BULLETIN OF THE POLISH ACADEMY OF SCIENCES
TECHNICAL SCIENCES, Vol. 72(4), 2024, Article number: e149818
DOI: 10.24425/bpasts.2024.149818

A system dedicated to Polish automatic speech recognition – overview of solutions

Karolina PONDEL-SYCZ* and **Piotr BILSKI**

The Faculty of Electronics and Information Technology on Warsaw University of Technology, Nowowiejska 15/19 Av., 00-665 Warsaw, Poland

Abstract. The paper presents the analysis of modern Artificial Intelligence algorithms for the automated system supporting human beings during their conversation in Polish language. Their task is to perform Automatic Speech Recognition (ASR) and process it further, for instance fill the computer-based form or perform the Natural Language Processing (NLP) to assign the conversation to one of predefined categories. The state-of-the-art review is required to select the optimal set of tools to process speech in the difficult conditions, which degrade accuracy of ASR. The paper presents the top-level architecture of the system applicable for the task. Characteristics of Polish language are discussed. Next, existing ASR solutions and architectures with the End-To-End (E2E) Deep Neural Network (DNN) based ASR models are presented in detail. Differences between Recurrent Neural Networks (RNN), Convolutional Neural Networks (CNN) and transformers in the context of ASR technology are also discussed.

Keywords: automatic speech recognition; deep neural networks; transformer; conformer.

1. INTRODUCTION

The Automatic Speech Recognition (ASR) is the scientific domain of quickly growing importance. Multiple practical applications (such as virtual assistants) show vast area of opportunities to accelerate and facilitate daily operations. The key operations consist in detecting the speech in the sound stream and identifying subsequent words and phrases inside. Later, the detected tokens can be extracted and passed, for instance to the text document (functionalities already available in the office suite such as MS Office). In many cases the context of words and phrases is important, requiring the Natural Language Processing (NLP) techniques. The ASR domain is challenging as the obtained results strongly depend on the analyzed language. The most widely exploited is English due to its worldwide popularity and structural simplicity. Each language requires a separate approach (e.g. considering flexion). The Polish language is challenging due to its complexity, irregular syntax, multiple homonyms and non-standard characters (adding to the difficulty in recognizing words). The additional problem is the ASR performed in difficult conditions, such as the noisy environment. From multiple telecommunication and signal processing domains [1] it is known that the background noise strongly influences the ability to recognize the particular words. Therefore it is important to evaluate capabilities of the existing methods to recognize Polish language in such conditions, which include external sources of disturbances and characteristics of the transmission channel (recognized as the non-flat spectrum bandpass filter) – see [2]. It affects the quality of the signal and can obstruct or even block further language processing. To tackle the presented problems, the Artificial Intelligence (AI) may be applied, with the focus on the Deep Neural Networks (DNN). They have proven their efficiency in multiple applications, especially complex and structured data processing. Currently there are many feed-forward and recurrent architectures (such as CNN or LSTM) applicable for the task. New solutions (i.e. transformers) also emerged recently. Their efficiency and computational complexity must be compared to determine their usage in the embedded applications. The following paper presents an analysis of available ASR solutions for processing the Polish language. The aim of the research was to identify the tools the most suitable for the speech analysis in difficult conditions. The selected algorithms will be compared and implemented in various scenarios. One of their applications can be the ASR component of automated system for the Medicine Doctor support during the patient's interview (Fig. 3). The structure of the paper is as follows. In Section 2, the problem to solve and the proposed system are presented. In Section 3, difficulties and challenges regarding Polish language processing as well as a brief leading to conventional ASR systems are described. Section 4 describes data sources applicable in the project. Section 5 describes End-to-End DNN (E2E) approach as well as the types of neural networks used in E2E. In Section 6, we described selected E2E models we plan to test in the context of Polish language recognition and usability for the problem posed in Section 2. Conclusions are in Section 7.

2. POLISH ASR FOR AUTOMATED ANALYSIS OF DOCTOR-PATIENT CONVERSATIONS

The research problem under consideration is the processing of speech during a conversation in Polish conducted under difficult environmental conditions. These include external sources

*e-mail: karolina.sycz@pw.edu.pl

Manuscript submitted 2023-08-29, revised 2024-02-07, initially accepted for publication 2024-03-11, published in July 2024.

of interference (such as ambient noise), but also the telecommunications channel, which is the medium of speech transmission. Recent crises (including the COVID-19 pandemic) confirm multiple scenarios where this is the case. One of the possible applications of such a system may be the ASR during the medical interview between the doctor and the patient. It should employ technology of an automatic conversation transcript for a healthcare professional. The resulting system purpose would be to record the conversation, extract particular keywords and use them to deliver additional functionalities, such as automated filling the medical forms or suggesting the diagnosis and further code of conduct. The crucial component of such a system will be the data set, which must be specifically prepared for the planned scenarios. The inherent part of the deep learning-based system is the set of recordings large enough to successfully train the model. The architecture of a system for such an application is shown in Fig. 3. There are three key components: the **ASR**, the **NLP** and the **post-processing module**. The speech signal extracted from the conversation is fed into the input of the ASR model whose first stage is the extraction of acoustic features. On their basis the model decodes the phonemes contained in the speech stream and assigns corresponding text representations to them. The output of ASR is a text (speech transcription). The recognized text is therefore fed to the input of the NLP model, whose task is to extract keywords necessary for further processing (like filling the computer-based form). The presented system should work autonomously with the maximum possible accuracy. This calls for the AI methods implementation, which are currently widely used in most data processing applications. Numerous examples of the algorithm implementation for ASR exist in the literature. Since the 1980s, Hidden Markov Models (HMM [3]), conventional models based on the acoustic analysis, language and lexicon structures (Fig. 1) and then hybrid systems have been developed [4]. The research on DNN [5] led to a shift from architectures based on feature extraction and pattern analysis to the E2E (Fig. 2) approach [6]. Their advantage is the ability to operate on the raw information (as 'the data speaks for itself'). This facilitates implementation of language-independent systems (based on multilingual data). The technology requires a large amount of data to produce satisfactory recognition results and huge computing power during the training in the GPGPU framework. The domain is mature enough to spawn not only research projects, but also commercial applications, e.g. Google STT [7].

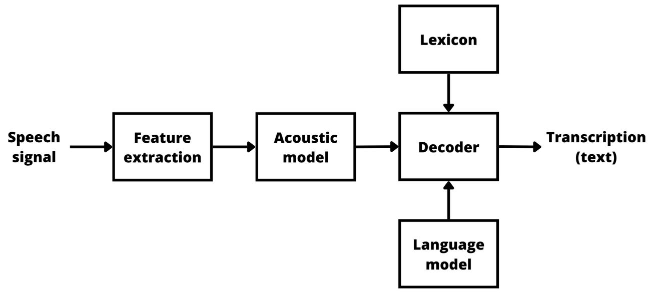

Fig. 1. A simplified diagram of a conventional ASR system consisting of three, separate models – acoustic, language and pronunciation. Training of the models is performed separately and requires forced audio and transcript alignment [8]

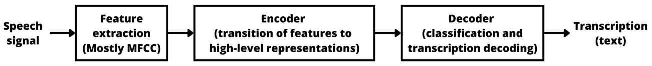
Speech signal
Feature extraction (Mostly MFCC)
Encoder (transition of features to high-level representations)
Decoder (classification and transcription decoding)
Transcription (text)

Fig. 2. A simplified diagram of the ASR system in the E2E approach, in which acoustic, language and pronunciation models are implemented by an integrated deep model that requires soft audio and transcript alignment [9]

The ASR and NLP models in Fig. 3 are based on the E2E architecture, in which 'raw' data are directly fed to the input of the network. The ASR covers encoding by mapping the input speech sequence to a sequence of features, matching the latter to the language and decoding the final classification results. As all recognition stages are integrated into a single network, it is often difficult to determine which part performs the particular subtask. The DNN directly maps acoustic signals to labels without any intermediate states. Development of the individual approach is driven by the specificity of Polish language. The system should be also able to retrain on newly delivered and publicly available data, which imposes anonymization of the voice recordings eliminating the threat of identifying speakers. This requires constructing the problem-specific data sets for training and testing the implemented algorithms. The proposed methodology includes using the generally available data sets (see Section 4) first, and then supplementing them with the application-oriented sets. Construction of the system consists in a sequence of steps:

a. Fine-tuning of the selected DNN models initially trained on the more general sets of recordings.
b. Examining the efficiency, training and adaptation to new tasks of the Polish language ASR models and systems outlined in the paper.
c. Based on the resulting transcription, training the NLP models to search for keywords (symptoms).
d. Implementing a method of automatically completing the medical form based on the information filtered from the transcript.

