Shala Kazakh: A Mixed Transcription of Kazakh and Russian

by

Mukhamejan Talap

Submitted to the Department of Robotics and Mechatronics
in partial fulfillment of the requirements for the degree of

Master of Science in Robotics

at the

NAZARBAYEV UNIVERSITY

Apr 2025

Author .....
Department of Robotics and Mechatronics
Apr 15, 2025

Certified by .....
Anara Sandygulova
Associate Professor
Thesis Supervisor

Accepted by .....
Yelyzaveta Arkhangelsky
Dean, School of Engineering and Digital Sciences

Shala Kazakh: A Mixed Transcription of Kazakh and Russian

by

Mukhamejan Talap

Submitted to the Department of Robotics and Mechatronics
on Apr 15, 2025, in partial fulfillment of the
requirements for the degree of
Master of Science in Robotics

Abstract

This thesis addresses the challenge of Automatic Speech Recognition (ASR) for "Shala Kazakh", a widespread code-switching phenomenon in Kazakhstan, where Kazakh speakers integrate Russian words and expressions in their speech. While ASR systems are rapidly improving for high-resource languages, they struggle to recognise code-switching scenarios, especially in low-resource languages, like Kazakh. This research introduces a novel approach by training a state-of-the-art (SOTA) Whisper model on both monolingual Kazakh and Russian datasets, additionally training it on a freshly collected 52-hour code-switching dataset that captures bilingual speech patterns gathered from TikTok. The experimental results demonstrate that incorporating a Russian dataset significantly improves transcription for the code-switching scenario. This work provides a framework for developing robust ASR systems for other low-resource languages like Kazakh with similar code-switching scenarios, contributing both technological advances and language preservation.

Thesis Supervisor: Anara Sandygulova
Title: Associate Professor

Acknowledgments

I want to express my sincere gratitude to my supervisor, Prof. Anara Sandygulova, for her invaluable guidance, patience, and support. I am also expressing my gratitude to my groupmate, labmate, and close friend, Zholaman Kuangaliyev. During the last year, we won multiple Hackathons and worked on real-world projects, strengthening my abilities as a researcher and engineer.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Linguistic and Historical Background

Kazakhstan's sociolinguistic landscape is shaped by its complex history, language policy shift and cultural identity formation. The coexistence of Kazakh and Russian has given rise to a widespread phenomenon known as "Shala-Kazakh", in which Kazakh is mixed with Russian words, expressions or entire syntactic structures. This phenomenon is an example of code-switching, a common linguistic practice in which bilingual speakers switch between languages depending on context, subject, or social setting. To illustrate the linguistic characteristics of Shala-Kazakh, consider this example from our dataset:

> қазір **первичный** тексеріс тексеріватырқ **первичный** яғни бұдан өтетін болса машина ұнаса қайтадан біз iii автоэкспресс шығаратын боламыз **полный** тексеріп беретіндей

This sample demonstrates how Russian words (highlighted in bold) such as "первичный" (primary) and "полный" (full) are seamlessly integrated into the Kazakh grammatical structure.

The formation of Shala-Kazakh lies in Kazakhstan's Soviet past. During this time (1920-1921), Russian was promoted as a unifying language across all nations living in the USSR, often at the cost of Indigenous languages like Kazakh. The Russian

language has become dominant in education, science, governance, and urban life. As a result, an entire generation of ethnic Kazakhs grew up speaking Russian as their first language. Although Kazakh has survived in rural areas, it has experienced significant lexical borrowings and structural influence from Russian.

After gaining its independence in 1991, Kazakhstan initiated various language revitalisation reforms, making Kazakh a state language while recognising Russian as an official language for interethnic communication. This new policy created a bilingual environment where Kazakh and Russian are widely used and frequently blended.

From a linguistic perspective, Shala-Kazakh presents a hybrid structure where Kazakh is the main framework, while Russian serves a lexical or expression role. This switching may occur within a sentence or between sentences and involve more than simple word substitution, but also Russian words taking Kazakh suffixes. This structural complexity makes automatic speech recognition for Shala-Kazakh challenging.

1.2 Technological Context and Importance

Automatic Speech Recognition (ASR), which is primarily designed for single-language transcription, is trained on clean, well-structured datasets. As a result, it often struggles with code-switching language transcription, especially when one of the languages is low-resource, such as Kazakh. In the case of Shala-Kazakh, these challenges are even worse:

- **Data scarcity:** Kazakh has limited annotated data, and code-switching datasets are nearly non-existent.
- **Phonetic complexity:** Russian insersions often bring phonological patterns not native to Kazakh.
- **Orthographic inconsistency:** There are no spelling and transcription conventions for code-switched speech.

Addressing these challenges is a technical advancement that reflects the commitment to language equity. Developing robust ASR models that handle code-switching

is vital to preserve linguistic integrity and support bilingual communication.

1.3 Research Gap

Despite the progress in current ASR systems on multilanguage and code-switching scenarios, most focus on high-resource languages such as English-Spanish, Hindi-English, etc. The Kazak-Russian is largely unexplored, while Kazakh is a low-resource language. Current Kazakh ASR systems typically ignore those code-switching scenarios due to the lack of these code-switched datasets.

1.4 Problem statement

The specific problem of this research is developing an ASR model handling the unique characteristics of "Shala Kazakh" by transcribing mixed-language speech. This involves fine-tuning state-of-the-art (SOTA) models such as the Whisper [14] model on combining Kazakh and Russian speech corpora and creating a novel dataset where Kazakh and Russian co-occur naturally in the speech.

1.5 Research Contribution and Impact

This thesis makes both technical and linguistic contributions:

- **Code-switching dataset:** It introduces a novel dataset with naturally occurring code-switching from TikTok videos.
- **Model fine-tuning:** Fine-tuning and evaluation of state-of-the-art (SOTA) multilingual models such as Whisper for handling this scenario.
- **Empirical insights:** Demonstration that incorporating the Russian dataset can significantly enhance transcription accuracy for Shala-Kazakh.

The successful completion of this research will provide a framework for code-switching in other low-resource languages. Developing a robust ASR that significantly

improves mixed-language transcription will set a new benchmark for code-switching ASR systems.

Chapter 2

Literature review

The ASR system in this paper [6] employs a Context-Dependent Deep Neural Network Hidden Markov Models(DNN-HMM) architecture for the Acoustic Model. Transfer learning was applied to a pre-trained Russian dataset. For the fine-tuning 120 hours of Kazakh speech out of 147 hours collected by Speech Technology Center Ltd were used. Another DNN model was trained only using the Kazakh dataset for comparison purposes. Comparing the results, the cross-language DNN model(DNN-2), outperformed the Kazakh-only model(DNN-1). Achieving an Equal Error Rate (EER) of 2.3% in Yes/No, reducing EER to 15% , 10% from 22%, 19% for Cities and Surnames, respectively, in the Interactive Voice Response(IVR) scenario. In the Keyword Spotting(KWS) scenario, DNN-2 showed a lower EER rate than DNN-1 in a test with spontaneous speech and a target word list of 10 and 100 items. The cross-language model achieved a 13% EER, while Kazakh-only 18% in 10 words and 20% compared to 29% in the 100 words case.

