "Establishing a baseline of romanian speech-to-text models," in *2021 International Conference on Speech Technology and Human-Computer Dialogue (SpeD)*, IEEE, 2021, pp. 132–138.

[408] B. Lőrincz, E. Irimia, A. Stan, and V. B. Mititelu, "Rolex: The development of an extended romanian lexical dataset and its evaluation at predicting concurrent lexical information," *Natural Language Engineering*, pp. 1–26, 2022.

[409] C.-L. Gasan and V. PĂIȘ, "Investigation of romanian speech recognition improvement by incorporating italian speech data," *LINGUISTIC RESOURCES AND TOOLS FOR NATURAL LANGUAGE PROCESSING*, p. 235, 2023.

[410] A. Stan and J. O'Mahony, "An analysis on the effects of speaker embedding choice in non auto-regressive tts," *arXiv preprint arXiv:2307.09898*, 2023.

[411] E. Eren and C. Demiroglu, "Deep learning-based speaker-adaptive postfiltering with limited adaptation data for embedded text-to-speech synthesis systems," *Computer Speech and Language*, p. 101 520, 2023.

[412] B. Lorincz, "Contributions to neural speech synthesis using limited data enhanced with lexical features," in *Proc. 2021 ISCA Symposium on Security and Privacy in Speech Communication*, 2021, pp. 83–85.

[413] A. F. Khan *et al.*, "When linguistics meets web technologies. recent advances in modelling linguistic linked open data,"

[414] L. Stankevičius, M. Lukoševičius, J. Kapočiūtė-Dzikienė, M. Briedienė, and T. Krilavičius, "Correcting diacritics and typos with a byt5 transformer model," *Applied Sciences*, vol. 12, no. 5, p. 2636, 2022.

[415] L. Pakalniškis, "Giliuoju mokymusi grįstas diakritinių ženklų atstatymas lietuvių kalbai," Ph.D. dissertation, Kauno technologijos universitetas, 2022.

[416] A. Stan and B. Lőrincz, "Generating the voice of the interactive virtual assistant," in *Virtual Assistant*, IntechOpen, 2021.

[417] Y. Hifny, "Recent advances in arabic syntactic diacritics restoration," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2021, pp. 7768–7772.

[418] J. Náplava, M. Straka, and J. Straková, "Diacritics restoration using bert with analysis on czech language," *arXiv preprint arXiv:2105.11408*, 2021.

[419] S. Esmail, K. Bar, and N. Dershowitz, "How much does lookahead matter for disambiguation? partial arabic diacritization case study," 2021.

[420] K. M. Scott, S. Ashby, and R. Cibin, "Implementing text-to-speech tools for community radio in remote regions of romania," in *Adjunct Proceedings of the 2020 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2020 ACM International Symposium on Wearable Computers*, 2020, pp. 123–126.

[421] A. Al-Thubaity, A. Alkhalifa, A. Almuhareb, and W. Alsanie, "Arabic diacritization using bidirectional long short-term memory neural networks with conditional random fields," *IEEE Access*, vol. 8, pp. 154 984–154 996, 2020.

[422] F. IORDACHE, L. GEORGESCU, D. ONEAȚĂ, and H. CUCU, "Romanian automatic diacritics restoration challenge," in *Proceedings of the 14th international conference "linguistic resources and tools for natural language processing*, 2019, pp. 64–74.

[423] K. L. O. Ogheneruemu, "Development of yoruba diacritic restoration for under dot and diacritic mark for yoruba text using deep learning model," M.S. thesis, Kwara State University (Nigeria), 2022.

[424] A. T. Özge, Ö. Bozal, and U. Özge, "Diacritics correction in turkish with context-aware sequence to sequence modeling," *Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 30, no. 6, pp. 2433–2445, 2022.

[425] S. Esmail, K. Bar, and N. Dershowitz, "How much does lookahead matter for disambiguation? partial arabic diacritization case study," *Computational Linguistics*, vol. 48, no. 4, pp. 1103–1123, 2022.

[426] S. Sardarov, "Development and design of deep learning-based parts-of-speech tagging system for azerbaijani language," Ph.D. dissertation, Khazar University,Azerbaijan, 2022.

[427] J. Juričić, "Označavanje vrsta riječi pomoću neuronskih mreža," Ph.D. dissertation, University of Split. Faculty of Science. Department of Informatics, 2022.

[428] S. A. Harjanto and A. Romadhony, "Question template extraction using sequence labeling approach," in *2024 International Conference on Data Science and Its Applications (ICoDSA)*, IEEE, 2024, pp. 242–247.

[429] F. Aydinov, I. Huseynov, S. Sayadzada, and S. Rustamov, "Investigation of automatic part-of-speech tagging using crf, hmm and lstm on misspelled and edited texts," in *Proceedings of the 2022 5th artificial intelligence and cloud computing conference*, 2022, pp. 21–28.

[430] J. Juričić and B. Žitko, "Pos-only tagging using rnn for croatian language," in *International Conference on Digital Transformation in Education and Artificial Intelligence Application*, Springer, 2023, pp. 45–62.

[431] A. Gupta and H. Fatima, "Topic modeling in healthcare: A survey study," *NEUROQUANTOLOGY*, vol. 20, no. 11, pp. 6214–6221, 2022.

[432] J. Kenei, E. Opiyo, and J. Machii, "Modeling and visualization of clinical texts to enhance meaningful and user-friendly information re-trieval," in *Med. Sci. Forum*, The 2nd International Electronic Conference on Healthcare, vol. 1, 2022.