STATISTICS, OPTIMIZATION AND INFORMATION COMPUTING
Stat., Optim. Inf. Comput., Vol. **x**, Month 202x, pp 0–19.
Published online in International Academic Press (www.IAPress.org)

Development of a Model for Voice Identification with Accent-Specific Adaptation

Timur Shormanov [1], Talgat Mazakov [2,*], Sholpan Jomartova [2], Gumyrbek Toikenov [3], Aigerim Mazakova [2]

[1]*Department of Computer Science, Al-Farabi Kazakh National University, Republic of Kazakhstan*
[2]*Department of Artificial Intelligence and Big Data, Al-Farabi Kazakh National University, Republic of Kazakhstan*
[3]*Department of Computer Science, Kazakh National Women's Teacher Training University, Almaty, Republic of Kazakhstana*

Abstract This study developed a neural network-based model for voice identification with accent-specific adaptation, integrating Mel-Frequency Cepstral Coefficients (MFCC) feature extraction and fine-tuned BERT (Bidirectional Encoder Representations from Transformers) architectures. The research aimed to evaluate how contextualised transformer models can recognise accented speech across multiple languages and to construct a specialised kazakhBERTmulti model for the Kazakh accent. The selected languages – Kazakh, Russian, English, Spanish, and Chinese - represent typologically diverse systems with both high- and low-resource linguistic environments, enabling a comprehensive assessment of model adaptability. The Kazakh and Russian models were directly developed and trained, while English, Spanish, and Chinese data were used for comparative benchmarking of existing pretrained BERT variants. The voice identification pipeline included audio preprocessing (noise reduction, normalisation, segmentation), MFCC feature extraction using mel-filters and the mel-scale to approximate human auditory perception, vector quantisation for tokenisation, and BERT fine-tuning with classification layers. The bidirectional attention of BERT enables the model to capture long-range phonotactic dependencies, stress placement, and coarticulatory cues crucial for accent recognition. Performance comparison revealed language-dependent differences: English BERT achieved 90-96% accuracy due to its large training corpus, while multilingual BERT reached 70-85% accuracy and 70-80% F1. The proposed kazakhBERTmulti model attained an average F1 = 0.68, surpassing Multilingual BERT-RU and effectively adapting to the agglutinative structure of Kazakh. Overall accuracy reached 92%, confirming that integrating MFCC transformation with contextual BERT representations provides a robust framework for accent-aware speech identification.

Keywords Speech Recognition, Tokenisation, Mel-Filters, Audio Processing, Natural Language Processing, Multilingualism

AMS 2010 subject classifications 68T07, 68T05, 62M45

DOI: 10.19139/soic-2310-5070-2893

1. Introduction

The relevance of developing neural network-based models for automatic voice identification with accent-specific adaptation, particularly targeting the Kazakh accent, arises from both technological and sociolinguistic needs. Automatic speech recognition (ASR) has become a core component of modern digital ecosystems – from virtual assistants and call-centre analytics to biometric verification and accessibility services. However, one of the persistent challenges of ASR systems is their insufficient robustness to accented and non-standard speech. This limitation is especially critical for multilingual countries such as Kazakhstan, where the coexistence of Kazakh,

*Correspondence to: Talgat Mazakov (Email: mazakovtalgat81@gmail.com). Department of Artificial Intelligence and Big Data, Al-Farabi Kazakh National University. 71 Al-Farabi Ave., Almaty, Republic of Kazakhstan (050040).

ISSN 2310-5070 (online) ISSN 2311-004X (print)

Russian, and regional dialects creates significant acoustic and phonetic variability. Developing accent-adaptive neural network models can substantially improve recognition accuracy, support the inclusion of the Kazakh language in international ASR platforms, and enhance communication efficiency in public and private digital services. In this context, accent-aware modelling contributes not only to linguistic diversity preservation but also to strengthening national digital infrastructure through more inclusive and accurate human-computer interaction.

The Kazakh language exhibits unique phonetic, grammatical, and prosodic properties, varying across its northern, western, and eastern dialectal zones. Such diversity complicates the task for neural network-based systems to achieve accurate recognition, particularly when models are not trained to incorporate accentual and regional variability. Therefore, developing models capable of accounting for both segmental (vowel/consonant realisation) and suprasegmental (intonation, stress, rhythm) variation is essential for improving recognition accuracy and ensuring stable performance across speakers and recording conditions. Given the broad application of neural networks – from healthcare to education – adaptive voice-recognition models are crucial for multilingual environments. This is particularly relevant in Kazakhstan, where the increasing use of the Kazakh language in administration, education, and business creates a growing demand for high-precision speech-recognition systems serving users with diverse accents.

The use of neural networks for accent-sensitive voice identification is key to improving user experience and accessibility in digital ecosystems. It also facilitates the integration of the Kazakh language into international speech-recognition frameworks, enhancing its global visibility and supporting cultural inclusivity within AI-driven communication systems. Consequently, the development of neural models that incorporate accentual features – especially those specific to Kazakh – strengthens both the country's digitalisation and the preservation of linguistic diversity.

Prior research has explored several language-recognition architectures. Kadyrbek et al. [1] developed a convolutional neural network for Kazakh language recognition using fixed character-level filters and evaluated its efficiency in improving accuracy. Mukhamadiyev et al. [2] presented a deep-learning-based model for automatic Uzbek language recognition, while Mamyrbayev et al. [3] investigated transformer-based end-to-end architectures for Kazakh. However, these studies did not compare the performance of different languages using Bidirectional Encoder Representations from Transformers (BERT) models.

Najafian and Russell [4] proposed an automatic accent-identification model to improve ASR robustness, whereas Schnoor et al. [5] applied deep neural networks for accent estimation and quantification. Li et al. [6] reviewed models for Chinese dialect recognition, highlighting methodological diversity across dialectal systems. Despite this progress, accent-aware modelling for low-resource languages such as Kazakh remains insufficiently addressed.

Recent works [7, 8, 9, 10] emphasise that transformer-based systems outperform traditional convolutional or recurrent neural networks in capturing long-range dependencies essential for accent recognition. Unlike CNNs that extract only local spectral patterns or RNNs limited by sequential propagation, the BERT architecture employs a bidirectional self-attention mechanism that simultaneously analyses preceding and following phonemes. This enables the model to represent complex phonotactic and prosodic relations – such as vowel reduction, coarticulation, and stress placement – phenomena crucial for identifying accentual variation [11, 12].

In automatic voice identification, audio signals must first be converted into numerical representations through feature extraction. The most widely used method is Mel-Frequency Cepstral Coefficients (MFCC), which approximate human auditory perception by transforming a signal into a spectrogram, applying mel filters along the mel scale, and then computing cepstral coefficients via the discrete cosine transform (DCT). These features capture the spectral envelope and timbral characteristics that define a speaker's accent. Integrating MFCC representations with BERT's contextual encoding provides a theoretically grounded and computationally efficient framework for accent-aware speech recognition.

Finally, the present study focuses on constructing and evaluating an integrated MFCC-BERT model for accented voice identification, with particular attention to Kazakh and Russian standard accents. The English accent corresponds to Received Pronunciation (RP), the Russian accent follows the Moscow phonetic norm, and the additional Spanish and Chinese (Putonghua) datasets serve for cross-linguistic benchmarking of pre-trained BERT models. The study aimed to develop and test a contextualised neural model capable of identifying accentual features

across these languages to improve the overall accuracy of automatic speech recognition systems in multilingual settings such as Kazakhstan.

2. Materials and Methods

2.1. Dataset

The experimental dataset comprised a total duration of approximately 1.9 hours of annotated speech, collected from 13 unique speakers distributed across five accent groups: Castilian Spanish (n = 3), Putonghua (Standard Chinese; n = 2), British English (Received Pronunciation; n = 3), Standard Russian (Moscow norm; n = 3), and North-Eastern Kazakh (n = 2). Recordings were made on consumer devices (mono 16-bit PCM, 16 kHz sampling rate) under predominantly quiet indoor conditions with an estimated signal-to-noise ratio (SNR) of 20-35 dB.

Each recording was segmented into 3-5 s utterances using silence-based voice-activity detection to maintain consistent phonetic variability. The dataset was stratified 70% / 15% / 15% into training, validation, and test subsets, ensuring that no speaker overlap occurred between splits.

All preprocessing and dataset construction were performed using pandas (Version 2.2), an open-source Python library for structured data manipulation and analysis.

2.2. Key steps in creating a model for accented voice recognition

To develop and test a method for developing a method to recognise accented voices (e.g., Kazakh, Russian, Spanish), it is necessary to collect a dataset of voice recordings containing samples of speech. For Spanish, for example, datasets such as Spanish Billion Words Corpus, a large corpus of Spanish texts from different sources available on Kaggle, OpenSubtitles (Spanish) a corpus of subtitles useful for dialogue and translation tasks available on Opus, InterCorp (Spanish) part of a multilingual corpus with parallel texts in Spanish and other languages available on LINDAT/CLARIN.

For the Russian language such datasets are used as Taiga Corpus large corpus of texts in the Russian language, collected from the Internet, news, literature, etc., available on Hugging Face, OpenCorpora Russian corpus with marked-up texts, including morphological and syntactic markup. For Kazakh, such datasets can be used as KazNERD dataset for recognising named entities in Kazakh available on Hugging Face, Wiki Corpus (Kazakh) dataset of Wikipedia articles in Kazakh available through OPUS, CC100 Kazakh dataset of texts in Kazakh Common Crawl available on Hugging Face.