The authors of the paper [15] explore the effectiveness of the bilingual ASR model for the Kazakh-Russian code-switching scenario. Researchers aim to determine whether training on mixed datasets is a better approach than a code-switched model alone. The dataset consists of call centre recordings, totalling 97.4 hours of Kazakh speech and 58.8 hours of Russian speech. Approximately 10% of the Kazakh speech includes language-switched Russian phrases. Performance was measured on a validation dataset consisting of 1.2 hours of Kazakh and 1 hour of Russian. The au-

thors used DNN-HMM models with bidirectional long-short-term memory(BLSTM) network architecture. The bilingual model achieved a Word Error Rate (WER) of 49.42% on the Kazakh evaluation set, with code-switching scenarios, compared to 52.38% with the monolingual model. For the Russian evaluation set (without code-switching), the bilingual model had a higher WER of 37.42% compared to 31.91% with the monolingual Russian model. After calculating the WER for matrix and embedded language words separately, the bilingual model showed a 14.69% improvement on the embedded data.

In this paper [11], the authors focused on building Kazakh ASR by using transfer learning from a Russian model due to limited data availability. The Russian model was trained on 100 hours of data with a neural network structure using 2 LSTM layers. For Kazakh, the dataset was collected from 50 native speakers who had read well-known Kazakh books, totaling 20 hours. Four configurations were tested: the LSTM model with and without Transfer learning and the BLSTM model with and without transfer learning. LSTM model with Transfer learning achieved an 8% improvement in training cost and 4% improvement in Label Error Rate compared to the model without fine-tuning. BLSTM showed the best performance, improving training cost by 24% and reducing LER by 32% compared to non-transfer one.

The paper [5] introduces the Kazakh Speech Corpus (KSC) to advance the Kazakh ASR system. The corpus consists of 332 hours of transcribed audio, with over 153000 utterances from 1600+ participants. The dataset was split into training (318.4 hours), validation (7.1 hours), and test (7.1 hours) sets. The authors trained the traditional DNN-HMM model using Kaldi and end-to-end (E2E) models (RNN-based and Transformer-based [16]) using ESPnet. On the conventional DNN-HMM model, they achieved a Character Error Rate (CER) of 4.6% and a Word Error Rate (WER) of 13.7%. RNN-based E2E model achieved 4.0% of CER and 11.7% of WER on the test set. The best-performing model was the Transformer-based model reaching CER of 2.8% and WER of 8.7% on the test set.

This paper [9] extends previous work on Kazakh ASR by developing a multilingual E2E model for recognising Kazakh, Russian, and English. The system used a

combined dataset of three languages with separate or combined grapheme sets, which include unique symbols for each language. As well as previously used 332 hours of the Kazakh dataset, they used a manually cleaned 334-hour subset of the Openstt dataset and a 330-hour subset of Mozilla's Common Voice, along with the Kazakh-accented English for the English dataset. The multilingual model with data augmentation achieved performance close to the monolingual one, showing an average WER of 21.1% versus 20.9% on test sets.

The authors of the paper [10] introduced KSC2, a significantly expanded KSC aimed at improving ASR for Kazakh, particularly in spontaneous and code-switching contexts. The new corpus includes over 1,128 hours of transcribed data, which is significantly larger than the previous one, incorporating KSC, data from KazakhTTS2 (which provides for data recorded by 5 professional speakers), and additional data collected from different sources such as television news, radio programs, parliament, and podcasts. KCS2 includes Kazakh-Russian code-switching, addressing common features in Kazakh communication. The model architecture is a transformer-based ASR system with 18 encoder and 6 decoder layers, and it was trained using ESPnet. ASR achieved an overall WER of 15.6%.

The paper [2]presents an approach to improving Kazakh speech recognition by fine-tuning a small Kazakh speech corpus that is pre-trained on a multilingual dataset using the Wav2Vec2 model [1]. The corpus is 6 hours of transcribed data, where 4.5 hours were used for training and 1.5 hours for testing. The study fine-tuned two versions of the model, Wav2Vec2-large-xlsr-53: Pretrained on 53 languages for 50,000 hours and Wav2Vec2-xls-r-300m: Pretrained on 128 languages for 300 million parameters. Fine-tuning was performed with varying training epochs(1, 10, 16, and 24 epochs). For the first model, WER reduced from 45.05% (1 epoch) to 31.33% (16 epochs) and CER reduced from 10.03% (1 epoch) to 7.08% (24 epochs). For the Wav2Vec2-xls-r-300m, WER reduced from 49.16% (1 epoch) to 32.56% (24 epochs), and CER reduced from 11.08% (1 epoch) to 7.97% (24 epochs). The model showed improved recognition of common words but struggled with rare words, proper nouns, and noisy data.

The study [7] evaluates the performance of state-of-the-art ASR models for the Kazakh language, comparing self-supervised learning (Wav2Vec2.0 and XLSR-53) with fully supervised learning (Whisper). The researchers used 4 datasets, ISSAI KSC (332 hours of diverse speech recordings), M2ASRKazakh-78 (78 hours of reading-style recordings), Kazcorpus (44 hours of a dataset split into studio-recorded and spontaneous speech subcorpora), and KazLibriSpeech (1000 hours of audiobook recordings). In the paper, they were using Wav2Vec2.0 and XLSR-53, Whisper, and Baseline E2E Transformer. Wav2Vec2 and XLSR-53 were pre-trained on KazLibriSpeech and M2ASRKazakh-78 and then fine-tuned on ISSAI KSC, utilising the KenLm language model (LM) during decoding at the end. The Wshiper model was trained on KSC and KazCorpus on a multilingual configuration without using the LM. A Transformer-based model was trained on KSC and KazCorpus using LM. Best test set performance: CER of 2.7% and WER of 8.5% was by Wav2Vec2. XLSR-53 showed XLSR-53 slightly higher error rates than Wav2Vec2.0, with a CER of 4.3% and WER of 13.5%. The Whisper's "large" multilingual model achieved a CER of 4.1]% and WER of 19.8%. Baseline E2E Transformer achieved a CER of 2.8% and WER of 8.7%. Pretraining on large, multilingual datasets significantly boosts performance, especially for low-resource languages. Language Model integration further improves the accuracy of ASR models.

Chapter 3

Data Preparation

We utilised several monolingual datasets for fine-tuning the Whisper model: Kazakh Speech Corpus 2 (KSC2), Kazakh Speech Dataset (KSD) [8], and Golos [4]—Russian Dataset for Speech Research. Despite that, I collected the dataset using a natural code-switching scenario.

3.1 Kazakh Speech Corpus 2

KSC2 is the first open-source, industrial-scale Kazakh voice corpus created exclusively for Automatic Speech Recognition (ASR) research and development. This collection comprises around 1,128 hours of high-quality transcribed voice data, far more than any prior Kazakh speech corpus.