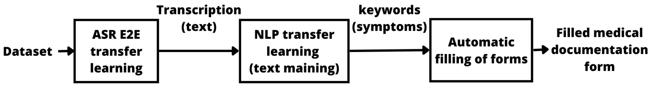
Dataset
ASR E2E transfer learning
Transcription (text)
NLP transfer learning (text maining)
keywords (symptoms)
Automatic filling of forms
Filled medical documentation form

Fig. 3. Proposed architecture for a medical interview support system, consisting of 3 modules: ASR, NLP and medical record filling. Each module is a DNN model

The comparison of algorithms applicable in the system is based on the Word Error Rate (WER). It is used widely in literature and allows to assess the efficiency of ASR and to compare results of different researchers. WER is the ratio of incorrectly recognized words to the total number of words in the transcription. So, the lower the WER, the more efficient ASR and vice versa:

$$WER = \frac{I+D+S}{N} \times 100, \tag{1}$$

where all symbols denote the number of words in the transcription, respectively: S – substitutions (incorrect words), D –

deletions (removed words), I – insertions (added words) and N – number of words in the reference transcription. The subsequent sections present the state-of-the-art in the ASR domain and justify the selected algorithms for the project.

3. CHARACTERISTICS OF THE POLISH LANGUAGE

Currently, the largest number of ARS solutions based on DNN and E2E architecture is for English and Chinese [10]. These languages are highly supported by a significant amount of data to train AI models. As there are approximately 7 000 different languages spoken by people worldwide [11], the need to build ASR systems dedicated to them (or multilingual) is substantial. The presented research concerns the Polish language, which is complex due to its structure, grammar and spelling. The following section presents peculiarities of the language (especially opposed to English), justifying the particular architecture of the ASR system.

3.1. ASR for Polish language

The problem of the speech recognition can be thought of as an attempt to transcribe an acoustic signal into distinct words. This task is divided into three subtasks of modeling:

1. Acoustic speech patterns (predicting which phoneme is uttered in a particular segment of the speech signal).
2. Language statistics (predicting the most probable word sequences).
3. Pronunciation (different variants for each word).

In a conventional ASR system (Fig. 1), these are performed by **acoustic (AM)**, **language (LM)** and **pronunciation (PM)** models [8] which are trained and optimized separately and exchange information. The conventional ASR systems are most often built on the basis of HMM, ANN or mixture of different approaches (e.g. GMM-HMM, HMM-ANN [12], etc.). To train an AM based on HMM, pairs 'acoustic feature – phoneme (label)' are required. This demands the use of forced alignment (assignment of corresponding graphical representations to each segment of audio data). Preparation of the training data and training process is time-consuming and requires an immense amount of work. ML is commonly based on statistical models (e.g. n-grams, see Section 3.3), because most ASR solutions were originally developed for a positional language (English) and later adapted to other ones (including Polish with almost arbitrary sentence formation). Conventional ASR systems are differentiated by the acoustic features used, (e.g. formant frequencies or Linear Prediction Coefficients (LPC) and others), when E2E mainly uses spectrograms or/and MFCC (Mel Frequency Cepstrum Coefficient – filter bank analysis and mel scale simulate nonlinear frequency recognition across the audio spectrum by the human ear [13]). Table 1 shows WERs for sample conventional Polish ASR systems. HMM-based AM achieves high efficiency for short commands, but low for continuous speech. Enhancing the system with DNN significantly improved recognition of Polish continuous speech, so the usage of DNN for Polish speech recognition is effective.

In the considered support system (Fig. 3) all these stages are performed by a single DNN. The NLP does not act as a language component of ASR, but is a separate model processing the ASR output sequence to perform data mining.

Table 1

WER results for the Convenional Polish ASR based on literature sources. All ASR systems in the table have HMM-based AM, LM based on n-grams (except Skrybot, whose authors only report that LM is based on statistical methods) and were created using datasets containing recordings and transcriptions in Polish. The AM of the ARM-1 NG system was enhanced with DNN. CGI stands for Computer Game Interface ASR system

Ref.	System	Speech	WER	Date
[14]	Social robot	short commands	3.9%	2016
[15]	CGI	short commands	0.7%	2021
[16]	Skrybot	continuous speech	27.2%	2021
[17]	ARM-1	continuous speech	26.4%	2016
[18]	ARM-1 NG	continuous speech	4.84%	2021

3.2. Acoustic modeling

In [19], Power Spectral Density (PSD) curves for Polish and American English speech are compared – despite significant differences between them, both curves have similar shapes and contain several maxima and minima (in some cases occurring in different frequency regions). The first maximum, at approximately the same frequency for both languages, reflects the influence of the fundamental frequency on the speech spectrum, while the remaining maxima represent the influence of vowel formants. They occur at separate frequencies for both languages, which is due to different phoneme systems and relative frequencies for phonemes. In [20] fundamental frequency of the laryngeal tone (F0) levels for Polish and American male speakers are compared. Small differences in fundamental speaking frequency (FSF) were found between these groups. The long-term spectral characteristics of Polish and English are similar in terms of PSD levels and statistical distributions with Polish having a fixed lexical accent location (on the penultimate syllable). Therefore inclusion of accented vowel models in Polish ASR allows for a reduction in verbal errors [21]. The set of phonemes present within the Polish language is not obvious, and researchers adopt two SAMPA conventions [22]. Both distinguish 37 phonemes for Polish, but the former accepts the existence of the phonemes 'ą' and 'ę' (written as 'o~' and 'e~'), while the latter questions their distinction, including additional phonemes 'ki' and 'gi'. To represent the pronunciation of words from the phonetic dictionary as accurately as possible and to train the AM optimally, in [14] it was decided to combine these concepts, obtaining a set of 39 phonemes used for the words' representation. According to [15], most phonemes (plosives, fricatives and affricates) occur in soundless-sonorous pairs. However, under certain circumstances, some of them become voiceless (so-called devoicing) and vice versa: voiceless phonemes become sonorous. In the prosody of Polish, the melody of a word (pitch contour) does not affect its meaning. The melody of a sentence may carry semantic information (e.g. a question, an emotion).

3.3. Language modeling

There are key differences in LM structure for English and Polish. Slavic languages are characterized by rich morphology – nouns, pronouns, adjectives, counts and verbs conjugate depending on the grammatical context. Inflectional forms are formed from lemmas using prefixes, suffixes and/or core changes. This results in large dictionaries with hundreds of thousands of entries. Sometimes suffixes differ by only one phoneme, so many forms of words sound very similar. These languages also differ in the level of gender lexicalization. In Polish, feminine nominal forms are common, while in English most nouns have no gender designation. In Polish there are 3 types in the singular and 2 in the plural [15, 23]. There are many other marked forms, such as diminutives and augmentatives, which are rare or absent in English [24]. Words have many forms, so the number of tokens is greater than in English. Polish is characterized by a relatively high morphological richness – there are four million inflectional forms out of about 180 000 basic ones [25]. A larger number of tokens results in rare data, where it is not possible to collect a corpus allowing for the calculation of probabilities for every sequence of a given length that may occur. There are sequences for which there are no probability estimates [26]. The number of permitted sentence forms is different: in English, sentences are usually constructed according to the Subject-Verb-Object (SVO) formation, while in Polish it is relatively unlimited and has no significant impact on the meaning of the sentence, which is resolved around compound inflection. The function of a word (e.g., whether a noun is a subject or a complement) is determined by its form, not by its position in the sentence. Polish sentences often lack a subject – it can be inferred from the form of the verb, and there are no partitions preceding nouns or other parts of speech [15, 25, 27]. A common approach is to use n-grams in LM [28], with the effectiveness of a given n-gram depending on the language. N-grams follow positional logic in English, but are less effective in inflectional languages such as Polish. N-grams are sequences of n words into which an entire utterance is divided. The technique is based on predicting subsequent words based on previous words and discovering the meaning of an utterance based on the local context. A given word is analyzed by taking into account neighboring words (e.g. in most commonly used three-grams, the meaning of the middle word is recognized on the basis of the single word preceding and following it). In Polish, due to its free sentence formation and complex inflection, knowledge of both local and global context is essential for recognizing meaning.

3.4. Pronunciation modeling

Polish, like other Slavic languages, has a simple relationship between orthography and pronunciation. With the help of basic rules, a grapheme-phoneme conversion can be made. For abbreviations, a spelling-letter converter can be used. Care should be taken regarding numerals and loanwords [29]. The rule of simple relation between orthography and pronunciation applies to vowels, but not to all consonants. Vowels are represented by a phoneme with an identical symbol. For consonants followed by the letter ‘i’ and a vowel, a softening symbol can be introduced for the corresponding vowel [16]. There are also consonants composed of more than one letter, which in pronunciation are a single voice like: ‘cz’, ‘sz’, ‘dż’, ‘dź’, as well as sounds that have two spellings, like: ‘ż/rz’, ‘h/ch’, ‘u/ó’. The exception is ‘ć/ci’, where the pronunciation depends on spelling. A significant difference between Polish and English is that the latter has a large number of homophones and many combinations of different words have similar pronunciation, while Polish has a much smaller number of homophones. In English, an unstressed vowel is usually pronounced as ’ɜ’ , ’ɪ’ (phonemes with similar sounds and spectrum) or ‘ə’, so unaccented vowels are almost indistinguishable [30].