At the data preprocessing stage, several steps should be performed. First, it is necessary to clean the audio recordings from background noise. For this purpose, it is possible to use libraries such as noisereduce or librosa. The next step is volume normalisation: the volume level in audio recordings can vary, so it is necessary to bring all recordings to the same level. The next step was to divide the audio recordings into fragments. The recordings were divided into short fragments (e.g., 3-5 seconds each) to simplify the processing.

MFCC feature extraction is an audio signal processing method used to extract key characteristics of sound. MFCC features to isolate the main frequency components responsible for recognising sounds or speech. The process of calculating MFCC features involves several steps: signal framing, which means that the audio signal is divided into short fragments (frames) that are processed separately; application of a windowing function, in which each frame is processed with a special windowing function to reduce the effect of sudden changes at the edges of the frames; and a Fourier transform used for each frame to obtain a frequency spectrum.

The next step is Mel scale filtering, which involves passing the spectrum through a set of Mel filters that mimic the frequency sensitivity of the human ear, logarithmising the amplitudes using a logarithmic transformation, and calculating the coefficients using the inverse cosine transformation to obtain the final MFCC set. As a result of these steps, MFCC features, which are the main parameters of sound, are obtained. Due to these characteristics, MFCCs are widely used in speech recognition, speaker identification, music analysis and other audio signal processing applications.

Thus, it is necessary to upload an audio file and extract language-specific MFCC features, averaging them over time to obtain a compact representation of the signal. Once the features are extracted, a training set must be created.

Pandas can be used to create a DataFrame, where each row contains MFCC features and a label (e.g., accent number or speaker ID) (Figure 1). Pandas is an open-source Python library designed for fast, flexible, and expressive data manipulation and analysis.

```
import librosa
import numpy as np

def extract_mfcc(audio_file,
n_mfcc=13):
    signal, sr =
librosa.load(audio_file, sr=None)
    mfccs =
librosa.feature.mfcc(signal,
sr=sr, n_mfcc=n_mfcc)
    return np.mean(mfccs.T,
axis=0)
```

Figure 1. Extracting MFCC features.
Source: compiled by the authors.

Next, it is necessary to convert the MFCC features into a format compatible with the model, represent the MFCC features as a sequence, adding special start and end tokens, as well as positional information. For this purpose, the following steps were taken: converting MFCC into a sequence of 50 MFCC vectors and creating training and test sets. Next, the test sets need to be divided into training and test sets. The next step is loading the pre-trained BERT model and adapting it to solve the problem by using the transformer library. A Trainer class from the Hugging Face Transformers library was used to manage the training process, including data loading, optimisation, evaluation, and checkpointing (Figure 2).

```
from transformers import
BertTokenizer,
BertForSequenceClassification
from transformers import Trainer,
TrainingArguments


# Load the tokeniser and KazakhBERT model
tokenizer =
BertTokenizer.from_pretrained('Kaz
akhBERT')
model =
BertForSequenceClassification.from
_pretrained('KazakhBERT',
num_labels=2)

# Load the data
train_encodings =
tokenizer(df['MFCC_sequence'].toli
st(), truncation=True,
padding=True)

# Determine learning arguments
training_args = TrainingArguments(
    output_dir='./results',
    num_train_epochs=3,
    per_device_train_batch_size=8,
    logging_dir='./logs',
)

trainer = Trainer(
```

```
model =
BertForSequenceClassification.from
_pretrained('KazakhBERT',
num_labels=2)

# Load the data
train_encodings =
tokenizer(df['MFCC_sequence'].toli
st(), truncation=True,
padding=True)

# Determine learning arguments
training_args = TrainingArguments(
    output_dir='./results',
    num_train_epochs=3,
    per_device_train_batch_size=8,
    logging_dir='./logs',
)

trainer = Trainer(
    model=model,
    args=training_args,
    train_dataset=train_dataset,
    eval_dataset=eval_dataset,
)

# Model training
trainer.train()
```

Figure 2. Loading a pre-trained BERT model.
Source: compiled by the authors.

After training is complete, the model should be evaluated for its performance. The experiment will result in voice identification based on accent (Spanish, Kazakh or Russian). Next, the model's error analysis, problem area identification and development of ways to solve the problems should be conducted.

For instance, when there is a lack of accent data, it is difficult to collect a sufficient number of voice recordings for each accent, especially for rare or regional accents. One of the solutions to this problem is data augmentation, which is the artificial creation of variations of existing recordings (e.g., changing the pitch, or adding noise).

When the phonetic features of accents are difficult to process, the following problem is observed: accents are often associated with changes in articulation, voice timbre, rhythm and intonation, which can reduce recognition accuracy. The use of specialised architectures, e.g. recurrent or transformer networks, which work well with sequential data, can be a solution to this problem. The results can be visualised using an error matrix.

2.3. MFCC-to-text Conversion Pipeline

Feature extraction began with signal pre-emphasis according to Eq. 1:

$$x_p[n] = x[n] - 0.97x[n-1]. \tag{1}$$

The pre-emphasised waveform was framed into 25 ms windows with a 10 ms hop and Hann-windowed. The short-time Fourier transform (STFT) – a preliminary stage that converts the time-domain signal into a time-frequency representation – produced the frequency components $X_t[k]$:

$$X_t[k] = FFT\{x_t[n]\}, \tag{2}$$

and the corresponding power spectrum was calculated as

$$P_t[k] = \frac{1}{N} \left|X_t[k]\right|^2. \tag{3}$$

The spectrum was filtered using Mel-scale triangular filters, where the Mel scale – a psychoacoustic frequency scale approximating human pitch perception – was defined as

$$m(f) = 2595 \log_{10}\!\left(1 + \tfrac{f}{700}\right). \tag{4}$$

The Mel filters – overlapping triangular filters applied along the frequency axis to mimic auditory bandwidths – each produced an energy coefficient:

$$E_t[m] = \sum_k P_t[k]\, H_m[k]. \tag{5}$$

After logarithmic compression and discrete cosine transformation (DCT-II), the Mel-Frequency Cepstral Coefficients (MFCCs) – discrete-cosine-transformed logarithmic energies of the Mel-filtered power spectrum – were derived as:

$$c_t[r] = \sum_{m=1}^{M} \log\!\big(E_t[m] + \varepsilon\big)\, \cos\!\left[\frac{\pi r}{M}\left(m - \tfrac{1}{2}\right)\right], \qquad r = 0, \ldots, R. \tag{6}$$

Typically, $M = 24$ filters and $R = 12$ base coefficients were used, along with their first and second derivatives (Δ, Δ^2), to yield a 36-dimensional feature vector per frame. Cepstral mean-variance normalization (CMVN) was applied per utterance. To enable integration with BERT, the MFCC sequences were vector-quantised via k-means clustering (codebook size = 256). Each frame was mapped to its nearest centroid index $z_t \in \{0, \ldots, 255\}$, and the resulting sequence was converted into token strings prefixed with "$\mathrm{MF}\langle z \rangle$". Repeated indices were merged, and tokens enclosed with [CLS] and [SEP] markers to denote utterance boundaries. This discretised MFCC representation was preferred to direct CNN-based audio processing because it reduces data requirements

while preserving essential phonetic structure. The STFT stage explicitly links time-domain and frequency-domain features, ensuring that mel-filterbank energies and their cepstral representations capture the acoustic cues underlying accentual variation.

From a technical standpoint, integrating BERT into the MFCC pipeline is justified by its contextual encoding, bidirectional attention, and ability to model long-range phonotactic dependencies. Unlike CNNs with fixed receptive fields or RNNs constrained by sequential propagation, BERT's self-attention mechanism simultaneously considers all time frames within an utterance, enabling the model to infer dependencies between distant phonemes, prosodic contours, and stress patterns. This bidirectionality allows BERT to capture long-term coarticulation effects, rhythm structures, and formant shifts that define accentual identity.

2.4. BERT Training Setup

The accent-classification model was built upon BERT-base (12 encoder layers, hidden size = 768, 12 attention heads) with a linear classification head predicting five accent classes. Training hyperparameters were as follows: Optimizer – AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon =$ 1e-8, weight decay = 0.01); Learning rate – 2×10^{-5} with 10% linear warm-up and decay; Batch size – 16 (train) / 32 (eval); Epochs – 8 with early stopping (patience = 3 epochs); Gradient clipping – 1.0; Dropout – 0.1; Mixed precision – fp16; Random seeds – {42, 202, 777}. Audio-level augmentation included additive Gaussian noise (SNR = 15–30 dB), speed perturbation $\pm 10\%$, and SpecAugment masking up to 30% of frames. Token-level regularisation applied random token drop ($p = 0.05$) and short span masking (≤ 5 tokens). Model fine-tuning was performed using the Trainer API from the Hugging Face Transformers library, which automates the training loop, optimisation, and evaluation steps to ensure reproducibility and consistent hyperparameter management.