KSC2 combines two previously available corpora, the Kazakh Speech Corpus (KSC) and KazakhTTS2. The authors supplemented the dataset with additional information from various sources, including television news, television and radio programs, parliamentary speeches, and podcasts.

KSC2's incorporation of Kazakh-Russian code-switching utterances is very useful for this research. This feature specifically addresses the "Shala Kazakh" phenomenon, in which speakers organically incorporate Kazakh and Russian aspects into their speech. The rate of code-switching varies by data source, with podcasts (9.6%), television shows (10.2%), and radio programs (5.8%) showing the greatest rates.

The dataset contains various speech styles, from carefully read speech (KSC, KazakhTTS2, television news, parliament speeches) to spontaneous discussions (television shows, radio programs, podcasts). This diversity is critical for creating a robust model capable of managing real-world speech patterns, such as genuine code-switching in ordinary discussions.

3.2 Kazakh Speech Dataset

Alongside the KSC2 corpus, the Kazakh Speech Dataset (KSD) fortifies the efficacy of our fine-tuning procedure. The KSD substantially adds to Kazakh speech recognition resources, with 554 hours of transcribed audio data gathered from 873 native speakers. The KSD provides many unique benefits for our study.

The dataset shows a balanced demographic distribution of speakers all over multiple regions of Kazakhstan: 42.98% from the southern region, 21.05% from the western region, 15.78% from the northern region, 11.40% from the eastern region, and 8.77% from the central region. This geographic variety helps the model generalise across regional accent variances, which is crucial for addressing the phonetic nuances in code-switching scenarios.

Age and Gender Distribution: The dataset comprises speakers from diverse demographic groups (64.3% aged 18-23, 15.3% aged 23-28, 12.2% aged 28-33, and 8.2% above 33) and demonstrates a balanced gender ratio (61.18% male and 38.82% female speakers). It helps in minimizing demographic biases within the refined model.

High-Quality Transcriptions: Each KSD audio recording was carefully verified by native Kazakh speakers and a competent linguist, guaranteeing transcription precision. This quality control technique was extremely helpful for our study, offering dependable ground truth for training and assessment.

Diverse Acoustic Conditions: Unlike laboratory-recorded datasets, KSD audio was obtained utilising mobile devices (Android and IOS) in fluctuating locations such as universities, residences, and workplaces. The diversity in recording settings enhances the model's capacity to manage real-world acoustic fluctuations.

The combination of KSD with KSC2 provided a comprehensive foundation for fine-tuning Whisper.

3.3 Golos Russian Dataset

Russian speech corpus was crucial for solving the Kazakh-Russian code-switching recognition problem. For this purpose, we employed the Golos dataset, a thorough Russian voice corpus specifically built for speech recognition research. Karpov et al. (2021) created this dataset as one of the most significant manually annotated Russian speech resources accessible to the academic community [4]. The Golos corpus contains over 1,240 hours of transcribed voice data, much more than earlier Russian datasets like the Russian components in Common Voice (111 hours) or M-AILABS (47 hours). This comprehensive dataset offered substantial insights into Russian phonetics and pronunciation patterns, essential for analyzing the Russian components of code-switched speech.

The dataset is structured into two primary domains:

Crowd Domain: This subset contains 979,796 audio recordings, totalling 1,095 hours, collected using the Yandex.Toloka crowdsourcing platform. These recordings cover a variety of speaker backgrounds and recording settings, closely resembling authentic speech contexts. Fairfield Domain: This subset contains 124,003 audio files with a total duration of 132.4 hours, recorded in studio conditions with the SberPortal bright screen at varied distances of 1, 3, and 5 meters from the speakers. This domain was very significant for training the model to process voice recorded from a distance, a frequent occurrence in practical applications.

3.4 Data Split

In the early stages of training the model, we dig into preparing the train, test and validation datasets, for KSC2 and KSD. On the first attempt, several thousand audio files were missing due to a mistake and preprocessing steps and left unnoticed till the

next attempt. At that stage, we merged these two datasets, which was wrong in our case since we could not check how the model performed on each test set or subfolder of one dataset during the test stage. The dataset structure can be seen in Table 3.1

Split	**KSC2**	**KSD**	**KSC2 + KSD**
Train	438,499	177,843	616,342
Validation	9,345	3,158	12,503
Test	6,733	3,129	9,862
Total	**454,577**	**184,130**	**638,707**

Table 3.1: Distribution of utterances across Kazakh speech datasets and their combination.

At one point, we decided to split the dataset to measure the metrics for each, and found this mismatch. For convenience and further data preparation, all the text and path to the audio file were collected into a single comma-separated values (.csv). There were missing .txt and .flac files in folders during the matching; 421 .txt and 417 .flac files were missing. While matching the Test folder, we had some issues in the tv_news folder since all the files (5236) were left with double extensions like .flac.flac or .txt.txt. We made a script to fix this issue. After fixing it, all text and audio files find their match. We left only the matching one; see Table 3.2

Split	**Subfolder**	**FLAC files**	**TXT files**	**Duration (hrs)**
Test	crowdsourced	3328	3328	7.03
	parliament	773	773	1.55
	podcasts	1547	1547	2.89
	radio	702	702	0.87
	talkshow	383	383	0.63
	tv_news	2618	2618	2.98
	Test Total	**9351**	**9351**	**15.95**
Train	crowdsourced	264952	264952	556.12
	parliament	23402	23402	45.94
	tts	159073	159073	315.49
	Train Total	**447427**	**447427**	**917.55**
Total		**456,778**	**456,778**	**933.50**

Table 3.2: Dataset structure showing the distribution of files across different subfolders for KSC2.

It is common practice to split the data into a 80/10/10 proportion. However,

the dataset was given with a test set, which was 1.7% of the dataset. To create a validation dataset, around 2% of files from each Train subfolder were extracted, see the new split in 3.3

Split	Subfolder	FLAC files	TXT files	Duration (hrs)
Test	crowdsourced	3328	3328	7.03
	parliament	773	773	1.55
	podcasts	1547	1547	2.89
	radio	702	702	0.87
	talkshow	383	383	0.63
	tv_news	2618	2618	2.98
	Test Total	**9351**	**9351**	**15.95**
Validation	crowdsourced	4168	4168	8.65
	parliament	868	868	1.71
	tts	5469	5469	10.81
	Validation Total	**10505**	**10505**	**21.17**
Train	crowdsourced	260784	260784	547.00
	parliament	22534	22534	44.00
	tts	153604	153604	305.00
	Train Total	**436922**	**436922**	**896.00**
Total		**456778**	**456778**	**933.12**

Table 3.3: Dataset structure showing the distribution of files across different subfolders after creating a validation set for KSC2

Later, the dataset was converted to .wav format from .flac. No audio was corrupted while preprocessing it.