4. DATASETS

Slavic languages are spoken by around 320 million people, mainly in central, eastern and southern Europe. The largest language is Russian (~160 million speakers), The next one being Polish (~50 million speakers) [31]. Despite this, the number of resources available to prepare ASR systems is still limited [27], so Polish is classified as a low-resource language [32]. There are both speech corpora containing only Polish e.g.: CORPORA [33] and multilingual dataset, including Polish, designed to create and train E2E ASR systems. Since the presented research is based on the E2E architecture, it was decided to use two multilingual open-source dataset: Multilingual LibriSpeech (MLS) [34] and Mozilla Common Voice (MCV) [35]. Also, the application-oriented dataset must be prepared.

4.1. Mozilla Common Voice – MCV [35]

MCV contains MP3 recordings of speech and corresponding text files for 112 languages, including Polish. Metadata includes age, gender and accent. The Polish part of MCV is constantly being expanded and currently contains 173 hours of speech recordings (in version 14.0 – 163 hours validated, 3208 voices). In MCV, the dataset for each language is divided into a training (train), development (dev) and test set [36].

4.2. Multilingual LibriSpeech – MLS [34]

It is a multilingual version of the LibriSpeech [37] dataset, originally developed only for English. The collection covers read speech using publicly available LibriVox audiobooks and Project Gutenberg text data. It contains 44 500 hours of English and a total of 6000 hours in 7 other languages, including Polish. The dataset is decomposed into training, development and testing parts. In the Polish set, the recordings are in subsets of 103.65, 2.08 and 2.14 hours, respectively (female and male voices).

4.3. Application-specific datasets

As the publicly available datasets do not consider all the requirements of the planned research, there is the need to prepare the individual collection of conversations. The requirements include specific topics of conversations (by different speakers) and various sound degradation sources (e.g. the inclusion of background noises, the use of different recording devices, etc.).

The data set prepared for the mentioned medical interview support system should contain recordings of conversations between a patient and a doctor where specific vocabulary is used (e.g. names of drugs, symptoms of illness) and are aimed at various groups of diseases. The interviews' quality should be intentionally degraded to create the playground for the applied system.

The dataset created for the proposed system requires constant extensions to include changing environmental conditions. So far, 17 scenarios are considered, assigned to variety of speakers (14 female and 15 male initially selected). The scenarios must be repeated in different configurations of the patient-doctor voice, in terms of gender. Recordings are made in various acoustic conditions, including: an acoustic test chamber characterized by a short reverberation time, an office, and a doctor-like room (medium reverberation time). In medical applications, it is crucial for speakers to have their mouths covered with a mask, which is also covered in part of the recordings. Recordings include various bit resolutions (16 and 24 bits per sample) and sampling rates (44.1kHz, 48kHz and 65.5kHz). Different positions of the recording equipment (orientations relative to the doctor's and the patient's heads) must also be taken into account, sound recording devices of varying quality (different microphones varying in price, dynamics, frequency response and characteristics and recorders). The scenarios are divided into doctor-patient utterances and matched with the corresponding recording.

Due to the fact that all the text is initially matched to the entire recording, semi-automatic annotation of the recordings must be performed on the utterances of a particular speaker (doctor or patient). For this purpose, the recordings are initially segmented using a speaker diarization method (separating the audio signal into segments based on who is speaking at a given time) based on voice activity detection – Oracle-VAD diarization and a deep neural model available in the open-source NeMo Toolkit. This allows for the determination of speaker embedded in the recording and speaker label timestamps. The output of the model results in Rich Transcription Time Marked (RTTMS) files containing details about recording. Considering that several E2E ASR models are capable of handling recordings of limited duration (e.g. Whisper up to 30 seconds), recordings and reference transcriptions details about divided into fragments according to the timestamp values of the speaker labels. Finally, the dataset will be based on pairs: recording and corresponding transcription. At this stage, the text is not further normalized, and a future normalization must be adapted to specific E2E ASR models.

5. END-TO-END ASR

In the solution proposed in Section 2, we decided to use the E2E architecture due to its simpler training procedure, lower data requirements and higher recognition efficiency for Polish continuous speech (WERs for E2E solutions tested for Polish are in Table 2). This section covers the E2E architecture and its variations. Here all recognition steps are performed by a single network, trained for the 'general' task of recognizing words. This calls for a global optimization for the network training. E2E models use soft alignment: each audio frame corresponds to all possible states with a defined probability distribution, which does not require the forced alignment [6]. ASR E2E is based on three types of DNNs: RNN, CNN and transformers. A popular solution used in E2E ASR is the Sequence-to-Sequence (Seq2Seq [38], Fig. 2), in which one sequence (speech signal features) is transformed into another one (strings – transcription). Seq2Seq architectures consist of an encoder and a decoder (Encoder-Decoder structure – ED). The role of the encoder is to take the speech signal and transform it into vector of a high-level representation (acoustic features) which is passed to the input of the decoder, that outputs a probability distribution for the current unit (e.g. a word). Finally, the most probable transcription of the input audio signal is obtained. Below we describe the most important solutions used in E2E ASR systems.

5.1. Recurrent Neural Networks – RNN

RNNs are suited for analyzing sequential data over time (such as speech) – they can take a sequence of values at the input and return a sequence of values at the output, 'remember' previous values of the output. In the context of ASR, the advantage of RNNs over conventional ASR is the absence of any prior knowledge of data (only the choice of input and output representations). They are also robust to temporal and spatial noise [39], but require pre-segmented training data and post-processing (transforming the results into labeled sequences). The real, often noisy input signal is labeled with a sequence of discrete letters or words, and speech represents connected units with unknown segmentation. RNNs can only be trained to produce a set of separate label classifications. Although the RNN has access to the entire previous sequence, the encoded hidden state information is usually rather local [40]. In response to these problems, types of RNNs have been developed specifically for ASR tasks: Connectionist Temporal Classification (CTC), Recurrent Neural Network Transducer (RNN-T) and Attention-based Encoder-Decoder (AED).

5.1.1. Connectionist Temporal Classification – CTC

It is a type of network output and associated scoring function for direct labeling of unsegmented sequences (see Fig. 4A) [41]. For a temporal classifier, no external post-processing is required, as CTC directly outputs the probabilities of label sequences by mapping the input speech sequence to the output label sequence. Its training leads to predicting labels in any input sequence, as long as the overall result is correct. Therefore there is no need to pre-segment data and match labels to the input data. If the length of the output labels is less than the length of the input speech sequence, a blank label is inserted in the former to align them. The utterance of each letter is characterized by a specific duration. To match the CTC output sequence to this duration, each letter in the output text is repeated, and the repetitions form a single letter [42]. The procedure described in the last step is the so-called 'collapsing together' of different paths into a single designation: the probability of some designation is calculated as the sum of the probabilities of all paths mapped to it by a many-to-one function (e.g. F(c-ar-) = F(-c-aa-r) = 'car'). This is possible because the paths are mutually exclusive. This is what makes it possible to skip data segmentation in CTC – this procedure allows the network to predict a label without prior knowledge of where it occurs. Thus, one may think that CTC is not a good model for problems in which the location of the label must be determined. However, it has been proved in experiments that CTC predicts labels in an approximate position relative to the input sequence [9, 42]. The CTC mechanism can also be combined with other networks (such as CNNs) as a loss function (CTC loss) in learning mapping input sequences to output sequences, even if they are of different lengths. Such a process involves performing backward calculations to identify all possible output sequences corresponding to a given input sequence. Losses are then calculated based on their probabilities. The Softmax activation function in Fig. 4 is currently the standard output layer component for making decisions:

$$Softmax = s(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}}. \tag{2}$$

Softmax transforms real numbers (from preceding parts of the network) into a probability vector. Here x is the softmax input vector (containing n elements for n possible categories).

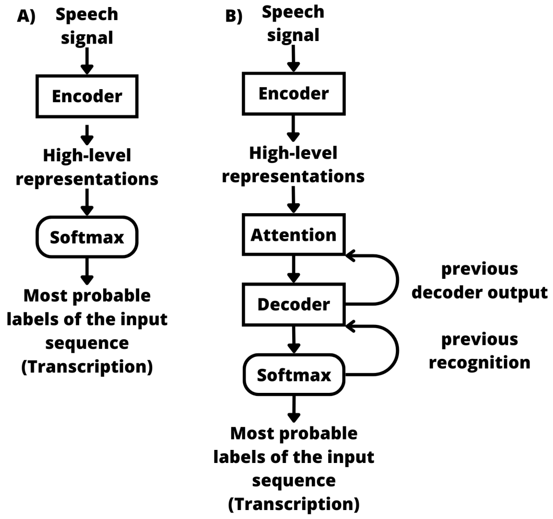
A)
Speech signal
Encoder
High-level representations
Softmax
Most probable labels of the input sequence (Transcription)
B)
Speech signal
Encoder
High-level representations
Attention
Decoder
Softmax
previous decoder output
previous recognition
Most probable labels of the input sequence (Transcription)

Fig. 4. Simplified CTC (A) and AED (B) structure [9]. The AED structure shows additional blocks of the attention mechanism and decoder, as well as added recursive paths

5.1.2. Recurrent Neural Network Transducer – RNN-T

RNN-T is an extended CTC model, in which information context learning is added (see Fig. 5). It is used in ASR as AM, while the additional long short-term memory network (LSTM, type of RNNs with additional memory cells [43]) is LM. Such a joint network combines language and acoustic features through a combination of high-level acoustic and language representations. In the CTC model, each recognition is conditionally separated, which is not the case for RNN-T. This allows them to be used in streaming recognition [9]. The performance of the RNN-T model can be explained based on four basic steps:

1. The role of the encoder is the same as that of the CTC model: generation features from an audio recording.
2. The prediction network generates a high-level representation based on the previous output of the whole model.
3. Based on the high-level representation joint network, which is usually a feed-forward network, combines the outputs of the decoder and the prediction network [44].
4. The output vector of step 3 is fed into a softmax layer to determine the output prediction of the whole model.