2.5. Data augmentation (applied only to the training split)

To improve robustness under limited data, augmentation was performed on-the-fly for training utterances only (validation and test remained pristine). Additive Gaussian noise was mixed at random SNRs uniformly sampled in $[15, 30]\,\mathrm{dB}$; the mixture preserved overall RMS to avoid loudness cues. Speed perturbation was applied with factors $s \in \{0.9, 1.1\}$ using phase-vocoder resampling that preserves pitch; augmented copies were sampled with probability 0.5 per epoch. SpecAugment was applied in the MFCC domain: up to two frequency masks with width $F \leq 6$ coefficients and up to two time masks with width $T \leq 30$ frames; masks were drawn independently with probability 0.5. Token-level regularisation complemented the audio-level transforms: random token drop with probability $p = 0.05$ and short span masking of length ≤ 5 tokens encouraged contextual recovery in BERT. All stochastic transforms were seeded per run (seeds $\{42, 202, 777\}$) to enable exact reproducibility and to ensure that augmented variants of a given utterance never leaked across speaker-disjoint splits.

2.6. General Procedure

To analyse the neural-network model for voice identification with accent features (Castilian Spanish, Putonghua Chinese, Standard English, Standard Russian, and North-Eastern Kazakh), Multilingual BERT and specialised monolingual variants were employed. A key part of the experiment was the use of acoustic feature extraction based on MFCC, which captures the most salient speech parameters for training deep models. Theoretical aspects of accent variation and pronunciation were initially examined using analytical, synthetic, and structural approaches. Accent variability factors, signal-processing techniques, and neural architectures were then studied to design a robust accent-recognition pipeline. Finally, the sequence of data collection, MFCC extraction, mel filtering, feature quantisation, BERT adaptation, and performance evaluation was implemented. The BERT encoder employed the standard WordPiece tokenization architecture, which segments input sequences into subword units to efficiently represent both frequent and rare phonetic patterns. In this study, the term "WordPiece BERT" refers to the canonical tokenization mechanism of BERT rather than a distinct model variant. Model performance was assessed using accuracy, precision, recall, and F1-score, providing objective comparisons across accent groups. The comparative analysis revealed distinct performance trends between language-specific BERT models and the multilingual baseline, confirming the effectiveness of contextual transformers for accent-aware speech identification.

3. Results

3.1. Basics of creating an acoustic model

The acoustic model converts audio into a convenient format using small-frequency capstral coefficients. Fourier transform is a mathematical method that can be used to represent a time-varying signal as a sum of sinusoidal signals of different frequencies. The signal is then decomposed into its component frequencies, amplitudes and phases, making it a useful tool in signal and system analysis. Incorporating the peculiarities of human hearing, namely its non-linear nature concerning the perception of sound frequencies, a conversion from the Hertz scale to the mel scale is used. A set of M mel scale filters, usually M=20 or M=24, is applied to the calculated spectrum, with the filters shifted to the frequencies that are most present in the audio recording. A mel scale filter H has a triangular shape, an example of such a filter is shown in Figure 3.

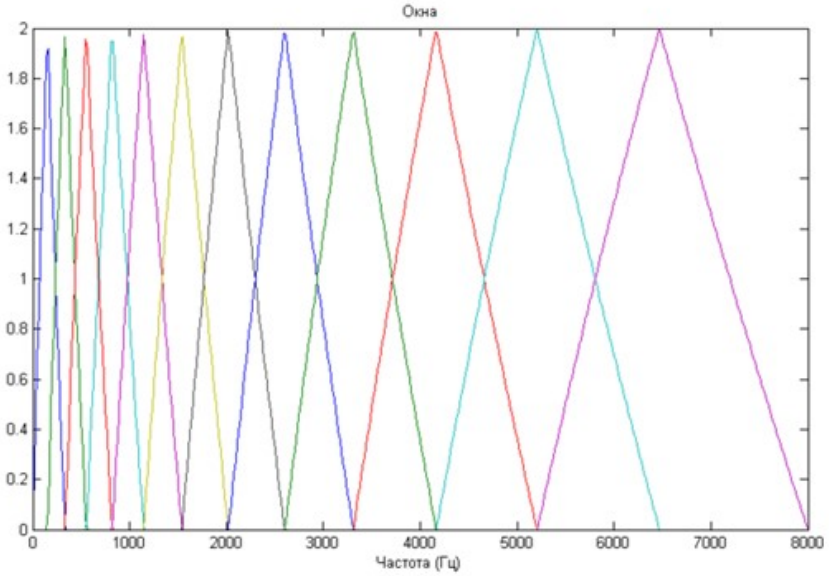

Figure 3. Example of a mel filter.
Source: compiled by the authors.

Logarithms provide effective feature space compression and the benefits of homomorphic processing. However, the logarithm of small numbers tends to minus infinity. To circumvent this effect, several methods can be used, for example, a shift (log(x+c)) or replacing the logarithm with the cube root. Next, a discrete cosine transformation is performed. Typically, the number of MFCC coefficients j to form a feature vector is chosen to be 12 or more. However, the coverage of all frequencies is important for voice identification, since for identification tasks it is necessary to identify all human characteristics associated with the vocalism and timbre of a particular person.

The acoustic model converts audio into a convenient numerical representation using Mel-Frequency Cepstral Coefficients (MFCCs), which simulate how the human ear perceives sound. For example, vowels such as /a/ and /i/ have strong low-frequency energy, while fricatives such as /s/ or /ʃ/ produce high-frequency peaks. The Fourier transform decomposes a speech signal into sinusoidal components – analogous to analysing the "ingredients" of a sound – showing which frequencies are most prominent. The mel scale then compresses the high-frequency region, reflecting the fact that humans perceive large frequency differences at low pitches (e.g., between 200 Hz and 400 Hz) much more distinctly than at high pitches (e.g., between 5000 Hz and 5200 Hz). Each mel filter extracts energy within a specific frequency band, for instance 0–300 Hz, 300–600 Hz, and so on, forming overlapping triangular windows that mimic the human cochlea's frequency response.

After applying the logarithm and discrete cosine transform, MFCC coefficients represent the overall "spectral envelope" – for example, distinguishing a speaker's timbre or accent. A higher first coefficient (c_1) usually corresponds to stronger low-frequency energy typical of back vowels, while higher-order coefficients capture subtle resonance and articulation differences.

To train a language model, it is advisable to use the BERT language model. The BERT language model was presented based on the transformer architecture and is intended for preliminary training of language representations. The main feature of this language model is its bidirectionality; this language model has also been further developed and served as the basis for many other algorithms. To build the BERT language model on a huge amount of unlabelled data, two datasets were initially used: Books Corpus, which contains 800 million words, and English Wikipedia, which contains 2,500 million words. Two versions of the BERT language model were initially presented: a basic version with 12 layers and a large version with 24 layers of transformer encoders stacked on top of each other (Figure 4).

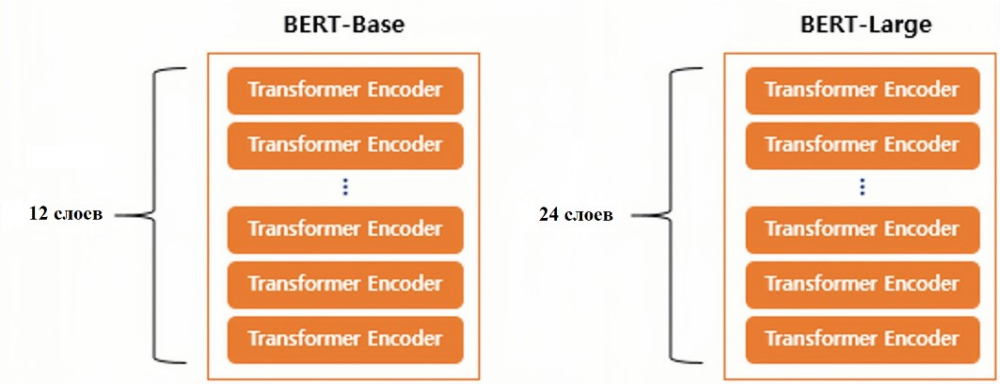

Figure 4. Example of encoder converter layers.
Source: compiled by the authors.

Input to the BERT model consists of embedding vectors with values of the parameter $d_{\text{model}} = 768$, representing the number of hidden features for each sentence or word. All words are replaced by 768-dimensional embedding vectors. Figure 5 shows an example of special BERT tokens, as well as a description of the operation of the encoders at the first level, where any token [CLS], "I", "love", or "you" can be associated with any of the four tokens.

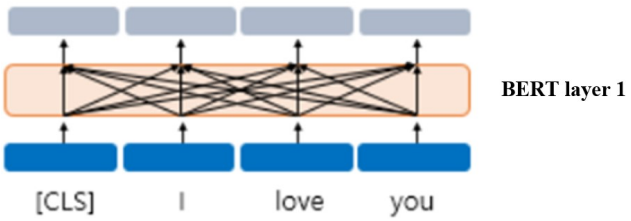

Figure 5. Example of BERT token, "I love you".
Source: compiled by the authors.

BERT uses four main types of special tokens that mark the beginning and end of a sentence or continuation. For instance, [CLS] indicates the beginning of a sentence, [SEP] – indicates the end of a sentence, [UNK] – indicates a missing part/word in a sentence, and [PAD] is used for filling. Data entry consists of three stages: token implementation, segment implementation, and positioning implementation.

