Fine-Tuning Large Language Models for Kazakh Text Simplification

Alymzhan Toleu , Gulmira Tolegen , Irina Ualiyeva *

Posted Date: 24 June 2025

doi: 10.20944/preprints202506.1947.v1

Keywords: fine-tuning; Kazakh language; large language models; text simplification

 doi:10.20944/preprints202506.1947.v1

Article

Fine-tuning Large Language Models for Kazakh Text Simplification

Alymzhan Toleu [1,2], Gulmira Tolegen [1,2] and Irina Ualiyeva [1,3,*]

[1] Institute of Information and Computational Technologies, Almaty, 050010, Kazakhstan
[2] AI Research Laboratory, Satbayev University, Almaty, Kazakhstan
[3] Faculty of Information Technology, Al-Farabi Kazakh National University, Almaty, 050040, Kazakhstan
* Correspondence: i.ualiyeva@gmail.com;

Abstract

This work addresses the task of text simplification for Kazakh, a morphologically rich and low-resource language. We propose KazSim, a fine-tuned simplification model based on large language models (LLM), using both the multilingual Llama series and Qwen2 backbones To support model training, we construct a parallel simplification dataset using a proposed Kazakh sentence complexity identification pipeline, selecting complex sentences from raw corpora with the proposed heuristic approach. As baselines, we include standard Seq2Seq models, Kazakh domain-specific large language models, and general-purpose instruction-following models in a zero-shot setup. Evaluation is performed on both an automatically constructed test set and a semi-manually created benchmark, using SARI, BLEU, ROUGE, and Bert-score. Results show that KazSim consistently outperforms all baselines, including domain-specific LLMs and zero-shot models, achieving good simplification quality while preserving meaning and controlling output length. We also examine the impact of prompt language on generation quality, comparing English and Kazakh instructions. While performance remains consistent overall, models tend to produce slightly better outputs when prompted in Kazakh, particularly in zero-shot and domain-specific settings.

Keywords: fine-tuning; kazakh language; large language models; text simplification

1. Introduction

Text simplification is a task in natural language processing (NLP) that reduces linguistic complexity while preserving the original meaning. It has applications in education, public communication, and accessibility. In education, simplification allows students in primary and secondary schools to better understand complex texts. In public services, simplified documents improve the readability of legal and administrative content. In NLP pipelines, simplification can be used as a preprocessing step to improve downstream tasks such as machine translation, summarization, and information retrieval, especially for low-resource and morphologically rich languages like Kazakh.

In the context of Kazakh, text simplification is an important task due to the complexity of formal and classical written content. Many texts in the literature contain complex syntactic constructions, long participial forms, and infrequently used vocabulary. This creates a barrier for young readers and for individuals whose literacy was primarily acquired through Russian language education, a result of historical language policies during the Soviet era, when Russian was the dominant medium of communication in Kazakhstan. Nevertheless, the research on Kazakh text simplification remains limited. Classical rule-based approaches and early statistical methods show limited performance when applied to morphologically rich languages. In contrast, recent large-scale language models (LLMs) [18–21], pre-trained on multilingual corpora and fine-tuned for generative tasks, have demonstrated promising results on simplification tasks in other languages [6,11,12].

In this paper, we present a generative approach for Kazakh text simplification by fine-tuning multilingual large language models. To support model training, complex–simple sentence pairs are

 doi:10.20944/preprints202506.1947.v1

constructed through a heuristic-based method in a semi-synthetic manner. In the data construction process, the complexity of Kazakh words in a sentence was measured by their frequency and morphological complexity, and used these scores to filter common words from the text. The final selection of Kazakh complex sentences was performed using a heuristic approach, based on maximum token length, maximum allowed number of conjunctions, and the ratio of common words in the text.

Based on the assembled dataset, we fine-tuned various multilingual LLMs for the Kazakh text simplification task. We evaluated both instruction-tuned and LLMs, including domain-specific LLMs such as kazLLM and Sherkala, as well as general-purpose LLMs in a zero-shot setting. To benchmark performance, we introduce KazSim, an instruction-fine-tuned model optimized specifically for simplification.

Evaluations were conducted on two test sets: one automatically constructed from the same pipeline used during training, and another semi-manually curated benchmark designed to reflect more natural simplification patterns. We present results based on standard evaluation metrics such as BLEU, ROUGE, and SARI.

Experimental results showed that zero-shot and domain-adapted models are limited in their ability to produce structurally simplified and length-controlled outputs. In contrast, KazSim achieved consistently better scores across all metrics and evaluation settings, confirming the importance of task-specific supervision and targeted data construction for simplification in low-resource languages. We further investigated the impact of instruction language by comparing English and Kazakh prompts for the task. While most models showed comparable performance across both settings, slight gains were observed when prompts are given in Kazakh particularly for zero-shot and domain-specific models. KazSim remained stable under both prompt variants, confirming its robustness and suitability for multilingual deployment.