While preparing the KSD dataset, there was no problem with any missing files during the matching text with audio, but two files were corrupted. Initially, the dataset was provided in openslr.org in 4 zip files, with JSON files for each folder. The Audio2 folder consists of 20 folders, from 21 to 40, with 2499 in 19 of them and 2269 in the 40th folder, totalling 49750 audio files. Audio3 has folders from 41 to 62, with 2499 in each of them, the last containing 1021 audio samples, totalling 53500. Audio4 ranges from 63 to 83, containing almost the same number of files as in the previous folders, the last being 519 and totalling 50500. Audio5 repeats the pattern of Audio4, ranging from 84 to 104. At the end, all audio folders, totalling 204248 audio files. See Table 3.4

The dataset given in openslr was not given with any test set. In the paper, the test

Folder	Subfolder Range	Audio Samples	Duration (hrs)
audio2	21-40	49750	127.31
audio3	41-62	53500	142.08
audio4	63-83	50499	147.36
audio5	84-104	50499	138.04
Total		**204248**	**554.79**

Table 3.4: Distribution of audio samples across different folders in KSD.

set was 15%, totalling 91 hours. In our case, we decided to split it in a ratio of KSC2. Around 2% of files were taken from each subfolder to create test and validation sets, see Table 3.5

Split	Files	Duration (hrs)
Train	196238	533.04
Validation	4005	10.87
Test	4005	10.88
Total	204248	554.79

Table 3.5: Dataset structure showing the distribution of files across different sets in KSD

The Golos dataset was predevied into the train and test classes, as shown in Table 3.6. To create a validation set, we took the same number of files from our train set, as illustrated in Table 3.7

By integrating the Golos Russian dataset with the KSC2 and KSD Kazakh corpora, we created a robust multilingual foundation for fine-tuning the Whisper model to accommodate Kazakh-Russian code-switching. The complementary strengths of both datasets, with Golos providing comprehensive coverage of Russian pronunciation patterns and the Kazakh corpora presenting native speech patterns, established an optimal training environment for developing a robust code-switching speech recognition system.

Split	Subfolder	Audio files	TXT files	Duration (hrs)
Train	crowd/0	100000	100000	94.72
	crowd/1	100000	100000	120.79
	crowd/2	100000	100000	113.46
	crowd/3	100000	100000	99.64
	crowd/4	100000	100000	136.22
	crowd/5	100000	100000	113.35
	crowd/6	100000	100000	135.86
	crowd/7	100000	100000	109.34
	crowd/8	100000	100000	105.38
	crowd/9	79796	79796	66.23
	farfield	124003	124003	132.46
	Train Total	**1103799**	**1103799**	**1227.45**
Test	crowd	9994	9994	11.20
	farfield	1916	1916	1.41
	Test Total	**11910**	**11910**	**12.61**
Total		**1115709**	**1115709**	**1240.06**

Table 3.6: Dataset structure showing the distribution of files across different folders in Golos.

Split	Folder	Audio files	TXT files	Duration (hrs)
Train	crowd/0	99003	99003	93.75
	crowd/1	98970	98970	119.56
	crowd/2	99010	99010	112.35
	crowd/3	98991	98991	98.64
	crowd/4	98946	98946	134.78
	crowd/5	99007	99007	112.20
	crowd/6	98973	98973	134.44
	crowd/7	99006	99006	108.27
	crowd/8	98942	98942	104.27
	crowd/9	78954	78954	65.53
	farfield	122087	122087	130.43
	Train Total	**1091889**	**1091889**	**1214.22**
Validation	crowd	9994	9994	11.18
	farfield	1916	1916	2.03
	Validation Total	**11910**	**11910**	**13.21**
Test	crowd	9994	9994	11.20
	farfield	1916	1916	1.41
	Test Total	**11910**	**11910**	**12.61**
Total		**1115709**	**1115709**	**1240.04**

Table 3.7: Dataset structure showing the distribution of files across different folders with validation set in Golos.

Chapter 4

TikTok Code-Switching Dataset

We developed a novel dataset comprising authentic social media content to complement the established speech corpora and address the specific challenges of spontaneous Kazakh-Russian code-switching. This dataset is a significant contribution to the research, as it captures naturally occurring code-switching in conversational contexts, a crucial element often missing from more controlled speech corpora.

4.1 Collection Methodology

The TikTok Code-Switching Dataset was systematically collected through the following process (see Figure B-1):

- **Source Selection:** We identified TikTok creators who regularly produce content featuring spontaneous Kazakh-Russian code-switching, focusing on conversational formats such as jokes, casual discussions, and live streams. These sources were selected to represent naturalistic speech patterns where code-switching occurs organically rather than in scripted contexts.
- **Automated Collection:** We developed a scraper that extracted videos from specified TikTok channels. This approach enabled efficient collection of data.
- **Audio Extraction and Standardization:** All collected videos were converted to audio-only files in WAV format with consistent technical specifica-

tions (mono channel, 16,000 Hz sampling rate) to ensure compatibility with the Whisper model architecture.

4.2 Preprocessing Pipeline

The raw audio underwent several preprocessing steps to optimize it for speech recognition training (see Figure B-2):

- **Duration Standardization:**All collected audio segments were split into a maximum length of 30 seconds to fit with Whisper's optimal processing capabilities during fine-tuning.
- **Speech Segmentation:**Using SileroVAD (Voice Activity Detection), I isolated speech segments from non-speech audio elements such as music, sound effects, or background noise. This step was critical for ensuring that only relevant speech data was included in the training corpus.
- **Initial Transcription:** The preprocessed audio segments were transcribed using a Whisper medium model that had been fine-tuned on KSC2 and KSD datasets with multilingual capability enabled. This approach provided initial transcriptions that accounted for Kazakh and Russian linguistic elements.
- **Manual Verification and Correction:** Each automated transcription underwent a thorough manual review by native speakers proficient in Kazakh and Russian. This critical quality control step ensured the accurate representation of code-switching points and corrected errors in the automated transcription process.

4.3 Dataset Characteristics

The final TikTok Code-Switching Dataset comprises 52 hours of high-quality speech data with several distinctive features:

- **Authentic Code-Switching:** Unlike laboratory or read speech corpora, this dataset captures genuine instances of Kazakh-Russian code-switching in real-world communication, providing valuable training examples for the model to learn natural switching patterns.

- **Diverse Speaking Styles:** The dataset includes varied speaking rates, emotional expressions, colloquialisms, and conversational elements often absent from more formal speech corpora.

- **Acoustic Diversity:** The recordings feature realistic acoustic conditions with varying levels of background noise, reverberation, and recording quality, helping to build robustness in the model's performance across different environments.

- **Contemporary Vocabulary:** The dataset captures current linguistic trends in code-switching, including modern terminology and evolving patterns of language mixing, by sourcing from recent social media content.

Overall, there are 52 hours of already transcribed audio files, with code-switching scenarios. To create train, validation and test sets, we took around 10% of the audio files for test and validation, see 4.1

Split	**Files**	**Duration (hrs)**
Train	6,110	41.8
Validation	764	5.18
Test	769	5.3
Total	7,643	52.28

Table 4.1: Dataset split statistics for spontaneous code-switching TikTok dataset.

Integrating this TikTok-derived dataset with the more structured KSC2, KSD, and Golos corpora created a comprehensive training foundation that balanced formal speech recognition capabilities with the ability to handle spontaneous code-switching. This combination proved instrumental in developing a Whisper model capable of accurately transcribing the fluid language boundaries characteristic of everyday Kazakh-Russian bilingual communication.