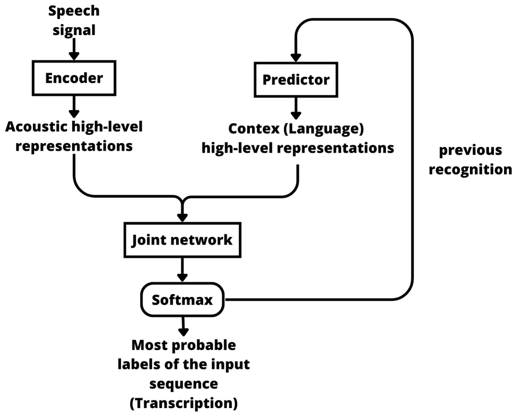
Speech signal
Encoder
Acoustic high-level representations
Predictor
Contex (Language) high-level representations
previous recognition
Joint network
Softmax
Most probable labels of the input sequence (Transcription)

Fig. 5. Simplified RNN-T structure [9]

5.1.3. Attention-based Encoder-Decoder – AED

AED solves a problem present in traditional ED architecture, in which the context vector is created only based on the last hidden state of the encoder, which can lead to limitations for long input sentences. The reason is that in an RNN, old information can be forgotten after propagation over many time steps and attention is scattered throughout the sequence (there is no obvious word alignment during decoding) [45]. The AED architecture (see Fig. 4B) solves this problem, as the encoder is a bidirectional RNN, while the decoder is an RNN working on the input from the previous state and a dynamic context vector. The latter is created by the attention layer located between the encoder and decoder. It accesses all the hidden states of the encoder and every part of the input sequence (word in the sentence) at the same time. The AED model is autoregressive at every step. It uses previously generated symbols as extra data of input during the generation of further ones. The performance of the AED model can also be explained based on four basic steps:

1. The encoder network has the same function as in CTC.
2. Taking the output representations of the encoder, the decoder outputs the sequence one element at a time.
3. The attention layer calculates attention weights between the previous decoder output and the encoder output of each frame with the attention function, then a context vector – the weighted sum of the encoder outputs – is generated.
4. The previous output label is given to the decoder input together with the context vector. Based on this, the decoder output is generated.

The attention mechanism (Fig. 6) gives each input a weight to evaluate its importance compared to other inputs. The attention weights between the previous decoder output and the encoder output of each frame are calculated using the attention function. Then a context vector – the weighted sum of the encoder outputs – is generated. The attention mechanism introduces the exploration of context relations in sequences to E2E systems. It simulates the human attention mechanism and allows the network to focus on important parts of the input data rather than the irrelevant ones [46].

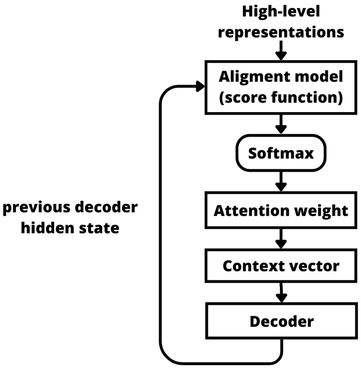
High-level representations
Aligment model (score function)
Softmax
previous decoder hidden state
Attention weight
Context vector
Decoder

Fig. 6. Attention mechanism used in AED structure

5.2. Convolutional neural networks – CNN

Although RNNs are well suited for classifying temporal sequences, their learning speed is slow for long input sequences due to iterative multiplications over time. Another approach to building E2E ASR models is to use CNNs. They are feedforward networks with convolution layers represented by a series of filters (matrices with numbers), each recognizing a particular pattern [47]. There are solutions fully based on CNNs such as described in [48]. CNNs in combination with CTC, or RNN-T layers can be found in the NVIDIA family models, such as Jasper [49], or its modifications, e.g. QuartzNet (described in Section 6.1). CNNs are also part of transformer-based architectures, such as speech-transformer (see Fig. 7) and conformer (see Fig. 8). They are consequently an integral part of most modern E2E ASR models. One reason for this is the widespread use of acoustic features in the form of mel-spectrograms and mel-frequency cepstrum coefficient (MFCC [13]) depiction, which allows speech sounds to be represented as images and adapts techniques originally used in the field of image recognition. The ASR task requires the model to consider long-term dependencies. According to [50], in CNNs of sufficient depth, higher layer features are able to capture temporal dependencies with relevant contextual information. By using small filter sizes along the frequency axis of the spectrogram, the model is adept at learning fine-grained localized features. Multiple stacked convolution layers are robust to translational frequency shifts (depending on age or gender of the speaker) [51]. In the aforementioned fully convolutional ASR, the model architecture is divided into four parts: a CNN front-end, CNN acoustic model, CNN language models and a beam-search decoder. The paper describes the equivalents of the techniques used for image processing and the process of adapting CNNs to speech signal (e.g., the pre-processing layer uses logarithmic compression and normalization of the mean variance per channel – the equivalent of the instance normalization layer used in CNN image processing [52]).

5.3. Transformer

Similar to AED, this network is based on ED, an attention mechanism, and is used to process sequential input data, but in transformer the input data is not processed sequentially. Hidden states can be computed in parallel, which reduces learning time. Transformer can also be self-supervised, so no labeling is required [53,54]. The no-data labeling requirements and parallel input access to all hidden states are attributes of the transformer that make it suitable for the ASR task. This section describes two major transformer-based ASR architectures: the speech-transformer and the conformer.

5.3.1. Speech-transformer

Speech-transformer is transformer adapted to ASR by replacing the embedding layer of the encoder with convolution layers, before passing features to transformer layers (see Fig. 7) [55]. Additional CNN layers in speech transformer reduce differences in dimensions of the input and output sequences. This is due to number of frames in the audio signal being greater than the number of output tokens (text) [55]. As in any ED-based ASR architecture, the encoder's task is to change the input speech sequence into a sequence of high-level representations – acoustic features. The decoder takes them with previously generated y_{i-1} character and returns the next one y_i [56]. In this architecture is the stack of M optional modules to extract more expressive representations (e.g. extra encoder blocks).

The 'Linear' transformation is done on the flattened output of the feature map. This allows for obtaining vectors of the

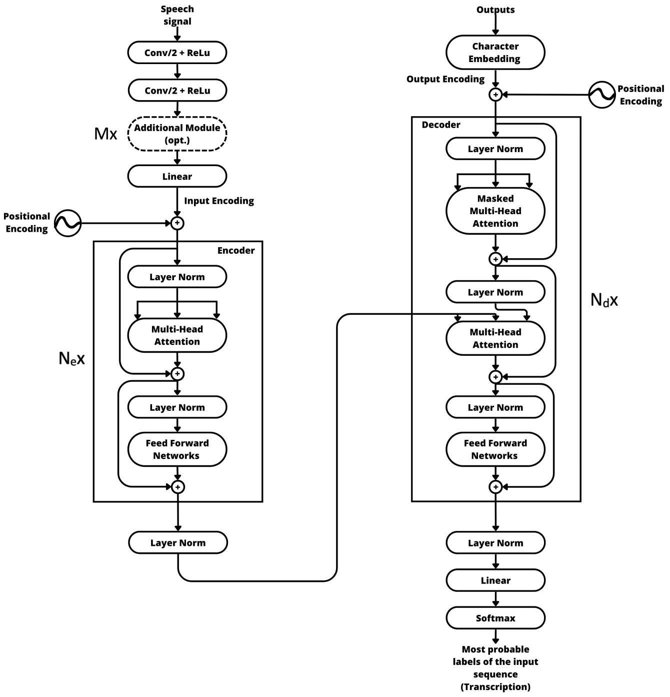

Fig. 7. Example of E2E ASR model in speech-transformer architecture [55], contains convolutional layers at the beginning, followed by transfromer layers with a multihead self-attention mechanism

correct dimension ('input encoding'). The sum of the input and positional encoding is fed into the encoder. It contains a MH attention [55] and a position-wise feed-forward network (FNN). The attention extracts contents from a set of queries Q and keys K of dimension d_k and values V of dimension d_v. The retrieval function is driven by similarities between queries and keys and returns a weighted sum of values:

$$Attention(Q,K,V) = Softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V. \tag{3}$$

The FNN is applied individually to each position in the sequence. Layer normalization and residual connections are implemented in each sub-block to achieve efficient training. Each decoder contains three sub-blocks: masked MH attention, MH attention with keys and values derived from the encoder outputs, and queries from the outputs of the previous sub-block, FNN. Masking ensures that predictions for a given position can only depend on known outputs in smaller positions. Decoder outputs are transformed into output class probabilities using linear projection and softmax function. The positional encoding mentioned above adds sequence information to each element of the sequence. This is necessary because the self-attention layer does not distinguish the order of the elements in the sequence.