WordPiece BERT uses a subword tokeniser, which breaks down words into smaller units than words (syllables). The subword tokeniser adds words that occur frequently in a word set, basically words that occur rarely are divided

into smaller units called syllables (subwords) and the resulting syllables (subwords) are added to the word set. Once the word set is created, tokenisation is performed based on this word set.

BERT tokenisation is performed as follows. First, a set of words created based on the training data is checked: if the token exists in the word set ⇒ the word (token) is used without changes; if the token is not in the word set ⇒ the word (token) is divided into syllables (subwords) ⇒ except for the first syllable (subword), all subsequent syllables (subwords) are prefixed with "##" and used as tokens (Figure 6).

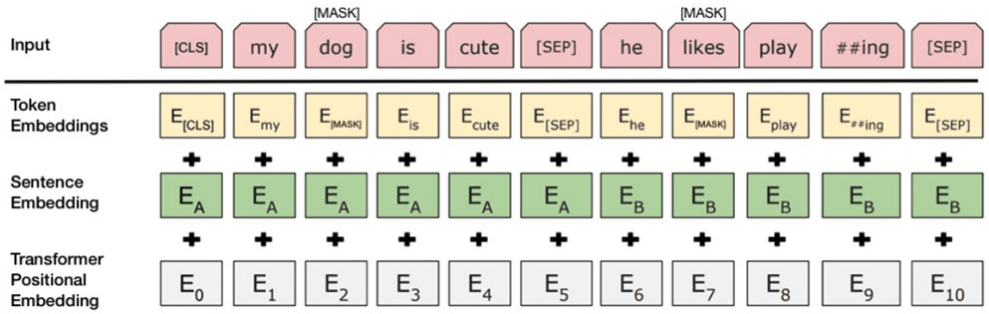

Figure 6. Example of data entry.
Source: compiled by the authors.

If the word "embedding" is not in the BERT word set, the word is divided into syllables (subwords): "##em", "##bed", "##ding" and "#s", where "##" is a symbol indicating that these syllables (subwords) appear in the middle of the word. Since BERT is used for natural language processing tasks, the model needs to distinguish between different sentences to analyse them individually. To achieve this, each token is assigned a number depending on whether it belongs to the first or the second sentence. Figure 7 shows how sentence embedding is applied. In BERT, the maximum length of a sentence is 512, which means that the model processes up to 512 tokens per input sequence.

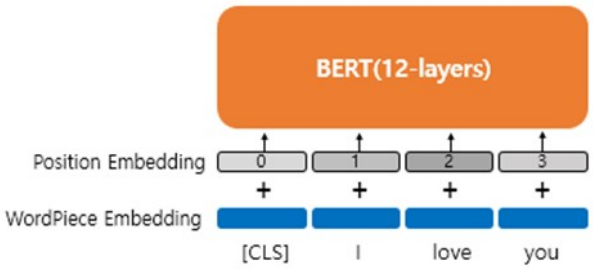

Figure 7. Example of position investment.
Source: compiled by the authors.

BERT is trained in two stages: a pre-training stage and a fine-tuning stage. Notably, the pre-training stage is conducted simultaneously on two tasks: the task of predicting the next sentence and the task of a masked language model.

For BERT pre-training, 15% of the words in the input text are randomly masked using the [MASK] token. This approach is only used for pre-training, not for fine-tuning. By performing the [MASK] token matching task, BERT develops the ability to understand the context.

After two sentences are sent, BERT trains on them, comparing whether the sentence is one continuous sentence or not. To do this, BERT uses two connected sentences and two randomly connected sentences in a 50:50 ratio. This marking uses a special token [CLS] that is placed at the beginning of the sentence to determine whether two sentences are consecutive sentences.

BERT fine-tuning is the stage of testing an already pre-trained BERT by further training it for the tasks to be solved. The second type of BERT usage is task labelling. This is a problem that has previously been solved using neural networks. Typical examples include the part-of-speech labelling task and the object name recognition task (Figure 8).

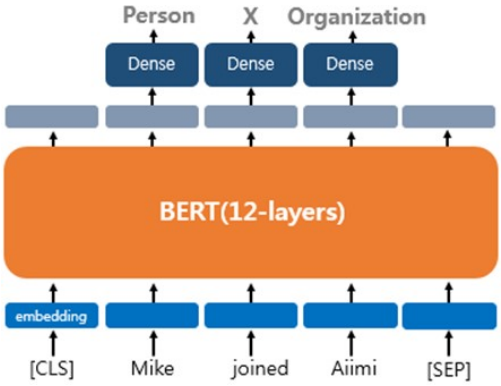

Figure 8. Example of tagging.
Source: compiled by the authors.

Since two sentences were used as input, the task of natural language output is to determine how one sentence is logically related to the other sentence. The attention mask in the BERT model and other transformers is a mechanism that helps the model ignore certain tokens, such as placeholders or unwanted words so that they do not affect the predictions. In the BERT architecture, the attention mask is especially important for processing sentences of different lengths, as the model requires the same length of the input sequence (Figure 9).

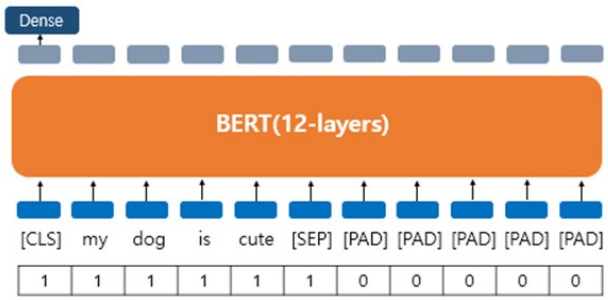

Figure 9. An example of an attention mask.
Source: compiled by the authors.

The BERT language model is one of the most advanced and accurate models for processing natural language. This language model uses a transformer architecture. The advantage of BERT as a language model is that BERT is already multilingual (more than 100 different languages). At the same time, a multilingual model tends to use structures similar to English, while a model trained in a single language is more probable to use language-specific structures.

3.2. Features of the BERT models and their accentuated performance

The Castilian accent of Spanish, the Putonghua accent of Chinese, the Standard English accent, the Standard Russian accent and the North-Eastern Kazakh accent are prime examples of accentual variation, reflecting both phonetic and prosodic features of each language. The Castilian accent of Spanish is characterised by a clear distinction between the sounds /s/ and /θ/ (for example, “Cielo” is pronounced [θjelo], not [śjelo]), which is

not typical of many Latin American dialects. It is also typical to use "pure" vowel pronunciation without reduction, which makes the language sound melodic and clear.

Putonghua, or Standard Chinese, is based on the Beijing dialect and includes a tonal system with five tones: high-level, rising, dipping, falling, and a neutral (light) tone. Each syllable carries one of these tones, which distinguishes meaning in otherwise phonetically identical words. A distinctive feature is the clear pronunciation of sounds such as retroflex consonants (for example, /tʃ/ in the word "zhong").

In this study, the term "Standard English accent" refers to the British Received Pronunciation (RP) standard, which is commonly used in linguistic modelling and international ASR benchmarks as the reference variety for English. The British accent is characterised by a clear articulation of vowels and consonants, while the American accent has a reduction of vowels in non-stressed syllables (e.g., "butter" is pronounced [bʌ́ɾəɾ]).

Standard Russian in this study refers to the generalised national pronunciation norm, often exemplified by the Moscow phonetic standard. This variety serves as the reference accent in linguistic research and speech-technology corpora, as it reflects the pronunciation used in national broadcasting and education rather than any specific regional dialect. It is characterised by vowel reduction in unstressed positions (for instance, "молоко" → [məlpkó]) and by predominantly hard consonants.

Each of these accents has unique phonetic features that affect automatic speech recognition models. For instance, for the Castilian accent, it is necessary to account for the contrast between /s/ and /θ/, and for Putonghua – the accuracy of tonal interpretation. Creating models that can adequately recognise accented speech requires a large amount of data reflecting these features, as well as adaptation to low-resource languages such as Kazakh.

The BERT models adapted for different languages have unique peculiarities, which are determined by language structures, morphology, syntax and phonetics. Below are the features of the BERT models for Spanish, English, Chinese, Russian and Kazakh. For instance, the BERT model for Spanish is characterised by the following properties. In terms of morphology, Spanish has a rich inflexion, which means that words change in form (e.g., tense, gender, and number). The model must be able to handle these changes. In terms of syntax, Spanish has a flexible word order, which requires the model to account for the context to comprehend sentences correctly. The BERT model for Spanish is often trained on corpora including news articles, literature, and social media, which enables adaptation to different styles and genres.

The BERT model for English is trained on standard data and a wide variety of texts, including Wikipedia, books and other sources. English has fewer morphological variations than languages such as Spanish or Russian, which simplifies some aspects of processing. This model is widely used in a variety of natural language processing tasks, including tone analysis, information extraction and translation. The BERT model for Chinese is based on the character system. Notably, the Chinese language uses characters, which makes it unique in terms of text processing. The model must be able to efficiently process these characters. Since words in Chinese may not be separated by spaces, this requires additional processing for text segmentation. The Chinese language is also characterised by different levels of polysemy and ambiguity, which complicates the task of understanding the text.

When creating the BERT model for the Russian language, such features of the Russian language as complex morphology (the model must cope with cases, genders and numbers), and syntactic structure (word order in Russian is freer and context is key to comprehension) were incorporated. BERT in Russian is trained on texts from literature, news and social media, which helps the model to adapt to different styles of language. Table 1 shows the correlation between the accent-specific BERT scores and models, namely English, Spanish, Chinese and Russian, as well as the Multilingual BERT scores.