The main contributions of this work are as follows:

- First, we introduce a new parallel dataset for Kazakh sentence-level text simplification, constructed via a heuristic complexity identification pipeline and automatic simplification with large language models.
- Second, we present KazSim, a fine-tuned instruction-following simplification model based on multilingual LLM backbones, optimized specifically for Kazakh.
- Third, we provide a comprehensive evaluation against baseline Seq2Seq models, domain-specific LLMs, and zero-shot scenario, using both automatic and semi-manual test sets with multiple evaluation metrics.
- Finally, we analyze the impact of prompt language on simplification quality for Kazakh, offering practical insights for future multilingual LLM deployment.

2. Related Work

Text simplification aims to rewrite the original text into a simpler form while preserving its meaning. One simple way is to replace complex words in the sentences with simpler synonymous words; this process is referred to as lexical simplification. A more advanced method is to reduce the complexity of sentence structures, a process known as syntactic simplification. Existing approaches in these two directions can be categorized into three types: (i) rule-based methods, (ii) data-driven methods, and (iii) generative approaches.

In this direction, most existing studies [2–6] generally follow the following sequence of steps: i) identify complex words, ii) generate a set of candidate substitutions, iii) select the most contextually appropriate alternatives, and iv) rank the candidate substitutions according to their simplicity.

Early rule-based lexical simplification (LS) system [1] is proposed to simplify English newspaper texts for aphasic readers by combining syntactic analysis and simplification modules. It used linguistic analysis to generate synonym lists from WordNet [2], ranked by frequency from the Oxford Psycholinguistic Database, selecting the most common synonyms for output.

 doi:10.20944/preprints202506.1947.v1

Data-driven lexical simplification approaches use large parallel datasets of complex and simple texts and employ machine learning techniques to learn text simplification rules. In this direction, Drndarević and Saggion [4] conducted an empirical analysis of lexical simplification in Spanish using a parallel corpus of original and manually simplified texts. Their study identified lexical substitution as the most frequent operation and proposed a taxonomy including definition insertion and simplification of named entities and numerical expressions. The authors highlighted frequency and word length as key features for synonym selection, while emphasizing the importance of word sense disambiguation for handling polysemy.

Shardlow[5] investigates techniques for automatically identifying complex words, a critical yet often under-addressed component of lexical simplification. Using a corpus derived from Simple Wikipedia edit histories, the study compares methods that include full simplification, frequency thresholding, and a supervised classification approach based on support vector machines. The results indicate that, while machine learning slightly improves precision, it suffers from a substantial loss in recall. The work emphasizes the trade-offs involved in CW identification and highlights its foundational role in downstream simplification tasks.

For syntactic simplification, early approaches were also based on the hand-crafted rules [7], they framed the task as a two-step process of analysis and transformation, using handcrafted rules to split complex structures like relative clauses into simpler sentences. Siddharthan [8] introduced a framework for text simplification using transformation rules applied to typed dependency structures. The study compared different generation strategies and highlighted the trade-offs between preserving original sentence structure and relying on full surface realisation, emphasizing robustness to parsing errors as a key factor in simplification quality. Woodsend and Lapata [9] propose a data-driven approach to sentence simplification using quasi-synchronous grammar and integer linear programming. Their model captures structural mismatches and complex rewrite operations, selecting optimal simplifications from a space of candidate rewrites. Experimental results show that their method improves readability while preserving grammaticality and meaning, without relying on handcrafted rules.

Recent work for text simplification were employed sequence to sequence techniques and the generative approaches based on pre-trained large language models. Zhang and Lapata [10] introduce a deep reinforcement learning framework for sentence simplification, combining an encoder-decoder architecture with a reward function that promotes fluency, simplicity, and meaning preservation. Their model, DRESS, learns simplification rewrites from monolingual corpora and outperforms prior approaches across multiple benchmarks, highlighting the effectiveness of reinforcement learning for optimizing simplification quality. Mallinson et al. [11] introduce a multilingual, zero-shot framework for sentence simplification that transfers simplification knowledge from English to typologically distinct, low-resource languages in the absence of parallel corpora. Their model employs a shared transformer encoder with task- and language-specific layers trained via multi-task learning, enabling the construction of language-agnostic sentence representations. Empirical results on German datasets show that this approach yields higher-quality simplifications than both unsupervised baselines and multi-stage pivoting methods, illustrating the potential of crosslingual supervision for simplification in under-resourced settings.