5.3.2. Conformer

Conformer is a convolution-augmented transformer for speech recognition (see Fig. 8) [57]. The structure includes an Conformer encoder and a LSTM-based decoder. The former is similar to a standard AM, taking input attributes x, and mapping them to high-level features. The decoder takes these and, given the context extracted by the encoder, outputs a probability distribution for the current entity (a word) [58]. Conformer encoder is based on the idea of transformer, but with added convolution. In its architecture 'SpecAugumentation' refers to the data augmentation method for ASR from [59] and 'Dropout', is used to prevent overfitting [60]. Conformer encoder structure uses additional layers of CNNs to capture global and local context. The combination of CNN and transformer enables learning local position-dependent features and exploiting global content-based interactions. At the same time, this combination extends self-attention with relative position based information that maintains uniformity.

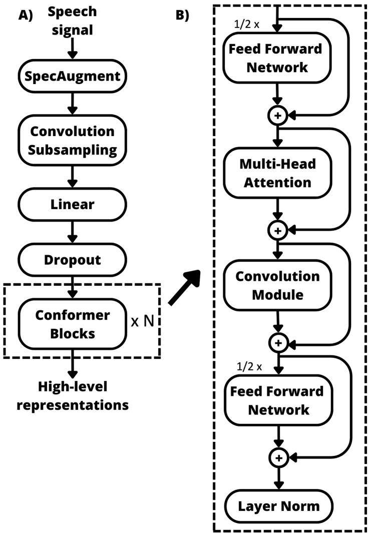

Fig. 8. Simplified structure of the conformer type encoder [57]. On the left is shown an simplified structure of the encoder using conformer blocks. The SpecAugment [59] block shown in the diagram refers to an augmentation method that operates on the logarithmic mel-spectrogram of the input audio, rather than the raw audio itself, treating it as a visual rather than an audio problem. Layer convolutional subsampling allows to reduce the dimensions of feature maps in CNN layers [61]. A dropout [60] layer protects the deep model from overfitting. On the right is shown the detailed structure of conformer blocks, which includes both layers of multi-headed attention and convolution layers

6. E2E DNN MODELS SELECTED FOR RESEARCH

For the considered ASR system (Fig. 3) we selected five E2E ASR models adapted for Polish ASR. Their performance for continuous speech is higher than conventional methods. E2E ASR is easier to design and train with lower data segmentation requirements (or absence thereof) and lack of forced alignment. Due to the limitations of RNNs mentioned in Section 5.1, we chose models mainly CNN and transformers-based. The goal is to compare their WER and use either as the standalone approach or the ensemble. In Table 2 we present WER values for selected models, based on the literature reports. The table also provides results for three previously undescribed datasets: **Polish Parliamentary Corpus** [62] (PPC) – a collection of documents from the proceedings of the Polish Parliament analyzed linguistically ([63] describes the use of recordings from the Polish Parliament to test ASR E2E models); **Europarl-st** [64] (E) – contains translations from publicly available video recordings of Euro-

Table 2

WER results for the investigated models based on literature sources. The abbreviations used are as follows: Arch. – ASR architecture, Network – type of network used, TF – transformer, CF – conformer, T – RNN-transducer

Ref.	Model	Arch.	Network	Data.	WER	Date
[66]	QuartzNet	BxR	CNN	MCV	14%	2023
[67]	FastConf.	ED	CF-T-CTC	MCV	5.99%	2023
[68]	ESPnet2	ED	CF	MCV	2.6%	2020
[68]	ESPnet2	ED	TF	MCV	15.1%	2020
[69]	Whisper	ED	CNN-TF	MCV	6%	2023
[70]	Wav2Vec 2.0	ED	CNN-TF	MCV	9.8%	2021
[71]	Wav2Vec 2.0	ED	CNN-TF	MCV/E	7.6%	2022
[72]	Wav2Vec 2.0	ED	CNN-TF	MLS	17.2%	2020
[73]	Whisper	ED	CNN-TF	MLS	5%	2022
[63]	Wav2Vec 2.0	ED	CNN-TF	PPC	32.1%	2023
[63]	Whisper	ED	CNN-TF	PPC	32.5%	2023
[70]	Wav2Vec 2.0	ED	CNN-TF	VP	7.1%	2021

pean Parliament debates for nine languages (including Polish), recordings and transcriptions; is divided into Training, Development and Test parts; **VoxPopuli** [65] (VP) – a multilingual speech corpus based on recordings from European Parliament events from 2009–2020.

6.1. QuartzNet

NVIDIA QuartzNet [66] model architecture (see Fig. 9) based on Jasper [49] with CNN trained using CTC loss function. The BxR architecture includes B blocks, each with R convolution sub-blocks, which allows for efficient implementation on the GPU. It uses spectrograms as input speech features. The main extension in the QuartzNet architecture is the replacement of 1D convolutions with 1D Time-Channel Separable Convolutions (1DTCSC). The input is the number of dimensions of the input data, while the output is the number of feature maps produced by a convolution filter. Here, 'time' refers to one-dimensional data. The filter moves along the time axis and the convolution is divided into time-wise and channel-wise operations. The former uses separate convolution filters for each data point in time. This allows us to analyze temporal data from different perspectives and detect patterns and relationships. The channel-wise operation consists in applying convolution to the resulting data to extract features in multiple channels. The 1DTCSC minimizes the model, reducing the number of network parameters, computational efficiency and preventing over-fitting [74]. The selected QuartzNet15x5 (available in the NVIDIA NeMo toolkit [75]) contains 15 CNN blocks multiplied 5 times. We use fine-tuned model from English to Polish: it exploits the encoder from the English version of QuartzNet, while the decoder was changed to output Polish alphabet characters and tuned using the Polish part of MCV [76]. The applied character coding scheme and text transcription is the standard character set available in the Polish part of the MCV dataset. The 'Conv-BN-ReLU' block applies one-dimensional convolution, batch normalization [77] and ReLU function [78]. The 'TCSDConv-BN-ReLU' block applied 1DTCSC, batch normalization and ReLU function.

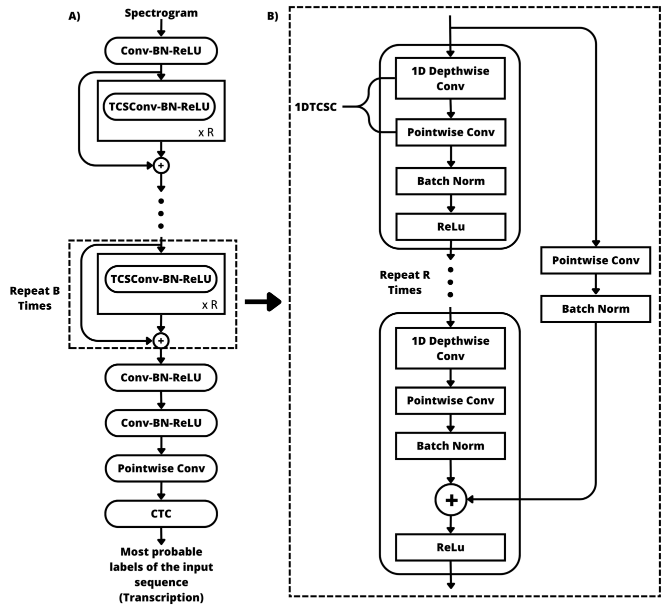

Fig. 9. Architecture of QuartzNet model [74]. Its structure mainly includes CNN layers, with a CTC layer at the output of the entire model

6.2. FastConformer Transducer – CTC

The model uses a FastConformer with joint transducer and CTC decoder loss [67]. To speed up the encoder, the down-sampling rate was increased (from 4× to 8×), reducing the sequence length of the speech features and the computational cost of subsequent attention layers [79]. The model uses a hybrid decoder, i.e. a combination of RNN-T and CTC (Transducer-CTC) [80]. It is available in the NVIDIA NeMo Toolkit [75] and was adapted to Polish using the MCV, MLS and VP. It also uses the Google SentencePiece Unigram tokenizer [81], transcribes text in uppercase and lowercase letters of the English alphabet together with spaces, periods, commas, question marks and several other characters.

6.3. Wav2Vec 2.0 XLSR-53

It is a framework developed by Facebook AI for self-supervised learning of speech representations, using a CNN and a transformer. The raw speech waveform is fed to the input of the CNN encoder, with output receiving hidden speech representations. They are then fed to the input of transformer encoder whose output is processed by a quantization module to represent targets for self-supervised learning. The model builds context representations on continuous speech [82]. [72] shows the use of the Wav2Vec 2.0 framework for unsupervised learning of the ASR XLSR-53 multilingual model. It covers 53 languages, including Polish. Tuning the model to the new languages was done by training with CTC loss and using MCV, MLS and Babel [83] datasets. The [63] tested the capabilities of the Whisper and Wav2Vec 2.0 models to detect keywords in the child abuse domain. A list of keywords for detection was defined and obtained from a set of open documents. All documents were searched and preprocessed by lemmatizing and removing stop words, numbers and date units.