Chapter 5

Fine-tuning Whisper

5.1 Model Introduction

ASR in multilingual contexts presents unique challenges, particularly for code-switching scenarios where speakers alternate between languages within a single utterance. This chapter examines the fine-tuning of OpenAI's Whisper model to address the specific case of Kazakh-Russian code-switching, a common linguistic phenomenon in Kazakhstan often referred to as "Shala Kazakh."

Whisper represents a significant advancement in speech recognition technology through its innovative approach to weak large-scale supervision. Radford et al. (2022) documented that Whisper was pre-trained on 680,000 hours of multilingual and multitask supervision, enabling it to develop robust cross-linguistic capabilities without requiring dataset-specific fine-tuning. This architecture, shown in Figure B-3, presents considerable advantages for addressing code-switching recognition challenges, as it was specifically designed to handle various speaking styles, accents, and background noise conditions.

In the paper, Whisper presented five different models. Whisper family includes tiny, base, small, medium and large models. Models from tiny to medium are offered in two versions of English, one with tiny.en at the end and multilingual without any postfix. Large models include large, large-v2, which has the same architecture as whisper-large but trained 2.5x more epochs with added regularization for improved

performance. Also, OpenAI presented the whisper-large-v3 model, which has the same architecture as the previous large family models, except for the new Cantonese language token and spectrogram input, which uses 128 Mel frequency bins instead of 80, and whisper-large-v3-turbo, fine-tuned version of large-v3, with reduced decoder layers. Its architectural details are shown in Table A.1.

5.2 Model Architecture

We primarily employed three versions of the Whisper model for our experiments: small, medium and large-v2. The model ingests log-mel spectrograms as input features and outputs text tokens through an autoregressive decoder. Our implementation preserves the original architecture while focusing on adapting the model weights to the "Shala Kazakh" scenario.

OpenAI, for training their models, used 12 hours of audio for speech recognition and 31 hours of audio for translation. Despite having such a small amount of data, it showcased the average Word Error Count (WER) presented in Table A.2.

5.3 Dataset preparation

The KCS2 dataset came pre-divided into train and test sets, with the test set being 7.5 hours of speech data and containing 9351 utterances. To create a validation set, we:

- Kept the same distribution of speech types that were available in the train set (crowdsourced, parliament and TTS).
- Extracted around 7.5 hours with the same approximate average length containing 9,351 utterances.

Unlike KSD was provided only with the train set, so to create validation and test sets, we:

- Calculated the proportion of train, test and validation sets in KSC2 and applied to KSD.
- Extracted from train set in the same proportion for all speakers getting 3,155 for test and 3,158 utterances for validation dataset.

After splitting the individual datasets, we merged them to get a larger speech corpus by combining KSC2 and KSD to fine-tune our models specifically for Kazakh speech recognition. The resulting combined corpus is shown in the Table 3.1

The Golos Russian speech corpus was provided pre-defined training and test sets. The dataset came with two combined crowd and far-field scenarios. To create a validation dataset we:

- Extracted utterances from the training set that match the size of the test set
- Maintained the same ratio for crowd and far-field scenarios in the validation set.

After several trainings, some of the data in the test turned out to be missing, so we decided to reshuffle the data structure again.

5.4 Data Preprocessing

Before starting the training, we resampled the dataset to 16kHz since whispers were initially trained on this sample rate. Whisper model family ingest log-mel spectrogram as input; feature extraction was applied to compute log-mel spectrograms. Since we do not have extensive audio data, we applied it to the transcription for around a tenth of thousands of hours, lowercasing and punctuation removal. Length filtering ensures that audio samples are between 0-30 seconds and that transcriptions are within model limits. Dataset shuffling with a fixed seed (42) for reproducibility.

5.5 Evaluation Metrics

To evaluate the performance of our ASR models for code-switching scenarios, we employ two widely used metrics: Word Error Rate (WER) and Character Error Rate (CER) [13].

5.6 Word Error Rate (WER)

WER is the edit distance between the reference text and the hypothesis (recognized) text at the word level. Mathematically, WER is calculated as:

$$\text{WER} = \frac{S + D + I}{N} \tag{5.1}$$

where:

- S is the number of substitutions (words incorrectly recognized)
- D is the number of deletions (words omitted in the hypothesis)
- I is the number of insertions (extra words in the hypothesis)
- N is the total number of words in the reference text

The WER value is typically expressed as a percentage, with lower values indicating better performance. Perfect recognition would result in a WER of 0%, while incorrect recognition could exceed 100% due to insertions.

5.7 Character Error Rate (CER)

CER applies the same concept as WER but at the character level rather than the word level. This metric is particularly useful for morphologically rich languages, where small character differences can significantly impact meaning. The formula for CER is:

$$\text{CER} = \frac{S_c + D_c + I_c}{N_c} \tag{5.2}$$

where:

- S_c is the number of character substitutions
- D_c is the number of character deletions
- I_c is the number of character insertions
- N_c is the total number of characters in the reference text

CER typically results in lower values than WER for the same transcript, as it provides a more granular measure of recognition accuracy.

5.8 Implementation

To consistently calculate these metrics across all experiments, we utilized the jiwer library [3], a Python package specifically designed for computing WER and related metrics. This library implements the standard Levenshtein distance algorithm for calculating the minimum number of operations needed to transform one string into another.

5.9 Language Model - KenLM Overview

KenLM - is a widely used Language Model (LM) toolkit for efficiently estimating and evaluating n-gram language models. Developed by Kenneth Heafield, it is designed for speed and memory efficiency, making it suitable for decoding tasks in ASR systems. KenLM is designed to handle large training text corpora, allowing for training higher-order n-gram models on extensive datasets.

Integrating LM in ASR systems improves transcription accuracy by incorporating statistical knowledge of word sequences in the target language, which helps correct speech model errors based on linguistic context.

Chapter 6

Results and Discussion

6.1 Results

We conducted three comparisons of the whisper model: small, medium, and large-v2. All models were trained, with a warmup being 10% of the total step count. We explored various training hyperparameters to identify optimal configurations for each model size. We mostly focused on changing its learning rate and number of steps to train the model. The first successful model was using parameters from Kozhirbayev's paper [7]. Most of the models were trained on the first attempt of dataset. At that moment, my main hardware for training this model was RTX TITAN 24GB VRAM. While using a single GPU, only whisper-small and whisper-medium models were possible to train, with a batch size of at maxiumum 1 for the medium was 1 and 4 for the small model. We increased the number of accumulation gradient steps to emulate higher batch sizes. Accumulation gradient steps are used to handle cases where the desired batch size exceeds GPU memory. Instead of updating the model weights every batch, gradients from several batches are summed up and updated. So, for that purpose, the number of steps was set to 64 for that single GPU. Later, we got access to the DGX server, where I can train my models. After many attempts with different parameters, we established baseline performance using the following configurations. Results shown in the Table 6.1.