Kew et al. [12] introduce BLESS, a comprehensive benchmark designed to systematically evaluate the sentence simplification capabilities of large language models (LLMs) for English, a highly resource language. Their analysis involved with many models of varying sizes and architectures, tested across multiple domains and prompted under few-shot settings. Their results indicate that many LLMs, including those not specifically trained for simplification, can match or exceed the performance of existing state-of-the-art systems, while also exhibiting broader coverage of simplification operations. Ryan et al. [13], MULTISIM a multilingual text simplification benchmark was introduced. Their work enables consistent evaluation across low-, medium-, and high-resource languages, and demonstrates that multilingual and few-shot models like BLOOM-176b can match or exceed the performance of fine-tuned models on non-English simplification tasks.

 doi:10.20944/preprints202506.1947.v1

3. Methodology

3.1. Task Formulation

In this work, we define Kazakh text simplification as a sequence-to-sequence generation task. The input is a complex sentence in Kazakh, and the goal is to generate its simplified version which is easier to read and understand, but still keeps the original meaning. The model learns to map the input sequence $x = (x_1, x_2, \ldots, x_n)$ to the output simplified sequence $y = (y_1, y_2, \ldots, y_m)$.

Formally, the model estimates the conditional probability:

$$P(y \mid x) = \prod_{t=1}^{m} P(y_t \mid y_{<t}, x) \tag{1}$$

where x is the complex input sentence and y is the simplified output. The variable t denotes the position index in the output sequence. At each time step t, the model predicts the token y_t based on the input x and the previously generated tokens $y_{<t} = (y_1, y_2, \ldots, y_{t-1})$.

The goal is to maximize the likelihood of generating the simplified sentence y given the input x.

3.2. Automatic Identification of Complex Kazakh Sentences

Figure 1 outlines a two-step process for selecting complex sentences from a Kazakh corpus. Several Kazakh books were pre-processed and tokenized for this process.

The first step focuses on identifying complex and common Kazakh words. In the second step, the text was split into sentences. Each sentence was evaluated using a heuristic method to decide whether it is complex. Sentences that meet the criteria were selected and saved as the final output.

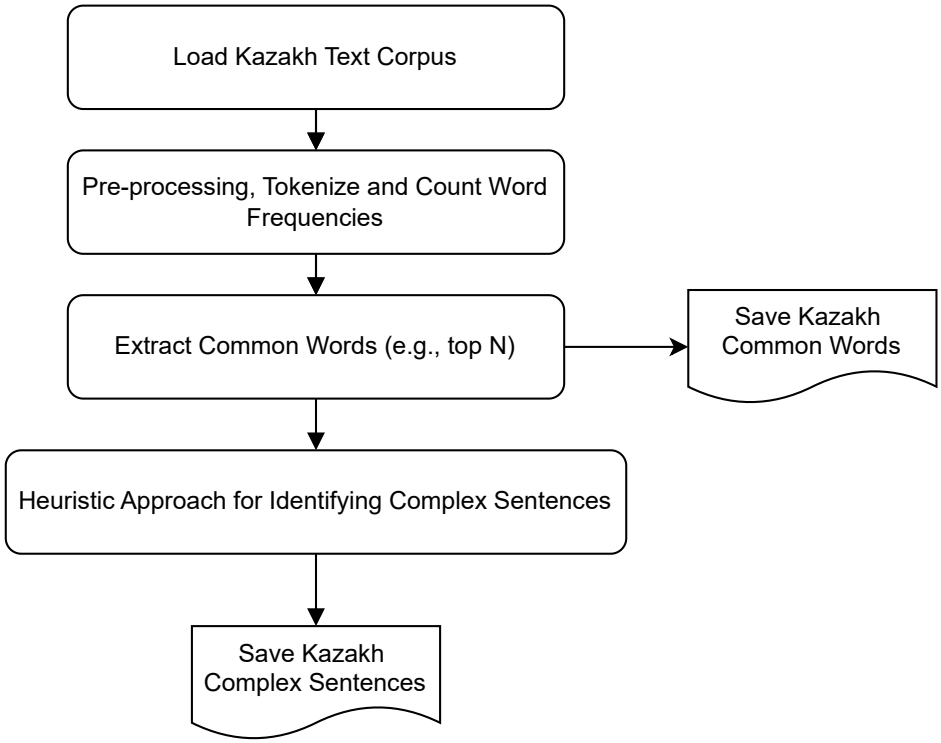

Figure 1. Pipeline for Selecting Complex Kazakh Sentences from Text Corpora.