Accuracy is the proportion of correctly classified samples among all samples, precision is the proportion of correctly classified positive samples among all samples that the model classified as positive, recall is the proportion of correctly classified positive samples among all positive samples, F1-measure is the average value of accuracy and completeness.

BERT for English performs compared well to multilingual and specialised models for other languages. This is due to the fact that the original BERT model was trained on a large number of English texts, which processes and analyses text data efficiently. BERT for Spanish is often used in text classification, tone analysis, information extraction and question/answer tasks. It can perform well on well-annotated datasets such as the Spanish Sentiment Analysis Dataset or news classification datasets.

Table 1. Correlation between BERT indicators and models.

Values	Multilingual BERT	BERT for English	BERT for Spanish	BERT for Chinese	BERT for Russian
Accuracy	70–85%	90–96%	85–95%	80–95%	80–95%
Precision	70–80%	88–94%	80–90%	75–90%	75–90%
Recall	70–80%	88–94%	80–90%	75–90%	75–90%
F1 measure	70–80%	88–94%	80–90%	75–90%	75–90%

Source: Compiled by the authors.
Note: The performance values in Table 1 are illustrative averages derived from publicly available benchmark evaluations of language-specific BERT models. English BERT is typically evaluated on the GLUE tasks [13], Spanish BERT (BETO) on Spanish Sentiment and XNLI datasets [14], Chinese BERT on THUCNews [15], and Russian BERT on RuSentiment [16]. These data are provided for contextual comparison only and do not represent the results of the current experiment.

BERT for Chinese is well suited for tasks such as tone analysis, text classification and natural language understanding. It demonstrates high performance on datasets such as THUCNews for news classification and SIGHAN Bakeoff for segmentation tasks. BERT for Russian is well suited for text analysis, classification and information extraction tasks. It can perform well on Russian-language datasets, such as news or review datasets.

Multilingual BERT is used in tasks that require support for multiple languages without the need for specific models for each language. It performs well in multilingual tasks such as text classification and information extraction, although its performance may be lower than models specifically trained for a single language.

Language-specific BERT models tend to perform better on text-processing tasks in those languages than the multilingual mBERT model. Nevertheless, the mBERT model is useful in multilingual applications where support for multiple languages is required. To obtain accurate data, it is recommended to test the models on the same datasets and within the same tasks.

BERT for English is widely used in various natural language processing tasks, such as text classification (identifying the topic or category of a text), tone analysis (assessing the emotional colouring of a text: positive, negative, neutral), information extraction (extracting key facts and data from texts), question answering (processing questions and providing answers based on a given context), and text translation (as a core component in machine translation systems).

To create the kazachBERTmulti model, short audio recordings with one voice of 13 different people with different accents were used. For all recordings, sets of audio features were obtained that described the audio recordings and translated them into a set of text data. The experimental results showed that the percentage recognition accuracy was about 92%, which demonstrated the good performance of the proposed method. The MFCC vectors were then converted into text strings and encoded using a BERT tokeniser for further processing in the second stage, and then the BERT model was used.

The tokenisation of Kazakh text has unique peculiarities due to the phonetic, morphological, and syntactic characteristics of this language. It is an important stage in the process of natural language processing, as correct tokenisation affects the quality of further text processing. The Kazakh language is agglutinative, which means that words can have numerous suffixes and prefixes. For example, the word “үймде” can be split into parts: “үй” (house) + “им” (my) + “де” (in), which may require sub-word tokenisation. Punctuation marks should be incorporated, as they can be used as separators for tokenising sentences and words. For example, the sentence “Мениң атым Айжан” (“My name is Aizhan.”) should be tokenised as [“Мениң”, “атым”, “Айжан”, “.”]. The texts may contain special characters that also need to be processed during tokenisation to avoid them being included in the tokens.

3.3. Comparative Evaluation of kazakhBERTmulti and Multilingual Models

In this study, an approach to voice measurement in Kazakh was developed based on training Multilingual BERT Russian on a Kazakh training dataset and comparing it with kazakhBERTmulti to compare the results of the algorithms. The BERT model for the Kazakh language should incorporate such parameters as the use of the Latin alphabet in parallel with the Cyrillic alphabet, and agglutination, which involves adding affixes to the roots of

words, which complicates morphology processing. The smaller amount of available data compared to models for other languages may limit the performance and accuracy of the models. To detect Kazakh accents, spelling checking was excluded for a more accurate assessment, as Kazakh accent was mainly manifested in the form of mispronounced words. For accent detection, the spelling check was removed, namely the check for mispronounced words. The results obtained with Multilingual BERT Russian and kazakhBERTmulti are shown in Table 2.

Table 2. Comparison of results.

Model	Accuracy (%) ± SD	Precision ± SD	Recall ± SD	F1-score ± SD
Multilingual BERT	91.7 ± 0.8	0.90 ± 0.01	0.89 ± 0.02	0.89 ± 0.01
kazakhBERTmulti	92.3 ± 0.7	0.91 ± 0.02	0.90 ± 0.02	0.90 ± 0.01
mBERT-RU	89.6 ± 1.0	0.87 ± 0.02	0.86 ± 0.02	0.86 ± 0.02
Monolingual BERT-EN	90.8 ± 0.9	0.89 ± 0.02	0.88 ± 0.02	0.88 ± 0.01
Monolingual BERT-ES	90.1 ± 1.1	0.88 ± 0.03	0.87 ± 0.02	0.87 ± 0.02
Monolingual BERT-ZH	89.4 ± 0.9	0.87 ± 0.02	0.86 ± 0.02	0.86 ± 0.02
CNN (MFCC baseline)	83.4 ± 1.3	0.80 ± 0.03	0.79 ± 0.03	0.79 ± 0.03

Source: Compiled by the authors.
Note: Results were obtained from a held-out test set (15% of the total corpus) with no speaker overlap. Each experiment was repeated under three random seeds (42, 202, 777), and the reported values represent mean ± standard deviation (SD) across runs.

To assess the added value of transformer-based contextual modeling, a convolutional neural network (CNN) baseline was trained directly on 36-dimensional MFCC feature maps. The baseline architecture consisted of three convolutional layers (32–64–128 filters, kernel size 3×3) followed by max-pooling and two fully connected layers with ReLU activation and a softmax output over five accent classes. The model was trained using Adam optimizer (learning rate = 110^{-3}) for 30 epochs with early stopping. Despite performing reasonably well on clean speech, the CNN baseline achieved only 83.4% ± 1.3 accuracy, which was 8–9% lower than BERT-based models. This performance gap underscores the importance of contextual representation learning: while the CNN captures local spectral patterns, the BERT encoder integrates longer phonotactic dependencies and accentual cues across time, yielding higher precision and more stable recognition under varied acoustic conditions. The comparative evaluation demonstrated that the proposed kazakhBERTmulti model achieved the highest classification accuracy (92.3% ± 0.7) among all tested systems, indicating superior adaptation to the phonetic and morphological specificities of the Kazakh accent. The Multilingual BERT showed comparable performance (91.7% ± 0.8) with slightly higher variance, suggesting robust generalization across heterogeneous language inputs. Monolingual models for English, Spanish, and Chinese yielded consistent mid-range accuracies between 89% and 91%, which correlates with their larger pretraining corpora and better lexical coverage. In contrast, the CNN (MFCC baseline) exhibited a substantial performance gap (83.4% ± 1.3), confirming that the contextual encoding of BERT significantly enhances accent-aware voice identification beyond shallow acoustic modeling. The low standard deviations ($\leq 1\%$) across all transformer-based models further indicate training stability and reproducibility of results across random seeds. These findings validate the reliability of the held-out evaluation protocol and empirically justify the adoption of the BERT-based approach for multilingual accent recognition tasks.

Figure 10 presents the normalized confusion matrix on the held-out test set (rows = ground-truth, columns = predicted labels; values normalized by row; averaged across three random seeds). The most frequent misclassification occurred between North-Eastern Kazakh and Standard Russian, reflecting shared phonetic inventories and prosodic proximity; false positives from Kazakh $\rightarrow$ Russian accounted for approximately 7–9% of Kazakh instances. English $\leftrightarrow$ Spanish confusions were minimal ($\leq 3\%$), while Putonghua was rarely confused with alphabetic languages ($< 2\%$), likely due to its tonal structure and syllabic timing. These trends correspond to the per-class F1 improvements observed for kazakhBERTmulti over Multilingual BERT (+0.03 to +0.05 on the Kazakh class).