6.4. Whisper

This is OpenAI's open-source, general-purpose, multilingual ASR model, based on the transformer ED with two CNN layers at the top of the encoder structure (speech-transformer), supporting 57 languages (including Polish [69]). The Whisper model exists in several versions: tiny, base, small, medium and large. It maps utterances and their transcription by predicting the raw text of the transcription without significant standardization or preprocessing [73]. This allows for skipping the separate step of reverse normalization of the text to obtain the correct transcription. The Whisper model was trained on an extensive dataset of audio and transcriptions from the Internet. It was varied in sound and transcription quality. While diversity in audio quality can help train the model to be robust to speech signal quality, diversity in transcription quality is not similarly beneficial. Because of this, automatic methods were used to filter transcriptions to improve their quality. Heuristics based on punctuation, capitalization and other features were also used to detect and remove machine-generated transcriptions from the training dataset

6.5. ESPnet

This is an open source Toolkit for E2E speech processing, includes DNN-based models described earlier such as: CTC, AED, BLSTM, RNN-T, hybrid CTC/attention, AED/BLSTM and transformer/conformer ASR (supports streaming) [84] and a pre-trained multilingual model with available Polish [85]. In [68] to test the implementation of the conformer architecture in the ESPnet toolkit, the corpora used were subjected to the same data preparation procedure as in Kaldi [86].

7. CONCLUSIONS

The paper presents the current state of ASR methodology for Polish language. The aim of the analysis was to identify the main tools and algorithms applicable to ASR and to identify those that could potentially be adapted to conversations conducted with impaired acoustic signal transmission. First, the available approaches were categorized and compared. We analyzed the performance of conventional and E2E systems. The best conventional ASR achieved a low WER for short commands or continuous speech with HMM enhancement by DNN (ARM-1 NG). The best E2E model considered is Whisper [73], with a WER of 5%, so it can provide a baseline for evaluating the quality of such models. The WERs for ARM-1 NG (4.84%) and Whisper (5%) are similar, but the second one is an open-source project that uses state-of-the-art ASR technology, making it easier to tune for a new task. It should be noted, that the low WER for the Whisper model was achieved for the most popular datasets (MCV and MLS) in developing and testing E2E ASR models. Tests of this model, on a less popular dataset, showed a significant drop in its performance (WER of more than 30% for the PPC dataset). Therefore, it can be assumed that the MCV and MLS datasets were used in training Whisper, results from the model fitting to the data. To determine the actual performance of the Whisper model, it is necessary to conduct tests of the model, for data that we know for sure were not involved in the training. Model over-fitting to data is one of the main problems of DNN-based solutions. In addition, such models have very high computational requirements. This is a significant challenge for these models, so scaling is important in the upcoming work.

REFERENCES

[1] J. Meyer, L. Dentel, and F. Meunier, "Speech recognition in natural background noise," *PLOS ONE*, vol. 8, no. 11, p. e79279, Nov. 2013, doi: 10.1371/journal.pone.0079279.

[2] H.K. Kim and R.C. Rose, "Speech recognition over mobile networks," in *Automatic Speech Recognition on Mobile Devices and over Communication Networks*. London: Springer London, 2008, no. 1, pp. 41–61, doi: 10.1007/978-1-84800-143-5_3.

[3] R.S. Chavan and G.S. Sable, "An overview of speech recognition using hmm," *Int. J. Comput. Sci. Mob. Comput.*, vol. 2, no. 6, pp. 233–238, Jun. 2013.

[4] S. Furui, "50 years of progress in speech and speaker recognition research," *ECTI Trans. Comput. Inf. Technol.*, vol. 1, no. 2, pp. 64–74, Jan. 1970, doi: 10.37936/ecti-cit.200512.51834.

[5] G.E. Hinton *et al.*, "Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 82–97, Nov. 2012, doi: 10.1109/msp.2012.2205597.

[6] S. Wang and G. Li, "Overview of end-to-end speech recognition," *J. Phys.*, vol. 1187, no. 5, p. 052068, Apr. 2019, doi: 10.1088/1742-6596/1187/5/052068.

[7] "Speech-to-Text: Automatic Speech Recognition | Google Cloud," (2023). [Online]. Available: https://cloud.google.com/speech-to-text (Accessed 2024-03-13).

[8] M.J.F. Gales and S. Young, "The application of hidden Markov models in speech recognition," *Found. Trends Signal Process.*, vol. 1, no. 3, pp. 195–304, Jan. 2007, doi: 10.1561/2000000004.

[9] J. Li, "Recent advances in End-to-End automatic speech recognition," *APSIPA Trans. Signal Inf. Proc.*, vol. 11, no. 1, Jan. 2022, doi: 10.1561/116.00000050.

[10] P. Rybak, R. Mroczkowski, J. Tracz, and I. Gawlik, "KLEJ: Comprehensive benchmark for Polish language understanding," in *Proc. 58th Annu. Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, 5-10 Jul. 2020, pp. 1191–1201, doi: 10.18653/v1/2020.acl-main.111.

[11] A. Magueresse, V. Carles, and E. Heetderks, "Low-resource Languages: A review of past work and future challenges," *arXiv (Cornell University)*, Jun. 2020, doi: 10.48550/arxiv.2006.07264.

[12] E. Trentin and M. Gori, "A survey of hybrid ANN/HMM models for automatic speech recognition," *Neurocomputing*, vol. 37, no. 1-4, pp. 91–126, Apr. 2001, doi: 10.1016/s0925-2312(00)00308-8.

[13] M.A. Mazumder and R.A. Salam, "Feature Extraction Techniques for Speech Processing: A Review," *Int. J. Adv. Trends. Comput. Sci. Eng.*, vol. 8, no. 1.3, pp. 285–292, 2019.

[14] A. Zygadło, A. Janicki, and P. Dąbek, "Automatic speech recognition system for Polish dedicated for a social robot," *PAR. Pomiary Automatyka Robotyka*, vol. 4/2016, pp. 27–36, Dec. 2016, doi: 10.14313/par_222/27.

[15] A. Janicki and D. Wawer, "Automatic speech recognition for Polish in a computer game interface," in *2011 Federated Conf. on Computer Science and Information Systems (FedCSIS)*. Szczecin, Poland: IEEE, 18-21 Sep. 2011, pp. 711–716. [Online]. Available: https://ieeexplore.ieee.org/document/6078265

[16] L. Pawlaczyk and P. Bosky, "Skrybot – a system for automatic speech recognition of Polish language," in *Man-Machine Interactions*. Berlin, Heidelberg: Springer, Jan. 2009, vol. 59, pp. 381–387, doi: 10.1007/978-3-642-00563-3_40.

[17] J. Jamrozy, M. Lange, M. Owsianny, and M. Szymanski, "Arm-1: Automatic speech recognition engine," in *Proc. PolEval 2019 Workshops*. Warsaw, Poland: Institute of Computer Science, Polish Academy of Sciences, 2019, pp. 79–88. [Online]. Available: http://poleval.pl/files/poleval2019.pdf

[18] R. Cecko, J. Jamroż, W. Jęśko, E. Kuśmierek, M. Lange, and M. Owsianny, "Automatic Speech Recognition and its Application to Media Monitoring," *Computational Methods in Science and Technology*, vol. 27, no. 2, pp. 41–55, Nov. 2021, doi: 10.12921/cmst.2021.0000015.

[19] W. Majewski, H.B. Rothman, and H. Hollien, "Acoustic compar-

isons of American English and Polish," *J. Phon.*, vol. 5, no. 3, pp. 247–251, Jul. 1977, doi: 10.1016/s0095-4470(19)31138-6.

[20] W. Majewski, H. Hollien, and J. Zalewski, "Speaking fundamental frequency of Polish adult males," *Phonetica*, vol. 25, no. 2, pp. 119–125, Mar. 1972, doi: 10.1159/000259375.

[21] G. Demenko *et al.*, "Development of large vocabulary continuous speech recognition for Polish," *Acta Phys. Pol. A*, Jan. 2012, doi: 10.12693/aphyspola.121.a-86.

[22] J.C. Wells, "Computer-coding the IPA: a proposed extension of SAMPA," (1995). [Online]. Available: https://www.phon.ucl.ac.uk/home/sampa/ipasam-x.pdf (Accessed 2024-03-13).

[23] R.F. Feldstein, *A concise Polish grammar*. Slavic and East European Language Research Center (SEELRC), Duke University, 2001.

[24] E. Rudnicka, M. Maziarz, M. Piasecki, and S. Szpakowicz, "A strategy of mapping Polish wordnet onto princeton wordnet," in *Proc. 24th COLING 2012: Posters*. Mumbai, India: COLING 2012, 8-15 Dec. 2012, pp. 1039–1048.