Despite fewer training steps, the Whisper medium model with 769M parameters

Model	Steps	LR	Grad. Accum.	# GPUs	GPU Type	WER	CER
Small	40K	1e-5	64	1	RTX TITAN [24GB]	25.2%	7.8%
Medium	35K	1e-5	64	1	RTX TITAN [24GB]	18.7%	5.9%
Large-v2	50K	1e-5	1	4	TESLA V100 [32GB]	25.7%	9.6%

Table 6.1: Performance comparison of models - baseline.

demonstrated the best performance. Based on promising results, we continued training models with extended training parameters, see Table 6.2.

Model	Steps	LR	Grad. Accum.	# GPUs	GPU Type	WER	CER
Small	80K	1e-5	64	1	RTX TITAN [24GB]	23.3%	7.0%
Medium	45K	1e-5	64	1	RTX TITAN [24GB]	18.4%	5.3%
Large-v2	50K	5e-6	1	4	TESLA V100 [32GB]	19.5%	6.8%

Table 6.2: Performance comparison models with extended training parameters.

Additional training with the whisper-medium model was conducted, with increased training steps of 80,000, and achieving 16.9% of WER and 5.3% of CER. These experiments confirmed that the medium model provided the best performance and training efficiency balance for the Kazakh speech recognition scenario.

During the whisper-medium model training, it was discovered that training is not going well on distributed GPUs. The problem is the Gradient accumulation steps. While training the whisper's medium model, since we had GPU limitations at that time, the training batch size was set to 1, but to emulate a higher batch size, gradient accumulation was set to 64. The total training time took 430 hours to train the single medium model. In case of training the model with a single GPU could fit well, since it does not require additional VRAM to train the model, but requires training time, which is essential in our use case. For example, training the large-v2 model, with 4 and 8 GPUs, with accumulation gradient steps set to 1, increases the effective batch size to 4 and 8. The math behind this effective batch size is (batch size per GPU) * (Number of GPUs) * (Gradient Accumulation Steps). While in the case of the medium model, it was 64. Now it's clear why the medium model was performing better in our case.

After a couple of times training a medium model with effective batch sizes of 4 and 8, we decided to switch back to the large model that doesn't require increasing

accumulation gradient steps. This decreases training time and gives good generalisation in our case, where only by changing the learning rate did we get an improvement of 6.2% on WER and 2.8% on CER.

Later, the training was using a new split dataset, where we can measure metrics for each dataset separately, or change the validation or train set, easily, which was later done. This time, I increased the number of GPUs to 8, and see how it goes with the new split dataset. New parameters to large-v2 were set like this: learning rate 5e-6, number of training steps is 100,000 and number of warm-up steps is 10,000. During training, We stopped the training at 60,000 steps, which gave similar results on the validation set while it was training. The training took 86 hours and gave around the same results as the previous large-v2, being slightly worse than the model on 50,000 steps, see Table 6.3. These changes in metrics are understandable since we added missing files back.

Dataset	**Number of Files**	**WER (%)**	**CER (%)**
KSC2	9351	23.94	9.75
KSD	4005	12.44	3.73
KSC2+KSD	13356	20.39	7.98

Table 6.3: Performance comparison on KSC2 and KSD datasets.

With the newly added russian dataset, our dataset was over 2,000 hours of data. During the training, I hit several limits of server we are training on. The most important limit for me was space, which was provided by the DGX server. While my dataset is on training, we preprocess it first and creates cache files, so while training, it won't go back to compute all the mel spectrograms. It is a better approach than training the model on the fly. The limitation with space is that we took all 15TB of space, which was available while preprocessing, so we decided to split the Golos set and took a subset of it of 110 hours for training, see updated structure Table 6.4

The second limitation we encountered was the processes that were provided by the server. The pre-processing itself, using mainly CPUs, while downsampling, creating mel-spectrograms and calculating the duration of audio for training. We occupied all of the processes, while others can't work on the server. We changed our ap-

Split	Folder	Audio files	TXT files	Duration (hrs)
Train	crowd/0	9000	9000	8.54
	crowd/1	9000	9000	10.84
	crowd/2	9000	9000	10.23
	crowd/3	9000	9000	8.98
	crowd/4	9000	9000	12.22
	crowd/5	9000	9000	10.16
	crowd/6	9000	9000	12.21
	crowd/7	9000	9000	9.80
	crowd/8	9000	9000	9.52
	crowd/9	7181	7181	6.00
	farfield	11160	11160	11.91
	Train Total	**97341**	**97341**	**110.41**
Validation	crowd	9994	9994	11.18
	farfield	1916	1916	2.03
	Validation Total	**11910**	**11910**	**13.21**
Test	crowd	9994	9994	11.20
	farfield	1916	1916	1.41
	Test Total	**11910**	**11910**	**12.61**
Total		**121161**	**121161**	**136.23**

Table 6.4: Dataset structure showing the subset distribution of files across different folders in Golos.

proach. Instead of training two checkpoints, one with Kazakh only and one with the mixed dataset, we continued my 60,000 checkpoint on the Golos small set. To train the model, 21,000 steps were enough to get more familiar with the new russian set, achieving 23.1% on evaluation set. During training with the Russian dataset, no Kazakh dataset was used while training or validation. Both of the checkpoints, one with Kazakh data only and one that was later tuned on Russian, were used to train on my freshly collected dataset.

First, we started with training on a Kazakh-only checkpoint and during the training, even after 50,000 steps, the evaluation WER didn't go lower than 50%. While, as expected, the other checkpoint took only 4,000 steps to get lower than 50%. Additionally, it was trained on 20,000 steps, but the best results were on 4,000. Not surprisingly, the tuning checkpoint on the russian dataset has an effect to generalize better on the code-switching scenario. The TikTok dataset, which was trained on only the Kazakh checkpoint, gave a WER of 48.81% and CER of 26.42%, which is not the best results so far. On the other hand, the mixed dataset gave WER of 36.02% and CER of 16.23%. It is 12% better than results on Kazakh checkpoint. While it seems to be way higher than initial results tested on KSC2 and KSD datasets, we should consider that the TikTok dataset contains spontaneous speech with a code-switching scenario.

For our ASR system, we implemented a language model using KenLM. We trained a trigram (3-gram) language model using the same textual data for our training set. The model was estimated using Kneser-Ney smoothing, the default and most effective algorithm in KenLM. After training, we converted the LM to a binary format to optimise loading speed and ensure memory efficiency during transcription. After the Whisper model generated its transcription, the language model was integrated into the decoding process as a post-processing step. Initial experiments with KenLM showed mixed results, where trigram weight was 30% with a number of audio samples candidates were tested to 5. We observed improvements up to 4%, which is a good improvement for some test cases, but the overall performance across all test scenarios did not consistently improve.

6.2 Future work

Based on the results of this study, there are several ways to explore to improve our ASR performance on the Shala-Kazakh case scenario. One key insight was that incorporating the Golos dataset with the Kazakh datasets led to better transcription accuracy on the code-switching dataset than using a Kazakh-only checkpoint. Due to technical limitations during the training, we relied on a subset of the original Russian speech corpus. More Russian data can provide a more robust foundation for our case scenario by removing this constraint in the future.