To identify morphologically simple and high-frequency lexical words in Kazakh, we define a scoring mechanism that integrates corpus-driven frequency statistics with morphological complexity estimation. The underlying objective is to assign higher priority to words that are both frequently observed in naturally occurring texts and exhibit minimal morphological complexity.

 doi:10.20944/preprints202506.1947.v1

Morphological complexity is estimated through QazAnalyzer [14,17], a finite-state morphological analyzer developed for the Kazakh language. The analyzer decomposes surface word forms into a stem and a sequence of morphological suffixes. We approximate a word's structural complexity by computing the count of attached suffixes. In cases where the analyzer fails to produce a valid segmentation or classifies a token as "unknown", a predefined penalty is assigned to reflect maximal complexity, thereby discouraging the selection of such out-of-vocabulary items.

Formally, for a given word w, the scoring function is defined as:

$$\text{score}(w) = \text{morph}_{\text{score}}(w) - \alpha \cdot \log(\text{freq}(w)) \tag{2}$$

where $\text{morph}_{\text{score}}(w)$ denotes the number of suffixes extracted from the morphological analysis, and $\text{freq}(w)$ is the word frequency obtained from a large-scale corpus. The parameter $\alpha > 0$ is a tunable coefficient that adjusts the influence of frequency in the overall ranking.

Table 1. Pearson Correlation Coefficients Between Key Variables

Variable Pair	Correlation (r)	p-value
score vs. $\text{morph}_{\text{score}}$	0.558	$< 1e{-}300$
score vs. Log freq	-0.886	$< 1e{-}300$
$\text{morph}_{\text{score}}$ vs. Log freq	-0.120	1.084×10^{-158}

Table 1 presents Pearson correlation coefficients between final score, morphological complexity, and log frequency. A moderate positive correlation ($r = 0.558$) is observed between final score and morphological score, indicating that structurally complex words tend to receive higher scores. A strong negative correlation ($r = -0.886$) between final score and log frequency confirms that frequently used words are consistently ranked lower. The weak negative correlation between morphological score and log frequency ($r = -0.120$) suggests that morphologically rich forms occur less frequently.

Table 2. Top 20 simplest and most frequent kazakh words of the corpus. (Kazakh words are shown in Latin script for convenience.)

Word	Frequency	$\text{morph}_{\text{score}}$	Score
da	8309	1	-8.025095
dep	6124	1	-7.719971
osy	4618	1	-7.437717
edi	3287	1	-7.097731
emes	2436	1	-6.798113
de	6485	2	-6.777247
eken	2276	1	-6.730175
degen	2263	1	-6.724447
endi	2253	1	-6.720018
kop	2197	1	-6.694848
gana	2181	1	-6.687539
oz	2163	1	-6.679251
bir	5809	2	-6.667164
ozi	1851	1	-6.523481
birak	1808	1	-6.499977
ne	1753	1	-6.469084
bul	4413	2	-6.392310
goi	1576	1	-6.362645
abai	4106	2	-6.320205
bar	4078	2	-6.313362

 doi:10.20944/preprints202506.1947.v1

Table 2 shows the top 20 most frequent and morphologically simple Kazakh words based on the lowest final scores. Most have a morph score of 1, reflecting minimal suffixation, and occur frequently in the corpus. A few structurally richer forms (morph score 2) are included due to their high usage.

To extract complex sentences, we implement a rule-based filtering approach that identifies complex candidates from an input set. Algorithm 1 first tokenizes each sentence and computes three primary features: i) total number of tokens, ii) the count of known coordinating or subordinating conjunctions, and iii) the proportion of common words (extracted from the first step).

A sentence is considered structurally complex if it satisfies the following two constraints: 1) it either exceeds a specified token length threshold or contains a high number of conjunctions, and 2) the proportion of common words within the sentence falls below a predefined threshold. Only sentences meeting both conditions are retained. This selection strategy enables the construction of a filtered corpus consisting of lexically and syntactically complex sentences. After extracting all complex sentences, we used GPT-4 to simplify these sentences.

Algorithm 1: Extract Complex Kazakh Sentences

Input: List of sentences T,
Common words set C,
Conjunctions set K,
Maximum token length L,
Maximum allowed conjunctions J,
Common word ratio threshold r
Output: List of complex sentences R
$R \leftarrow$ empty list;
foreach *sentence* $s \in T$ **do**
  Tokenize s into *tokens*;
  $n \leftarrow$ length of *tokens*;
  $c \leftarrow$ number of tokens in *tokens* such that token $\in K$;
  $w \leftarrow$ number of tokens in *tokens* such that token $\in C$;
  $ratio \leftarrow \frac{w}{\max(n,1)}$;
  if $n > L$ ***or*** $c > J$ **then**
    if $ratio < r$ **then**
      Append s to R;
return R

3.3. Seq2Seq Model

As a baseline for Kazakh text simplification, we implement a standard sequence-to-sequence architecture based on long short term memory (LSTM) layers. The model consists of an encoder and decoder, both incorporating word embeddings and multi-layer LSTM networks. The encoder processes the input complex sentence and encodes it into a hidden representation. It includes an embedding layer followed by an LSTM, and the final hidden and cell states are passed to the decoder. The decoder generates the simplified sentence sequentially, using its own embedding and LSTM layers, followed by a linear projection to the output vocabulary. During training, we apply teacher forcing with a fixed ratio of 0.5. At each decoding step, the model receives either the ground-truth token or its own previous output. The objective is to minimize cross-entropy loss between predicted and reference tokens. Padding positions are masked during optimization.