[25] G. Rehm and H. Uszkoreit, *The Polish language in the digital age*. Berlin, Heidelberg: Springer, Jan. 2012, doi: 10.1007/978-3-642-30811-6.

[26] A. Pohl and B. Ziółko, "Using part of speech n-grams for improving automatic speech recognition of Polish," in *Machine Learning and Data Mining in Pattern Recognition. Proc. of 9th Int. Conf., MLDM 2013*. New York, NY, USA: Springer Berlin, Heidelberg, 19-25 Jul. 2013, pp. 492–504, doi: 10.1007/978-3-642-39712-7_38.

[27] K. Marasek, L. Brocki, D. Korzinek, K. Wołk, and R. Gubrynowicz, "Spoken language translation for Polish," *arXiv (Cornell University)*, Nov. 2015, doi: 10.48550/arXiv.1511.07788.

[28] R. Rosenfeld, "Two decades of statistical language modeling: where do we go from here?" *Proc. of the IEEE*, vol. 88, no. 8, pp. 1270–1278, Aug. 2000, doi: 10.1109/5.880083.

[29] J. Nouza, P. Cerva, and R. Safarik, "Cross-Lingual adaptation of broadcast transcription system to Polish language using public data sources," in *Human Language Technology. Challenges for Computer Science and Linguistics. 7th Language and Technology Conf., LTC 2015*. Poznań, Poland: Springer Cham, 27-29 Nov. 2015, pp. 31–41, doi: 10.1007/978-3-319-93782-3_3.

[30] B. Ziółko, S. Manandhar, R. C. Wilson, M. Ziółko, and J. Galka, "Application of HTK to the Polish language," in *ICALIP 2008 Int. Conf. on Audio, Language and Image Processing*. Shanghai, China: IEEE, 7-9 Jul. 2008, pp. 31–41, doi: 10.1109/ICALIP.2008.4590266.

[31] J. Nouza, R. Safarik, and P. Cerva, "ASR for South Slavic Languages Developed in Almost Automated Way." in *INTERSPEECH 2016*. San Francisco, USA: ISCA Speech, 8-12 Sep. 2016, pp. 3868–3872, doi: 10.21437/Interspeech.2016-747.

[32] B. Ziółko, P. Żelasko, I. Gawlik, T. Pędzimąż, and T. Jadczyk, "An Application for Building a Polish Telephone Speech Corpus." in *LREC 2018, 11th Int. Conf. on Language Resource and Evaluation*, I. Calzolari, Ed. Miyazaki, Japan: ELRA, 7-12 May 2018, pp. 429–433.

[33] S. Grocholewski, "CORPORA – speech database for Polish diphones," in *Proc. 5th European Conf. on Speech Communication and Technology (Eurospeech 1997)*. Rhodes, Greece: ISCA Speech, 22-25 Sep. 1997, pp. 1735–1738, doi: 10.21437/Eurospeech.1997-492.

[34] V. Pratap, Q. Xu, A. Sriram, G. Synnaeve, and R. Collobert, "Mls: A large-scale multilingual dataset for speech research," in *INTERSPEECH 2020*. Shanghai, China: ISCA Speech, 25-29 Oct. 2020, pp. 2757–2761, doi: 10.21437/Interspeech.2020-2826.

[35] R. Ardila *et al.*, "Common Voice: a Massively-Multilingual Speech Corpus," in *Proc. of the 12th Conf. on Language Resources and Evaluation (LREC 2020)*. Marseille, France: ELRA, 11–16 May 2020, pp. 4218–4222. [Online]. Available: https://www.aclweb.org/anthology/2020.lrec-1.520.pdf

[36] "Common Voice open source, multi-language dataset of voices," (2023). [Online]. Available: https://commonvoice.mozilla.org/en/datasets (Accessed 2023-03-23).

[37] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *2015 IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*. South Brisbane, QLD, Australia: IEEE, 19-24 Apr. 2015, p. 5206–5210, doi: 10.1109/ICASSP.2015.7178964.

[38] J. Gu, Z. Lu, H. Li, and V. O. Li, "Incorporating copying mechanism in sequence-to-sequence learning," in *Proc. 54th Annu. Meeting of the Association for Computational Linguistics (Volume 1: Long Papers*. Berlin, Germany: Association for Computational Linguistics, 7-12 Aug. 2016, pp. 1631–1640, doi: 10.18653/v1/P16-1154.

[39] O. Vinyals, S.V. Ravuri, and D. Povey, "Revisiting recurrent neural networks for robust asr," in *2012 IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*. Kyoto, Japan: IEEE, 25-30 Mar. 2012, pp. 4085–4088, doi: 10.1109/ICASSP.2012.6288816.

[40] "Speech and Language Processing (3rd ed. draft) Dan Jurafsky and James H. Martin," (2023). [Online]. Available: https://web.stanford.edu/~jurafsky/slp3/ (Accessed 2023-07-30).

[41] M.W. Kadous *et al.*, "Temporal classification: Extending the classification paradigm to multivariate time series," Ph.D. dissertation, The University of New South Wales, 2002.

[42] A. Graves, *Supervised Sequence Labelling with Recurrent Neural Networks*, ser. Studies in computational intelligence. Berlin, Heidelberg: Springer, 2012, doi: 10.1007/978-3-642-24797-2.

[43] A.N. Shewalkar, "Comparison of rnn, lstm and gru on speech recognition data," Master's thesis, North Dakota State University, 2018.

[44] A. Graves, A.R Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. 2013 IEEE Int. Conf. on Acoustics, Speech and Signal Processing*. Vancouver, BC, Canada: IEEE, March 2013, pp. 6645–6649, doi: 10.1109/ICASSP.2013.6638947.

[45] D. Hu, "An introductory survey on attention mechanisms in nlp problems," in *Intelligent Systems and Applications. Proc. of the 2022 Intelligent Systems Conf. (IntelliSys)*, Y. Bi, R. Bhatia, and S. Kapoor, Eds. Amsterdam, Netherlands: Springer, 1-2 Sep. 2020, pp. 432–448, doi: 10.1007/978-3-030-29513-4_31.

[46] X. Zhu, D. Cheng, Z. Zhang, S. Lin, and J. Dai, "An empirical study of spatial attention mechanisms in deep networks," in *Proc. 2019 IEEE/CVF Int. Conf. on Computer Vision (ICCV)*. Seoul, Korea (South): IEEE Computer Society, 27 Oct.–2 Nov. 2019, pp. 46 687–6696, doi: 10.1109/ICCV.2019.00679.

[47] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," *arXiv (Cornell University)*, Nov. 2015, doi: 10.48550/arXiv.1511.08458.

[48] N. Zeghidour, Q. Xu, V. Liptchinsky, N. Usunier, G. Synnaeve, and R. Collobert, "Fully convolutional speech recognition," *arXiv (Cornell University)*, Dec. 2018, doi: 10.48550/arXiv.1812.06864.

[49] J. Li *et al.*, "Jasper: An end-to-end convolutional neural acoustic model," in *INTERSPEECH 2019*. Graz, Austria: ISCA Speech, 15–19 Sep. 2019, pp. 71–75, doi: 10.21437/Interspeech.2019-1819. [Online]. Available: https://www.isca-speech.org/archive/interspeech_2019/li19_interspeech.html

[50] Y. Zhang, M. Pezeshki, P. Brakel, S. Zhang, C.L.Y. Bengio, and A. Courville, "Towards end-to-end speech recognition with deep convolutional neural networks," *arXiv (Cornell University)*, Jan. 2017, doi: 10.48550/arXiv.1701.02720.

[51] W. Song and J. Cai, "End-to-end deep neural network for automatic speech recognition," *Stanford CS224D Reports*, 2015. [Online]. Available: http://cs224d.stanford.edu/reports/SongWilliam.pdf

[52] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Instance normalization: The missing ingredient for fast stylization," *arXiv (Cornell University)*, Jul. 2016, doi: 10.48550/arXiv.1607.08022.

[53] A. Vaswani *et al.*, "Attention is all you need," in *Advances in Neural Information Processing Systems 30 (NIPS 2017). Proc. of 31st Annu. Conf. on Neural Information Processing Systems*, U. Von Luxburg, Ed. Long Beach, CA, USA: Neural Information Processing Systems Foundation, Inc. (NeurIPS), Dec. 2017, pp. 5999–6010. [Online]. Available: http://papers.nips.cc/paper/7181-attention-is-all-you-need

[54] C. Subakan, M. Ravanelli, S. Cornell, M. Bronzi, and J. Zhong, "Attention is all you need in speech separation," in *2021 IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*. Toronto, ON, Canada: IEEE, 6-11 Jun. 2021, pp. 21–25, doi: 10.1109/ICASSP39728.2021.9413901.

[55] L. Dong, S. Xu, and B. Xu, "Speech-transformer: A no-recurrence sequence-to-sequence model for speech recognition," in *2018 IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*. Calgary, AB, Canada: IEEE, 15–20 Apr. 2018, pp. 5884–5888, doi: 10.1109/ICASSP.2018.8462506.