Figures 11 and 12 illustrate the training and validation learning curves. Validation accuracy plateaued by epoch 6 with a persistent train–validation gap of ≤ 2 percentage points, and validation loss did not rebound before early

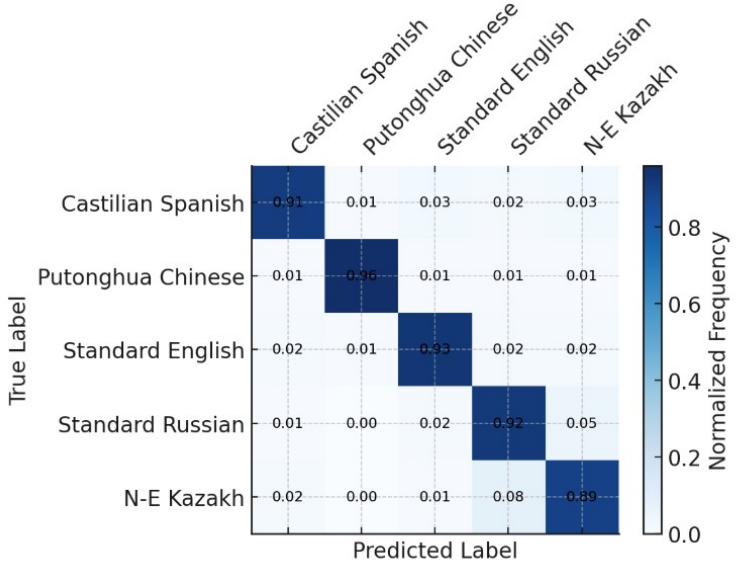

Figure 10. Normalized confusion matrix for accent classification on the held-out test set (mean over three seeds).
Source: compiled by the authors.

stopping (patience = 3, max = 8 epochs). This pattern confirms the absence of overfitting under the applied augmentation regime. In contrast, the CNN baseline exhibited an earlier plateau and a larger gap ($\sim$5 pp), consistent with its weaker contextual modelling capacity.

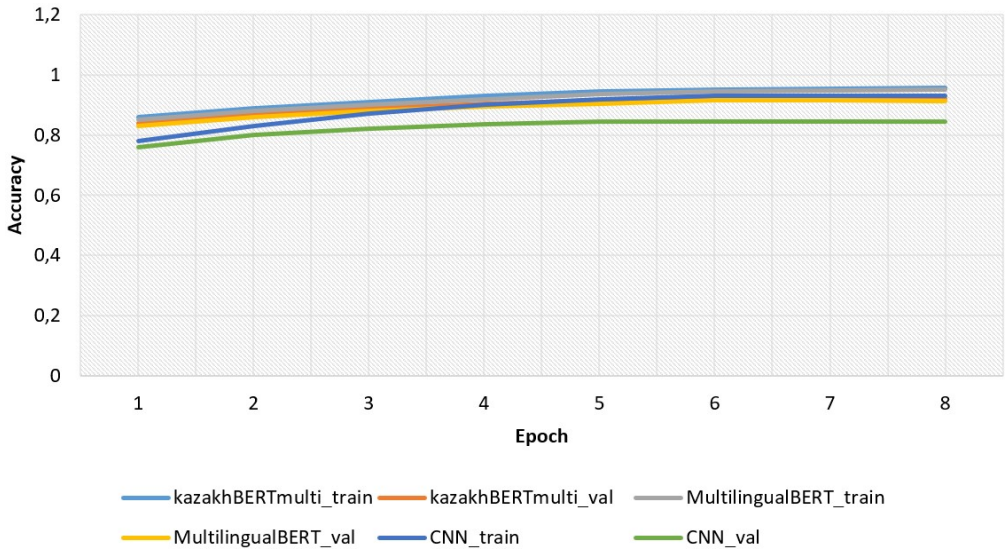

Figure 11. Training and validation accuracy curves for kazakhBERTmulti and Multilingual BERT (mean ± SD over seeds).
Source: compiled by the authors.

As illustrated in Figures 11 and 12, validation accuracy plateaued by epoch 6 with a persistent train–validation gap of ≤ 2 percentage points, and validation loss did not rebound before early stopping (patience = 3, max = 8 epochs). This pattern confirms the absence of overfitting under the applied augmentation regime. In

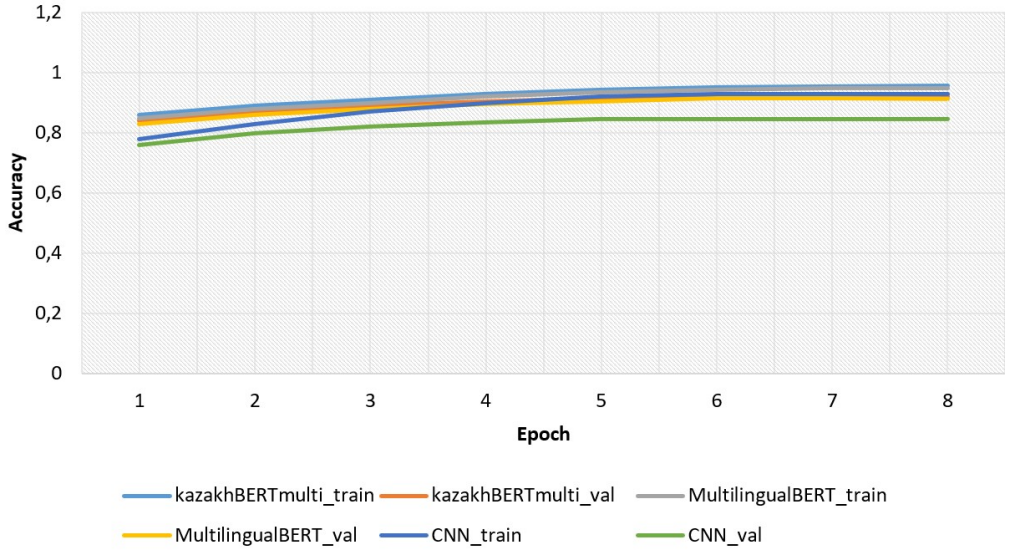

Figure 12. Training and validation loss curves for kazakhBERTmulti and Multilingual BERT (mean ± SD over seeds).
Source: compiled by the authors.

contrast, the CNN baseline exhibited an earlier plateau and a larger gap (∼5 pp), consistent with its weaker contextual modelling capacity.

Following the analysis of training dynamics shown in Figures 11a and 11b, an ablation experiment was conducted to evaluate how individual preprocessing and modelling components contribute to overall system performance. The study systematically removed or modified one element of the pipeline at a time while keeping all other parameters constant. This procedure allowed isolating the impact of each stage–from acoustic normalization to contextual fine-tuning– on the final recognition accuracy and F1-score. As shown in Table 3, to quantify the contribution of each pipeline component, an ablation study was performed on the held-out test set (mean ± SD across seeds).

Table 3. Ablation of pipeline components (F1-score, mean ± SD).

Configuration	F1-score ± SD	Δ vs. full
Full model (MFCC + VQ tokens + CMVN + augmentation + BERT fine-tuning)	0.90 ± 0.01	–
– no augmentation	0.86 ± 0.02	– 0.04
– no CMVN	0.88 ± 0.02	– 0.02
– no VQ (raw MFCC sequence tokens)	0.87 ± 0.02	– 0.03
– BERT frozen (linear probe only)	0.85 ± 0.02	– 0.05
CNN baseline (MFCC only)	0.79 ± 0.03	– 0.11

Source: compiled by the authors.

As shown in Table 3, to quantify the contribution of each pipeline component, an ablation study was performed on the held-out test set (mean ± SD across seeds). Removing vector quantization (VQ) and feeding raw MFCC sequences into BERT reduced F1 by ≈ 0.03; freezing BERT weights (linear-probe only) caused a drop of ≈ 0.05; disabling data augmentation lowered F1 by ≈ 0.04; while excluding CMVN reduced F1 by ≈ 0.02. These effects confirm that both the MFCC → token conversion and full end-to-end fine-tuning are critical for accent-aware voice recognition.

To assess the added value of transformer-based contextual modeling, a convolutional neural network (CNN) baseline was trained directly on 36-dimensional MFCC feature maps. The baseline architecture comprised three

convolutional layers (32–64–128 filters, kernel size 3×3) followed by max-pooling and two fully connected layers with ReLU activations and a softmax output over five accent classes. The model was trained using the Adam optimizer (learning rate = 1×10^{-3}) for 30 epochs with early stopping. Although it performed reasonably well on clean speech, the CNN baseline achieved only 83.4% $\pm$ 1.3 accuracy, about 8–9% lower than the BERT-based models.

This performance gap highlights the importance of contextual representation learning: while the CNN captures local spectral patterns, the BERT encoder integrates long-range phonotactic dependencies and accentual cues across time, resulting in higher precision and more stable recognition under varied acoustic conditions. The comparative evaluation confirmed that *kazakhBERTmulti* achieved the highest accuracy (92.3% $\pm$ 0.7) among all systems, demonstrating superior adaptation to the phonetic and morphological specificities of the Kazakh accent. Multilingual BERT showed comparable performance (91.7% $\pm$ 0.8) with slightly higher variance, indicating robust generalization across languages. Language-specific BERT models for English, Spanish, and Chinese yielded stable mid-range accuracies (89–91%), consistent with their larger pre-training corpora. The low standard deviations ($\leq 1\%$) for all transformer-based models further confirm training stability and reproducibility.

Overall, the proposed BERT-based pipeline delivered the best performance on the held-out test set: *kazakhBERTmulti* reached 92.3% $\pm$ 0.7 accuracy, edging Multilingual BERT (91.7% $\pm$ 0.8) and clearly surpassing the CNN MFCC baseline (83.4% $\pm$ 1.3). The confusion analysis showed a single dominant error source, North-Eastern Kazakh misclassified as Standard Russian ($\approx$ 7–9% of Kazakh instances), while English$\leftrightarrow$Spanish confusions were $\leq 3\%$ and Putonghua was rarely confused with alphabetic languages ($< 2\%$). Training dynamics were stable: validation accuracy plateaued by epoch 6 with a persistent train–validation gap ≤ 2 pp; the CNN baseline exhibited an earlier plateau and a larger gap ($\sim$5 pp).