This baseline provides a reference point for evaluating the performance gains introduced by instruction-tuned language models, and serves as a foundation for analyzing simplification behavior under low-resource conditions.

 doi:10.20944/preprints202506.1947.v1

3.4. Fine-tuning Kazakh Text Simplification

To improve over the baseline Seq2Seq model, we fine-tune large pre-trained language models (LLMs) for the Kazakh text simplification task. While the Seq2Seq model is effective for learning from aligned complex–simple sentence pairs, it often struggles with long-distance dependencies, rare morphological patterns, and fluency. Large language models, pre-trained on massive multilingual corpora, are better suited for such challenges. Their ability to generalize from limited fine-tuning data makes them a promising solution for low-resource languages like Kazakh.

We use three instruction-tuned LLMs in the experiments: Llama-3.2-3B, Llama-3.3-70B, and Qwen2-72B-Instruct. These models are selected due to their open access, multilingual support, and compatibility with parameter-efficient tuning. In this work, each model is evaluated in two modes: 1) zero-shot or instruction-only inference without additional fine-tuning, and 2) fine-tuning on our Kazakh simplification dataset. This allows us to analyze the effect of instruction tuning alone versus task-specific adaptation on simplification quality.

Due to the computational cost of full fine-tuning, we apply Low-Rank Adaptation (LoRA)[22] to all models. LoRA introduces small trainable matrices into the attention and feed-forward layers of the transformer without modifying the original weights. This allows for efficient training of large models on limited hardware. For convenience, we denote the fine-tuned LLMs for Kazakh text simplification as KazSim.

4. Experiments

4.1. Dataset

We selected 8709 sentence pairs for training and 500 for testing. Table 3 reports token and character-level statistics for complex and simple sentences across both splits. As expected, complex sentences are consistently longer than their simplified counterparts in terms of both token count and character length. Vocabulary size is higher in the training set due to scale, while type-token ratio is elevated in the test set, reflecting reduced repetition in smaller samples. These statistics provide a general overview of length and lexical variation prior to modeling. In addition to the test set described above, we include a semi-manually created test set with 163 of complex–simple sentence pairs for Kazakh text simplification for evaluation purposes.

Table 3. Token and character-level statistics for complex and simple sentences in each dataset split. TTR - type token ratio.

Dataset	avg_tokens	max_tokens	min_tokens	median_tokens	avg_chars	vocab_size	TTR
Train - Complex	21.32	421	5	19	148.94	50667	0.2728
Train - Simple	16.20	284	1	15	113.60	34791	0.2466
Test - Complex	22.47	446	7	20	158.42	6749	0.6007
Test - Simple	16.90	238	1	15.5	119.28	4770	0.5646

4.2. Baselines

We consider three categories of baselines: classical sequence-to-sequence models, domain-specific large language models trained on Kazakh data (kazLLM and Sherkala), and general-purpose LLMs evaluated in a zero-shot setting.

1) Seq2Seq refers to a standard encoder–decoder architecture built with LSTM layers. 2) Sherkala [16] is a domain-adapted LLM pretrained on a mix of Kazakh and multilingual corpora. It supports instruction-based prompting but has not been fine-tuned specifically for text simplification. We evaluate the Sherkala-Llama-3.1-8B model to establish a performance reference for general-purpose generation in a Kazakh-rich setting.

 doi:10.20944/preprints202506.1947.v1

3) kazLLM-Llama-3.1 are large language models trained with high-resource coverage of Kazakh, Russian, Turkish, and English data. Similar to Sherkala, it is evaluated without any simplification-specific tuning. We test both 8B and 70B variants to assess the role of scale in the absence of task alignment.

4) Zero-shot LLMs refer to publicly available instruction-tuned models (e.g., Llama-3.2-3B, Llama-3.3-70B and Qwen2-72B) used without any additional fine-tuning. These models are prompted using a standard instruction template, but are not specially adapted to Kazakh or to simplification. They provide an upper-bound baseline for off-the-shelf generation and allow us to assess the gap between general-purpose LLMs and models explicitly adapted to the target language and task.