We plan to expand our code-switched dataset by collecting additional natural language mixing speech. This will make the model more robust and let us capture the diversity of linguistic patterns presented in spontaneous bilingual communication.

Our experiments with the trigram Language Model showed modest results in some test cases but were inconsistent in transcription overall. This may be a result of the small data that was provided to train the language model, or n-grams themselves may be ill-suited for the nature of the code-switching scenario in general. In the future, we plan to explore more context-aware language models. Specifically, using Large Language Models (LLM) such as KazLLM [12] on a small number of parameters, or their distilled version, could provide better generalisation in a bilingual context.

These directions will aim to improve transcription accuracy for Shala-Kazakh and create a more inclusive and linguistically adaptable ASR system for low-resource bilingual communities.

Appendix A

Tables

Model	Layers	Width	Heads	Parameters
Tiny	4	384	6	39M
Base	6	512	8	74M
Small	12	768	12	244M
Medium	24	1024	16	769M
Large	32	1280	20	1550M

Table A.1: Model architecture specifications and parameter counts.

Model	Kazakh WER (%)
Whisper tiny	165.2
Whisper base	109.2
Whisper small	70.3
Whisper medium	48.8
Whisper large	43.8
Whisper large-v2	**37.7**

Table A.2: Word Error Rate (WER) performance of different Whisper model sizes on Kazakh language speech recognition. Lower values indicate better performance.

Appendix B

Figures

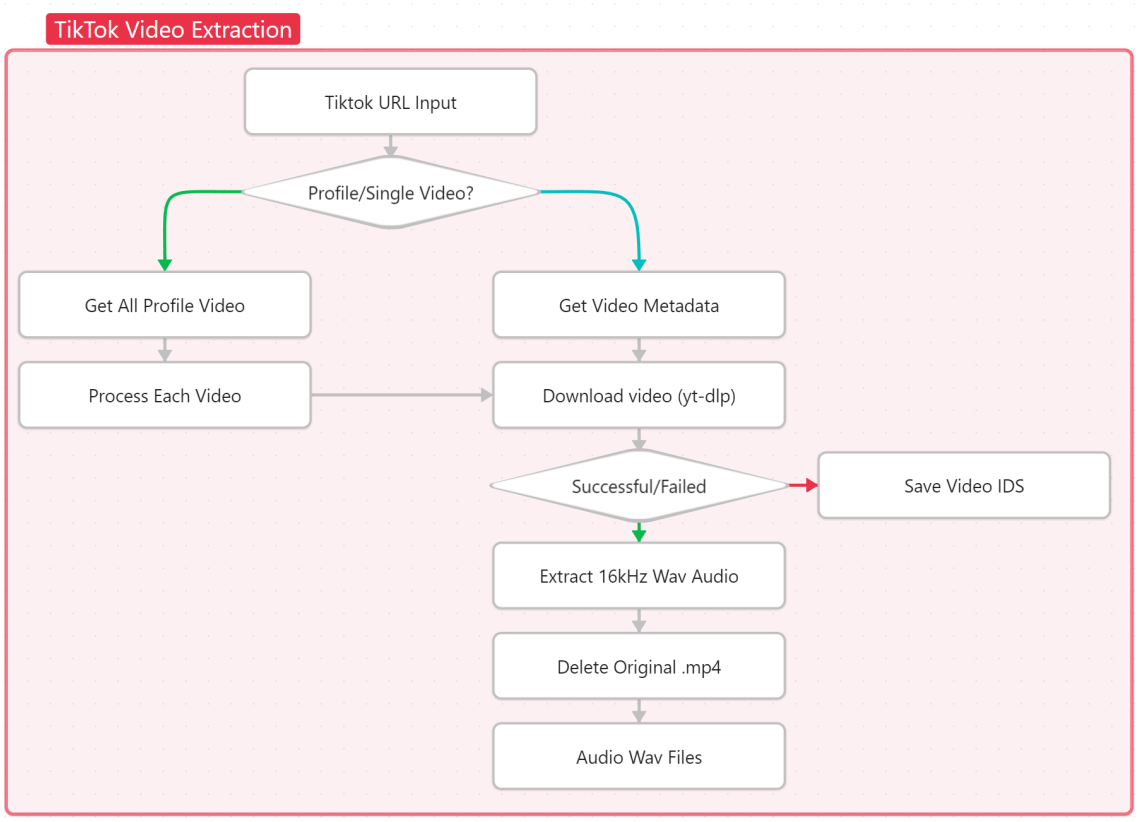

Figure B-1: TikTok video extraction and audio conversion pipeline for Kazakh-Russian code-switched speech collection.