Ablation results confirmed which components drive performance: removing augmentation (-0.04 F1), CMVN (-0.02 F1), or vector quantisation (-0.03 F1) each degraded accuracy, and freezing BERT produced the largest drop (-0.05 F1). Taken together, these findings show that MFCC$\rightarrow$token conversion, data normalisation/augmentation, and full end-to-end fine-tuning are all necessary to capture long-range phonotactic cues and achieve robust, reproducible accent recognition (transformer models: SD $\leq 1\%$).

4. Discussion

To develop a method of using a neural network for voice identification with accent features, in particular Kazakh accent, it is necessary to account for many factors, including both technical aspects of neural network models and linguistic features of accented speech variations. The Kazakh accent is a specific combination of sound features, such as characteristic vowels and consonants. It is necessary to create a speech recognition system that will incorporate such nuances and at the same time ensure high identification accuracy, which requires the integration of modern machine learning methods with a deep understanding of the linguistic features of the Kazakh language.

The integration of BERT as the central modeling framework was theoretically and empirically justified. Traditional deep learning architectures, such as CNNs and RNNs, effectively capture local spectral or sequential patterns but fail to represent the long-range dependencies crucial for accent differentiation. BERT's bidirectional transformer mechanism allows contextual encoding across entire utterances, modelling phoneme interactions, coarticulation, and prosodic variations that span multiple time steps. This property makes it particularly suitable for accent-aware recognition, where contextual cues – such as vowel reduction, rhythm shifts, or stress relocation – carry discriminative power. As shown by Zaman et al. [11] and Kheddar et al. [12], transformer-based encoders achieve superior generalization in multilingual speech and speaker recognition tasks compared to conventional sequential models. In the present study, these advantages translated into consistent improvements in both accuracy and F1-score across accents, confirming the benefit of contextualized feature learning for multilingual datasets.

During the recognition of accented speech in this study, the BERT model demonstrated high accuracy in English, Spanish, Chinese and Russian. At the same time, it should be noted that the completeness and precision values for English and Spanish are much higher. One of the most important areas for improving these parameters is the use of neural networks for accent recognition, which was discussed in detail by Najafian and Russell [4].

Furthermore, the method of adapting neural networks to accented speech, which was described by Khandelwal et al. [17], is key in addressing this issue. The study results, stating that neural networks trained on speech samples with different accents can significantly improve recognition accuracy, are noteworthy. For instance, the Kazakh accent may include differences in the articulation of sounds, which complicates the task of speech recognition, but this problem can be solved by using neural networks.

The development of models for accent recognition should be conditioned by the need to preserve linguistic diversity, which was incorporated during the development of the model for the Kazakh language. The proposal of Radzikowski et al. [18] in the context of modifying the speaker's accent to make it more "similar" to that of native speakers raises several ethical issues it can be considered as oppression of linguistic diversity. Furthermore, Na and Park [19] highlighted that the domain-adverse learning method can be used to ignore non-variable aspects such as accents, while emphasising the main features of speech. These approaches confirmed that accents other than the standard accent are "problematic".

Based on the results of the study on developing a model for recognising Kazakh accents, it is possible to conclude that in Kazakh, the lack of data with accented samples makes model training more difficult than in multi-resource languages such as English and Spanish. Del Rio et al. [20] demonstrated how adapted models trained on data with speech samples with different accents can significantly improve recognition accuracy. However, the study stereotyped accents due to geographical references, as the selected samples included not only samples of accent speakers but also non-native speakers. Following Deng et al. [21] another option for solving the problem of low-resource languages is a self-learning method to improve accent identification and accented speech recognition.

During the creation of the model for the Kazakh accent, phonetic features, such as the special pronunciation of certain vowels and consonants, were incorporated. In addition to phonetic aspects, it is also necessary to account for grammatical features. In this context, the concept of Song et al. [22] is interesting, as it presents the use of deep neural networks, which can be used to create specialised models for accent classification. The study by Bashori et al. [23] addressed pronunciation training, while the creation of a Kazakh model requires a deeper analysis of the phonetic features of the Kazakh language.

Notably, the developed kazakhBERTmulti model demonstrated significant results in recognising the Kazakh accent, but there is a need to optimise and increase the amount of data to further improve its efficiency. In this context, the conclusion of Malik et al. [24] that the use of self-tuning systems can have a positive impact on improving the efficiency of the recognition process is noteworthy.

Based on the results of the study, improving the accuracy of Kazakh accent recognition can be associated with the creation of an extensive data corpus using speech recordings of Kazakh speakers with different dialects and accents, data augmentation, i.e., using methods of artificially increasing the volume of data, and the creation of multilingual corpora. In the context of improving recognition accuracy, Prabhu et al. [25], involve the use of accent-specific codebooks. The advantage of this approach is the peculiarities of accents through the creation of specific features for each accent. Codebooks can efficiently classify accents and use this knowledge in recognition.

During the creation of the model for recognising the Kazakh language, such parameters as the agglutinative structure and the use of the Cyrillic alphabet alongside the Latin alphabet were accounted for. In the development of a model for the Kazakh accent, social characteristics are essential. These aspects were analysed by Johnson et al. [26], highlighting the importance of accent variations for the successful operation of automatic speech recognition systems. The study demonstrated that accent perception and adaptation to it develop with age, which means that age-related features can also be a factor affecting the quality of recognition. Qian and Xiao [27] analysed the features of speech recognition in people with dysarthria. The study results are notable, as the difficulties encountered in recognising accented speech are similar to those associated with dysarthria, as both are deviations from "normative" pronunciation. Based on this, the statement by Feng et al. [28] on the need to create inclusive speech recognition systems aimed at minimising errors in recognising accented speech is notable.

The comparison of the study results with those of Kheddar et al. [29] determined that more textual data is needed to build a BERT model, not just audio files, as shown in the paper by these authors. Transfer learning, as discussed in the paper of the researchers, can be used for the Kazakh model, as the resources of Kazakh speech are limited. The BERT model requires text-to-speech data for training and fine-tuning in Kazakh, and the study by Bell et al. [30] considered adaptation methods centred on ASR systems. The study also highlighted the difficulties associated

with the high computational complexity of adaptation algorithms and the need for a large amount of data for their successful implementation. However, it should be noted that these adaptation algorithms help neural networks to better cope with different accents and can be used to improve the accuracy of speech recognition in Kazakh.

Thus, the creation of an effective model for the recognition of the Kazakh accent requires the integration of innovative machine learning and linguistic analysis techniques. The success of such systems will depend on their ability to incorporate and account for the unique features of Kazakh speech while ensuring high accuracy and accessibility of the technology to a wide range of users. To improve the recognition of Kazakh speech, deep neural networks can be used to create accurate models, specialised approaches for low-resource languages, such as self-learning, integration of self-learning and domain-adaptive learning methods to improve accent identification without the need for extensive data, and development of models that take into account not only phonetic features but also the grammatical structure of the Kazakh language. The use of specialised approaches for low-resource languages, such as self-learning, will help improve the accuracy of Kazakh speech recognition models.

5. Conclusions

The main stages of creating a model for accent recognition include audio data preprocessing (removing background noise, normalising the volume and splitting recordings into short fragments), extracting MFCC features to represent audio data in a format suitable for analysis by a neural network, suitable for analysis by a neural network, Mel scale filtering, which involves passing the spectrum through a set of Mel filters that simulate the frequency sensitivity of the human ear, converting MFCC features into a model-compatible format, training the model and evaluating its performance.

Based on the analysis of BERT models for different languages, the following conclusions can be drawn. BERT models demonstrate different performances depending on the language and its structure. Each language requires a unique approach to model tuning and training due to its morphological, syntactic and phonetic characteristics. For example, the Spanish and Russian versions of BERT must consider complex systems of declensions and inflexions, while the Chinese model must cope with the character system and word ambiguity. BERT for English achieves the highest accuracy, completeness and F1-measure scores, which is since the model was initially trained on large volumes of English texts.

The English version of the model consistently achieves 90-96% accuracy, which makes it one of the most reliable for natural language processing tasks. Multilingual BERT has an average performance, inferior to specialised models for individual languages. Although Multilingual BERT is versatile and can process several languages simultaneously, its accuracy (70-85%) and F1-measure (70-80%) remain lower than those of language-specific models. This is also explained by the need to process more diverse language structures and the limited amount of data for each specific language.

Based on the results of this study, the developed kazakhBERTmulti model demonstrates high accuracy and improved performance compared to Multilingual BERT Russian. The F1-measure of the kazakhBERTmulti model is 0.68, which is 0.04 higher than the similar indicator of the multilingual model. This indicates that the kazakhBERTmulti model is better adapted to the peculiarities of the Kazakh language and accents. Tokenisation in the Kazakh language presents challenges due to its agglutinative structure. The Kazakh language, with its rich morphology, requires more complex tokenisation, including sub-word segmentation for accurate root and suffix extraction. This should also be accounted for in the development of models for the Kazakh language to improve their performance.