4.3. Model Setup and Training

Two variants of the Seq2Seq models were explored: a small and a large version. Both models share the same overall architecture and training settings, including 2-layer LSTM encoder and decoder, a dropout rate of 0.3, a batch size of 128, and the use of the Adam optimizer with a learning rate of 1e-3. Early stopping is applied with a patience of 20 epochs to prevent overfitting. The small model uses an embedding dimension of 128 and a hidden size of 256, while the large model doubles these values with an embedding dimension of 256 and a hidden size of 512.

For all three LLM models, LoRA was applied to the transformer layers using rank-32 adapters with a dropout of 0.1. Fine-tuning is performed using the constructed training set of parallel complex and simplified Kazakh sentences. LLMs are trained with the AdamW optimizer, learning rate of 2e-5, a batch size of 8, and for 3 epochs. All preprocessing steps, including tokenization, follow the original tokenizer associated with each model.

Figure 2 shows the training loss comparison between small and large Seq2Seq models. The larger model converges significantly faster and achieves a lower final training loss, indicating greater learning capacity and better optimization behavior. In contrast, the small model converges more slowly and stabilizes at a higher loss, which may indicate underfitting due to limited model expressiveness. This comparison confirms that increased model capacity contributes positively to the learnability of the simplification task.

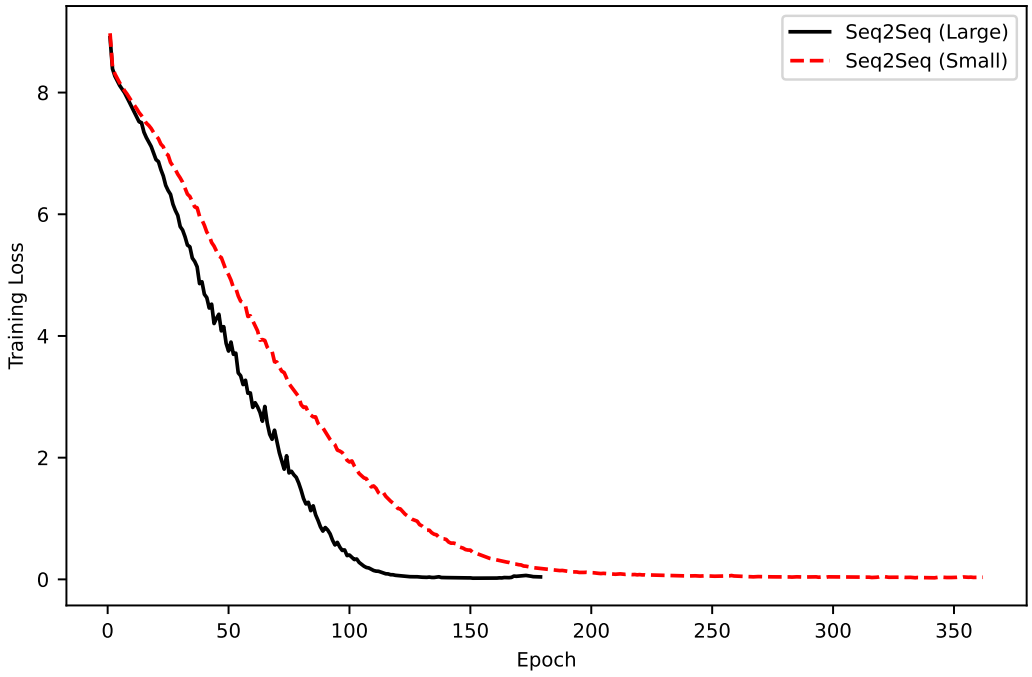

Figure 2. Training loss comparison between small and large Seq2Seq models.

 doi:10.20944/preprints202506.1947.v1

Figure 3 presents the training loss trajectories of the KazSim model fine-tuned on three large language models: Qwen2-72B, Llama-3.2-3B, and Llama-3.3-70B. Among them, Llama-3.3-70B achieved the lowest and most stable training loss throughout, indicating better alignment with the target simplification objective. The 3B model shows higher and more volatile loss, while Qwen2-72B demonstrates intermediate behavior.