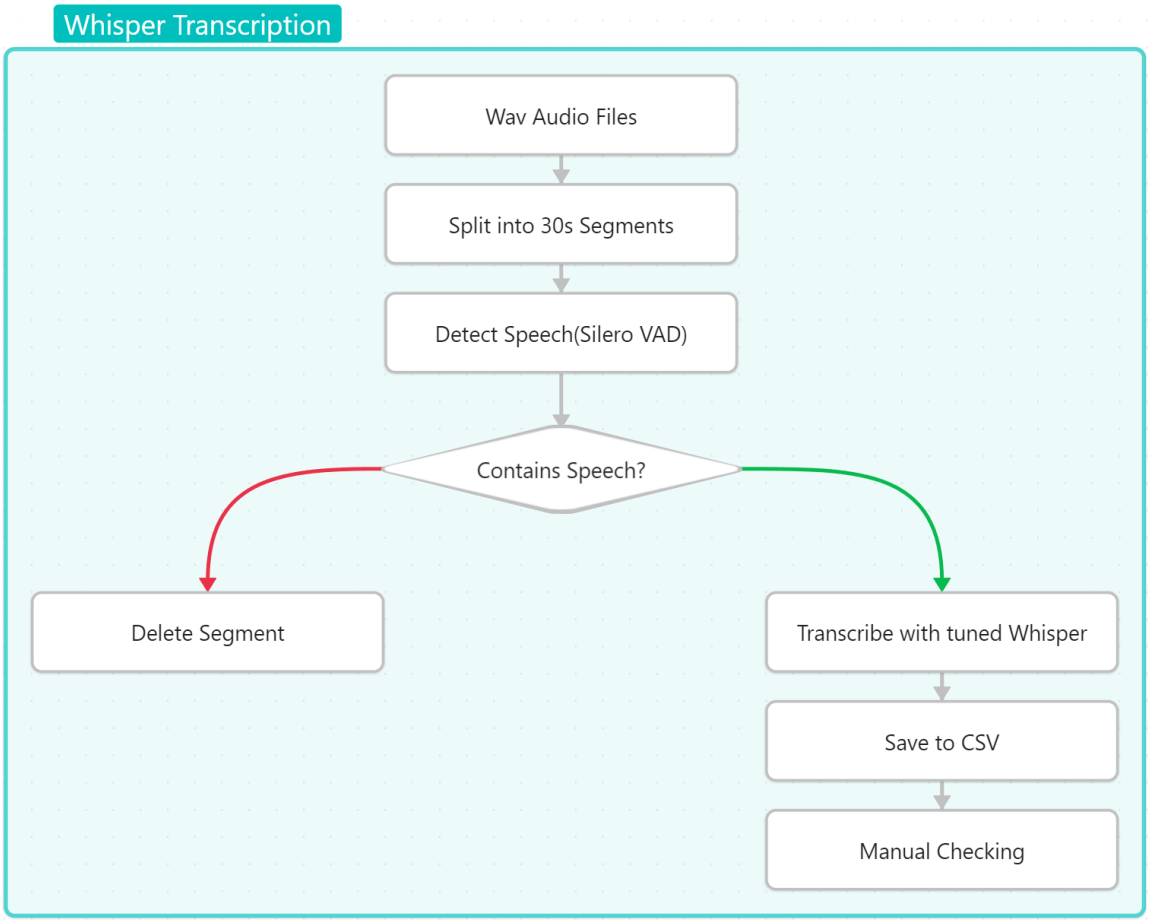

Figure B-2: Whisper-based transcription workflow with speech detection, segmentation, and manual verification for code-switched audio.

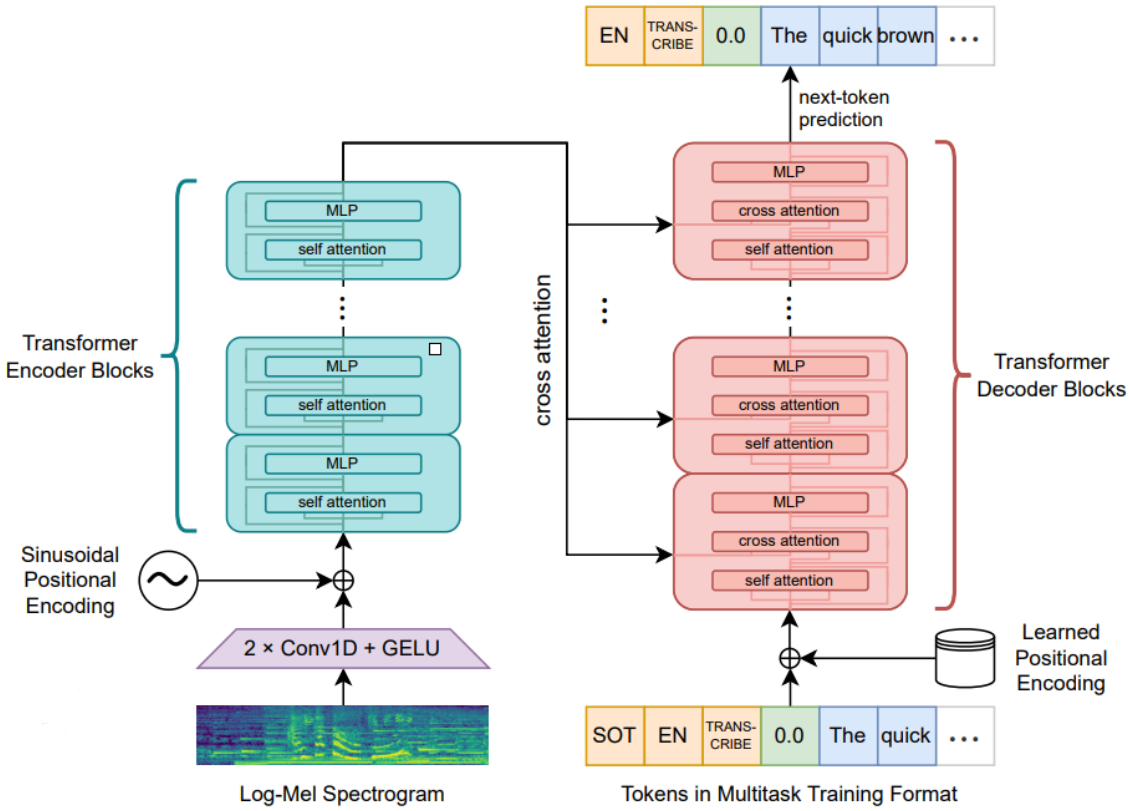

Figure B-3: Whisper architecture

Bibliography

[1] Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations, 2020.

[2] Kairatuly Bauyrzhan, Mansurova Madina, and Ospan Assel. Fine-tuning the wav2vec2 model for kazakh speech: A study on a limited corpus. In *2023 IEEE International Conference on Smart Information Systems and Technologies (SIST)*, pages 124–128. IEEE, 2023.

[3] Jitsi. jiwer: Similarity measures for automatic speech recognition evaluation. https://github.com/jitsi/jiwer, 2023.

[4] Nikolay Karpov, Alexander Denisenko, and Fedor Minkin. Golos: Russian dataset for speech research. *arXiv preprint arXiv:2106.10161*, 2021.

[5] Yerbolat Khassanov, Saida Mussakhojayeva, Almas Mirzakhmetov, Alen Adiyev, Mukhamet Nurpeiissov, and Huseyin Atakan Varol. A crowdsourced open-source kazakh speech corpus and initial speech recognition baseline, 2021.

[6] Olga Khomitsevich, Valentin Mendelev, Natalia Tomashenko, Sergey Rybin, Ivan Medennikov, and Saule Kudubayeva. A bilingual kazakh-russian system for automatic speech recognition and synthesis. volume 9319, pages 25–33, 09 2015.

[7] Zhanibek Kozhirbayev. Kazakh speech recognition: Wav2vec2.0 vs. whisper. *Journal of Advances in Information Technology*, 14:1382–1389, 01 2023.

[8] Madina Mansurova and Nurgali Kadyrbek. The development of a kazakh speech recognition model using a convolutional neural network with fixed character level filters, July 20 2023.

[9] Saida Mussakhojayeva, Yerbolat Khassanov, and Atakan Varol. A study of multilingual end-to-end speech recognition for kazakh, russian, and english, 08 2021.

[10] Saida Mussakhojayeva, Yerbolat Khassanov, and Atakan Varol. Ksc2: An industrial-scale open-source kazakh speech corpus. pages 1367–1371, 09 2022.

[11] Amirgaliyev E. N., Kuanyshbay D. N., and Baimuratov O. Development of automatic speech recognition for kazakh language using transfer learning, 2020.

[12] ISSAI Institute of Smart Systems and Artificial Intelligence. Kaz-llm-rus. https://issai.nu.edu.kz/ru/kazllm-rus/. Accessed: May 12, 2025.

[13] Youngja Park, Siddharth Patwardhan, Karthik Visweswariah, and Stephen C Gates. An empirical analysis of word error rate and keyword error rate. In *Interspeech*, volume 2008, pages 2070–2073, 2008.

[14] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision, 2022.

[15] Dmitrii Ubskii, Yuri Matveev, and Wolfgang Minker. Impact of using a bilingual model on kazakh-russian code-switching speech recognition. In *CEUR Workshop Proceedings*, 2020.

[16] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023.