[56] E. Tsunoo, Y. Kashiwagi, T. Kumakura, and S. Watanabe, "Towards online end-to-end transformer automatic speech recognition," *arXiv (Cornell University)*, Oct. 2019, doi: 10.48550/arXiv. 1910.11871.

[57] A. Gulati *et al.*, "Conformer: Convolution-augmented transformer for speech recognition," in *INTERSPEECH 2020*. Shanghai, China: ISCA Speech, 25-29 Oct. 2020, pp. 5036–5040, doi: 10.21437/Interspeech.2020-3015.

[58] C.C. Chiu *et al.*, "State-of-the-art speech recognition with sequence-to-sequence models," in *2018 IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*. Calgary, AB, Canada: IEEE, 15-20 Apr. 2018, pp. 4774–4778, doi: 10.1109/ICASSP.2018.8462105.

[59] D.S. Park, W. Chan, Y. Zhang, C.C. Chiu, B. Zoph, E.D. Cubuk, and Q.V. Le, "SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition," in *INTERSPEECH 2019*. Graz, Austria: ISCA Speech, 15-19 Sep. 2019, pp. 2613–2617, doi: 10.21437/Interspeech.2019-2680.

[60] N. Srivastava, G.E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, Jan. 2014. [Online]. Available: https://jmlr.org/papers/v15/srivastava14a.html

[61] J. Xu, H. Kim, T. Rainforth, and Y. Teh, "Group equivariant subsampling," in *Advances in Neural Information Processing Systems (NEURLPS 2021)*, vol. 34. Curran Associates, Inc., 2021, pp. 5934–5946. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2021/file/2ea6241cf767c279cf1e80a790df1885-Paper.pdf

[62] M. Ogrodniczuk, "Polish parliamentary corpus," in *Proc. LREC 2018 Workshop "ParlaCLARIN: Creating and Using Parliamentary Corpora*," F. d. J. Darja Fišer, Maria Eskevich, Ed., Miyazaki, Japan, 7 May 2018, pp. 15–19. [Online]. Available: http://lrec-conf.org/workshops/lrec2018/W2/pdf/book_of_proceedings.pdf

[63] J.C. Vásquez-Correa and A. Álvarez Muniain, "Novel Speech Recognition Systems Applied to Forensics within Child Exploitation: Wav2vec2. 0 vs. Whisper," *Sensors*, vol. 23, no. 4, p. 1843, Feb. 2023, doi: 10.3390/s23041843.

[64] J. Iranzo-Sánchez *et al.*, "Europarl-st: A multilingual corpus for speech translation of parliamentary debates," in *2020 IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*. Barcelona, Spain: IEEE, 4-8 May 2020, pp. 8229–8233, doi: 10.1109/ICASSP40776.2020.9054626.

[65] C. Wang *et al.*, "VoxPopuli: A Large-Scale Multilingual Speech Corpus for Representation Learning, Semi-Supervised Learning and Interpretation," *arXiv (Cornell University)*, Jan. 2021, doi: 10.48550/arXiv.2101.00390.

[66] "STT Pl Quartznet15x5: Speech To Text (STT) model based on QuartzNet for recognizing Polish speech," (2023). [Online]. Available: https://catalog.ngc.nvidia.com/orgs/nvidia/teams/nemo/models/stt_pl_quartznet15x5 (Accessed 2023-07-31).

[67] "NVIDIA FastConformer-Hybrid Large (pl): Polish FastConformer Hybrid (Transducer and CTC) Large model with Punctuation and Capitalization," (2023). [Online]. Available: https://catalog.ngc.nvidia.com/orgs/nvidia/teams/nemo/models/stt_pl_fastconformer_hybrid_large_pc (Accessed 2023-09-12).

[68] P. Guo *et al.*, "Recent developments on espnet toolkit boosted by conformer," in *2021 IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*. Toronto, ON, Canada: IEEE, 6-11 Jun. 2021, pp. 5874–5878, doi: 10.1109/ICASSP39728.2021.9414858.

[69] "Whisper: Github repository," (2023). [Online]. Available: https://github.com/openai/whisper (Accessed 2023-06-25).

[70] A. Babu *et al.*, "XLS-R: Self-supervised cross-lingual speech representation learning at scale," in *INTERSPEECH 2022*. Incheon, Kore: ISCA Speech, 18-22 Sep. 2022, pp. 2278–2282, doi: 10.21437/Interspeech.2022-14. [Online]. Available: https://www.isca-speech.org/archive/pdfs/interspeech_2022/babu22_interspeech.pdf

[71] N.Q. Pham, A. Waibel, and J. Niehues, "Adaptive multilingual speech recognition with pretrained models," in *18-22 INTERSPEECH 2022*. Incheon, Kore: ISCA Speech, Sep. 2022, pp. 3879–3883, doi: 10.21437/Interspeech.2022-872.

[72] A. Conneau, A. Baevski, R. Collobert, A. Mohamed, and M. Auli, "Unsupervised cross-lingual representation learning for speech recognition," in *INTERSPEECH 2021*. Brno, Czechia: ISCA Speech, 30 Aug.–3 Sep. 2021, pp. 2426–2430, doi: 10.21437/Interspeech.2021-329.

[73] A. Radford, J.W. Kim, T. Xu, G. Brockman, C. Mcleavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," in *Proc. of the 40th Int. Conf. on Machine Learning*, ser. Proceedings of Machine Learning Research, A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, Eds., vol. 202. Honolulu, HI, United States: PMLR, 23–29 Jul 2023, pp. 28 492–28 518. [Online]. Available: https://proceedings.mlr.press/v202/radford23a.html

[74] S. Kriman *et al.*, "Quartznet: Deep automatic speech recognition with 1d time-channel separable convolutions," in *2020 IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*. Barcelona, Spain: IEEE, 4-8 May 2020, pp. 6124–6128. [Online]. Available: https://ieeexplore.ieee.org/xpl/conhome/9040208/proceeding?isnumber=9052899&sortType=vol-only-seq&searchWithin=Quartznet

[75] "NVIDIA NeMo: conversational AI toolkit," (2023). [Online]. Available: https://github.com/NVIDIA/NeMo (Accessed 2023-06-23).

[76] J. Huang *et al.*, "Cross-language transfer learning, continuous learning, and domain adaptation for end-to-end automatic speech recognition," in *2021 IEEE Int. Conf. on Multimedia and Expo (ICME)*. Shenzhen, China: IEEE, 05-09 Jul. 2021, pp. 1–6, doi: 10.1109/ICME51207.2021.9428334.

[77] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. of 32nd Int. Conf. on Machine Learning (ICML 2015)*, D.B. Francis Bach, Ed., vol. 37. Lille, France: PMLR, 6–11 Jul. 2015, pp. 448–456. [Online]. Available: https://proceedings.mlr.press/v37/

[78] A.F. Agarap, "Deep learning using rectified linear units (relu)," *arXiv (Cornell University)*, Jan. 2018, doi: 10.48550/arXiv.1803.08375.

[79] D. Rekesh *et al.*, "Fast conformer with linearly scalable attention for efficient speech recognition," *arXiv (Cornell University)*, May 2023, doi: 10.48550/arXiv.2305.05084.

[80] "NVIDIA Hybrid-Transducer-CTC models," (2023). [Online]. Available: https://docs.nvidia.com/deeplearning/nemo/user-guide/docs/en/main/asr/models.html#hybrid-transducer-ctc (Accessed 2023-07-25).

[81] "Google SentencePiece Unigram: Github repository," (2023). [Online]. Available: https://github.com/google/sentencepiece (Accessed 2024-01-23).

[82] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," in *Advances in Neural Information Processing Systems 33: Annu. Conf. on Neural Information Processing Systems (NeurIPS 2020)*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds. Curran Associates, Inc., 6-12 Dec. 2020, pp. 12 449–12 460. [Online]. Available: https://proceedings.neurips.cc/paper/2020/hash/92d1e1eb1cd6f9fba3227870bb6d7f07-Abstract.html

[83] P. Roach *et al.*, "Babel: An eastern european multi-language database," in *Proc. of 4th Int. Conf. on Spoken Language Processing (ICSLP 96)*, vol. 3, Philadelphia, PA, USA, 3-6 Oct. 1996, pp. 1892–1893, doi: 10.21437/ICSLP.1996. [Online]. Available: https://www.isca-speech.org/archive/icslp_1996/roach96_icslp.html

[84] S. Watanabe *et al.*, "Espnet: End-to-end speech processing toolkit," in *INTERSPEECH 2018*. Hyderabad, India: ISCA Speech, 2-6 Sep. 2018, pp. 2207–2211, doi: 10.21437/Interspeech.2018-1456.

[85] "ESPnet Model Zoo: Github repository," (2021). [Online]. Available: https://github.com/espnet/espnet_model_zoo (Accessed 2024-03-13).

[86] D. Povey *et al.*, "The Kaldi speech recognition toolkit," in *Proc. ASRU. IEEE*, Hawaii, USA, 2011. [Online]. Available: https://infoscience.epfl.ch/record/192584