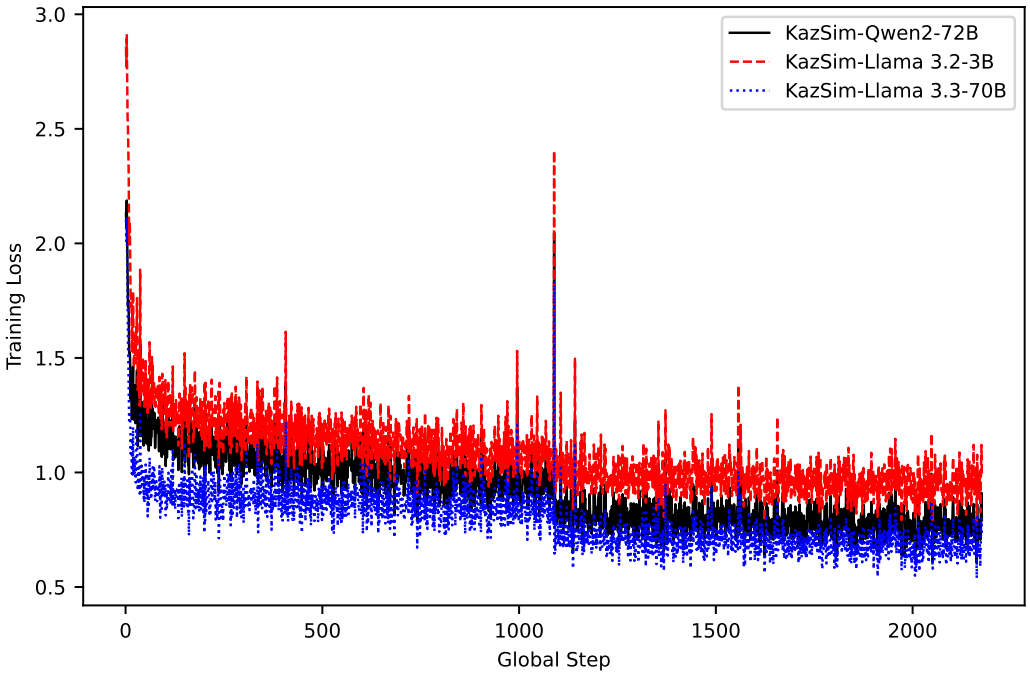

Figure 3. Training loss of trajectories of KazSim fine-tuned on three large language models.

4.4. Evaluation Metrics

To evaluate simplification quality, we use the following standard automatic metrics: BLEU, ROUGE-L, SARI, Bertscore. Each metric captures different aspects of output quality, including lexical overlap, structural alignment, and simplification-specific edits.

BLEU measures n-gram overlap between the system output and the reference. While originally developed for machine translation, it is widely used in simplification.

ROUGE-L computes the longest common subsequence between the prediction and the reference. Compared to BLEU, it is less sensitive to exact token matches and better reflects sentence-level alignment and fluency.

SARI is designed specifically for simplification. It evaluates the system output against both the reference and the original complex sentence, and scores three operations: addition, deletion, and retention.

BERTScore [15] leverages contextual embeddings from pretrained language models to compute similarity between candidate and reference sentences. It reports precision, recall, and F1 based on semantic alignment, and is particularly useful for capturing meaning preservation even when surface-level tokens differ.

4.5. Results

Table 4 and Table 5 present the evaluation results for baseline sequence-to-sequence models and various LLMs on the Kazakh text simplification task.

Table 4 presents evaluation results on the Kazakh text simplification dataset. Seq2Seq baselines fail to generalize, producing negligible BLEU and ROUGE scores, indicating traditional architectures lack sufficient capacity to model simplification under low-resource constraints. Zero-shot LLMs also struggle, with all variants producing overextended outputs (length ratios between 3.17 and 5.38).

 doi:10.20944/preprints202506.1947.v1

Table 4. Evaluation results on the Kazakh text simplification dataset.

Model	BLEU	ROUGE-1	ROUGE-L	Length Ratio	Precision	Recall	F1
Seq2Seq-small	0.0038	0.25%	0.25%	0.90	66.26	65.86	66.04
Seq2Seq-large	0.007	3.81%	3.63%	0.82	67.39	65.88	66.61
kazLLM-Llama-3.1-8B	20.72	42.09%	40.57%	1.22	82.52	84.51	83.45
kazLLM-Llama-3.1-70B	21.52	44.57%	43.70%	1.06	81.63	86.65	84.04
Sherkala-Llama-3.1-8B	19.59	42.35%	40.83%	1.17	82.58	85.25	83.85
Llama-3.2-3B (zero-shot)	0.0028	0.56%	0.57%	5.38	57.57	60.47	58.91
Qwen2-72B (zero-shot)	0.013	3.08%	2.97%	4.27	59.71	63.39	61.28
Llama-3.3-70B (zero-shot)	0.055	23.47%	22.47%	3.17	70.12	77.13	73.24
KazSim (Llama-3.2-3B)	25.7	47.01%	46.02%	0.99	84.97	85.61	85.25
KazSim (Qwen2-72B)	27.8	48.56%	47.60%	0.96	75.20	81.14	78.03
KazSim (Llama-3.3-70B)	**33.5**	54.21%	53.00%	0.98	**87.49**	**87.70**	**87.56**

Table 5. Evaluation results on the semi-manually created test set for Kazakh text simplification.