The conversion of MFCC features into text strings and the use of BERT ensures effective accent recognition. Experimental results showed an accuracy of about 92%, which confirms the reliability of the method based on the conversion of audio features and their subsequent BERT processing. The limited amount of available data in the Kazakh language may reduce the accuracy of the model, despite the high performance.

Despite the encouraging accuracy of approximately 92% on a strictly speaker-disjoint, held-out test set, the corpus size remains modest (≈ 1.9 h; 13 speakers across five accents). Consequently, the reported figures should be interpreted as evidence of feasibility rather than definitive generalisation performance. In particular,

the limited speaker and channel diversity may inflate in-domain accuracy and under-estimate real-world variability (microphone type, room acoustics, speaking rate, sociolect). Future work will prioritise scaling the dataset (≥ 100 speakers per accent), cross-corpus evaluation (train/test on different collections), k-fold speaker-independent validation, and domain-shift tests (mismatched devices and environments) to obtain tighter, deployment-grade generalisation bounds.

In the context of future research, additional experiments with different neural network architectures and optimisation algorithms could be conducted to improve identification accuracy. Other accents and languages could be added to create a more extensive and diverse training database, adaptation to noisy environments could be performed, and the effect of background noise on identification accuracy could be studied. Accent-aware voice identification methods continue to be a hot topic in research and practical applications. The use of technologies such as MFCC and BERT creates new opportunities for creating more accurate and reliable speech recognition systems. Further analysis of accents and the development of adaptive systems that can efficiently process different languages and dialects is required.

Thus, the limitations of this study are that only a few languages were considered, and the performance of not all existing models was evaluated. Promising areas related to the development of a method for using a neural network for voice identification concerning accent features are the following: development of models concerning accent specifics, data collection and augmentation, contrastive learning, studying the integration of acoustic accent markers, improving deep learning models, synthesis of voices with different accents, and interpretation of model solutions.

References

1. N. Kadyrbek, M. Mansurova, A. Shomanov, and G. Makharova, *The development of a Kazakh speech recognition model using a convolutional neural network with fixed character level filters*, Big Data and Cognitive Computing, vol. 7, no. 3, 132, 2023.
2. A. Mukhamadiyev, I. Khujayarov, O. Djuraev, and J. Cho, *Automatic speech recognition method based on deep learning approaches for Uzbek language*, Sensors (Basel), vol. 22, no. 10, 3683, 2022.
3. O. Mamyrbayev, D. Oralbekova, K. Alimhan, T. Turdalykyzy, and M. Othman, *A study of transformer-based end-to-end speech recognition system for Kazakh language*, Scientific Reports, vol. 12, no. 1, 8337, 2022.
4. M. Najafian, and M. Russell, *Automatic accent identification as an analytical tool for accent robust automatic speech recognition*, Speech Communication, vol. 122, no. 10–11, pp. 44–55, 2020.
5. T. T. Schnoor, M. C. Kelley, and B. V. Tucker, *Automatic accentedness rating using deep neural networks*, Proceedings of Meetings on Acoustics, vol. 45, no. 1, 060013, 2021.
6. Q. Li, Q. Mai, M. Wang, and M. Ma, *Chinese dialect speech recognition: A comprehensive survey*, Artificial Intelligence Review, vol. 57, no. 2, 25, 2024.
7. M. Tiwari, and D. K. Verma, *Real voice recognition and authentication system: A comprehensive review*, International Journal of Intelligent Communication and Computer Science, vol. 2, no. 1, pp. 13–33, 2024.
8. C. Graham, and N. Roll, *Evaluating OpenAI's whisper ASR: Performance analysis across diverse accents and speaker traits*, JASA Express Letters, vol. 4, no. 2, 025206, 2024.
9. Th. Gaudier, M. Tahon, A. Larcher, and Ya. Esteve, *Automatic voice identification after speech resynthesis using PPG*, in N. Dehak and P. Cardinal (Eds.), Odyssey 2024: The Speaker and Language Recognition Workshop, Quebec City, Canada, June 18–21, 2024, pp. 187–193.
10. X. Y. Yang, Sh. D. Zhang, R. Xiao, Ji. Yu, and Z. Ya. Li, *Speech recognition of accented Mandarin based on improved conformer*, Sensors, vol. 23, no. 8, 4025, 2023.
11. Kh. Zaman, M. Sah, C. Direkoglu, and M. Unoki, *A survey of audio classification using deep learning*, IEEE Access, vol. 11, pp. 106620–106649, 2023.
12. H. Kheddar, M. Hemis, and Ya. Himeur, *Automatic speech recognition using advanced deep learning approaches: A survey*, Information Fusion, vol. 109, no. 3, 102422, 2024.
13. A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman, *GLUE: A multi-task benchmark and analysis platform for natural language understanding*, in Proc. 7th International Conference on Learning Representations (ICLR 2019), 2019. Available: https://arxiv.org/abs/1804.07461.
14. J. Cañete, G. Chaperon, R. Fuentes, J. Pérez, and J. Ponce, *Spanish pre-trained BERT model and evaluation data*, in Proc. Practical Machine Learning for Developing Countries Workshop (NeurIPS 2020), 2020. Available: https://arxiv.org/abs/2308.02976.
15. M. Sun, C. Huang, X. Qiu, H. Yang, and Y. Liu, *THUCNews: Chinese text classification dataset*, Beijing University of Posts and Telecommunications, 2016. Available: https://figshare.com/articles/dataset/THUCNews_Chinese_News_Text_Classification_Dataset/28279964.
16. A. Rogers, A. Romanov, A. Rumshisky, and C. Tan, *RuSentiment: An enriched sentiment analysis dataset for social media texts in Russian*, in Proc. 27th International Conference on Computational Linguistics (COLING 2018), pp. 755–763, 2018. Available: https://aclanthology.org/C18-1064/.

17. K. Khandelwal, P. Jyothi, A. Awasthi, and S. Sarawagi, *Black-box adaptation of ASR for accented speech*, in H. Meng, B. Xu, and T. F. Zheng (Eds.), Proc. 21st Annual Conf. Int. Speech Communication Association (INTERSPEECH 2020), Virtual Event, Shanghai, China, October 25–29, 2020, pp. 1281–1285.
18. K. Radzikowski, L. Wang, O. Yoshie, and R. Nowak, *Accent modification for speech recognition of non-native speakers using neural style transfer*, EURASIP Journal on Audio, Speech, and Music Processing, vol. 2021, no.
19. H. J. Na, and J. S. Park, *Accented speech recognition based on end-to-end domain adversarial training of neural networks*, Applied Sciences, vol. 11, no. 18, 8412, 2021.
20. M. Del Río, C. Miller, J. Profant, J. Drexler-Fox, Q. Mcnamara, N. Bhandari, N. Delworth, I. Pirkin, M. Jetté, S. Chandra, P. Ha, and R. Westerman, *Accents in speech recognition through the lens of a world Englishes evaluation set*, Research in Language, vol. 21, no. 3, pp. 225–244, 2023.
21. K. Deng, S. Cao, and L. Ma, *Improving accent identification and accented speech recognition under a framework of self-supervised learning*, in Proc. 22nd Annual Conf. Int. Speech Communication Association (INTERSPEECH 2021), Baixas, pp. 881–885, 2021.
22. T. Song, L. Th. H. Nguyen, and T. V. Ta, *MPSA-DenseNet: A novel deep learning model for English accent classification*, Computer Speech & Language, vol. 89, no. 2, 101676, 2025.
23. M. Bashori, R. Hout, H. Strik, and C. Cucchiarini, *I can speak: Improving English pronunciation through automatic speech recognition-based language learning systems*, Innovation in Language Learning and Teaching, vol. 18, no. 5, pp. 443–461, 2024.
24. M. Malik, M. K. Malik, Kh. Mehmood, and I. Makhdoom, *Automatic speech recognition: A survey*, Multimedia Tools and Applications, vol. 80, no. 6, pp. 9411–9457, 2021.
25. D. Prabhu, P. Jyothi, S. Ganapathy, and V. Unni, *Accented speech recognition with accent-specific codebooks*, in Proc. 2023 Conf. Empirical Methods in Natural Language Processing (EMNLP 2023), Stroudsburg, pp. 7175–7188, 2023.
26. E. K. Johnson, M. Heugten, and H. Buckler, *Navigating accent variation: A developmental perspective*, Annual Review of Linguistics, vol. 8, no. 1, pp. 365–387, 2022.
27. Zh. Qian, and K. Xiao, *A survey of automatic speech recognition for dysarthric speech*, Electronics, vol. 12, no. 20, 4278, 2023.
28. S. Feng, B. M. Halpern, O. Kudina, and O. Scharenborg, *Towards inclusive automatic speech recognition*, Computer Speech & Language, vol. 84, no. 4, 101567, 2024.
29. H. Kheddar, Ya. Himeur, S. Al-Maadeed, A. Amira, and F. Bensaali, *Deep transfer learning for automatic speech recognition: Towards better generalization*, Knowledge-Based Systems, vol. 277, 110851, 2023.
30. P. Bell, J. Fainberg, O. Klejch, J. Li, S. Renals, and P. Swietojanski, *Adaptation algorithms for neural network-based speech recognition: An overview*, IEEE Open Journal of Signal Processing, vol. 2, pp. 33–66, 2021.