Model	BLEU	ROUGE-1	ROUGE-L	Length Ratio	Precision	Recall	F1
kazLLM-Llama-3.1-8B	17.09	38.80%	36.96%	1.04	82.52	83.13	82.78
kazLLM-Llama-3.1-70B	16.35	41.24%	39.59%	1.25	81.84	85.76	83.72
Sherkala-Llama-3.1-8B	17.08	40.61%	38.63%	1.13	82.80	84.40	83.55
Llama-3.2-3B (zero-shot)	0.002	0.67%	0.67%	5.60	57.11	59.96	58.42
Qwen2-72B (zero-shot)	0.012	3.51%	3.19%	4.30	60.33	64.11	61.94
Llama-3.3-70B (zero-shot)	0.045	20.71%	19.21%	3.06	69.80	74.98	72.08
KazSim (Llama-3.2-3B)	17.82	39.89%	38.49%	1.01	83.72	83.11	83.37
KazSim (Qwen2-72B)	18.31	40.16%	38.83%	0.97	75.20	81.14	78.03
KazSim (Llama-3.3-70B)	**20.33**	42.26%	40.50%	0.99	**84.76**	**83.87**	**84.27**

While Llama-3.3-70B shows moderate gains in BLEU (5.53) and F1 (73.24), the absence of length control limits overall performance.

Domain-specific models, including kazLLM and Sherkala, demonstrate stable performance across all metrics and clearly outperform both Seq2Seq baselines and zero-shot LLMs. BLEU scores range from 19.59 to 21.52, and all three configurations achieve F1 scores above 83, indicating strong surface-level fluency and content preservation. In contrast to zero-shot outputs, length ratios remain close to 1.0, confirming better control over generation length.

Among these models, kazLLM-70B achieves the highest BLEU score (21.52) and the highest recall (86.65), with a length ratio of 1.06. Sherkala-8B, while slightly behind in BLEU (19.59), achieves the highest precision (82.58) and a longer average output (length ratio = 1.17). kazLLM-8B falls between the two, with balanced precision and recall (82.52 / 84.51) and a BLEU score of 20.72, showing that smaller-scale models can still generalize well when exposed to sufficient domain-specific data.

KazSim models outperform all baselines. KazSim (Llama-3.3-70B) achieves the best overall results, with a BLEU of 33.5, F1 of 87.56, and a near-optimal length ratio of 0.98. Other KazSim variants (Qwen2-72B and Llama-3.2-3B) also perform well, confirming that targeted finetuning on task-specific Kazakh data is critical for achieving high-quality simplification.

Table 5 reports results on the semi-manually created test set for Kazakh text simplification. Compared to the generated test, performance patterns remain consistent, but scores are generally lower across all metrics. This drop suggests that the manually curated references are more diverse and structurally dissimilar from the model outputs, increasing the difficulty of achieving high lexical overlap.

Zero-shot models again show weak performance, with BLEU scores below 5 and F1 scores ranging from 58.42 to 72.08. Length ratios remain substantially inflated (3.06–5.60), indicating persistent overgeneration. Despite minor gains in F1, these models continue to underperform across all metrics, confirming the limitations of zero-shot simplification in low-resource settings.

 doi:10.20944/preprints202506.1947.v1

Domain-specific models, kazLLM and Sherkala, maintain relatively good performance. BLEU scores fall between 16.35 and 17.09, and F1 remains above 82 for all configurations. Length ratios range from 1.04 to 1.25, consistent with more controlled generation behavior.

KazSim again outperforms all other approaches. KazSim (Llama-3.3-70B) achieves the highest BLEU (20.33) and F1 (84.27), with a near-optimal length ratio of 0.99. Other KazSim variants (Llama-3.2-3B and Qwen2-72B) also perform strongly, confirming that the benefits of fine-tuning extend to harder test cases with more diverse simplification references.

In comparison to the automatic test split, all models show slightly reduced BLEU and ROUGE scores on the manual set. This suggests that the manual references contain more lexical and syntactic variation, reducing surface-level overlap. However, models like KazSim that are trained on task-aligned supervision still generalize well, with minimal drop in F1 and consistent length control. Overall, the results reinforce the robustness of KazSim across evaluation settings and confirm that simplification in low-resource languages requires explicit adaptation not only to the language but also to the task.

Figure 4 presents SARI scores for all models evaluated on both the automatic and semi-manual test sets. SARI is used as the primary metric for measuring simplification quality, as it captures the balance between content preservation, deletion of unnecessary information, and appropriate addition of simplified expressions.

Seq2Seq baselines perform the worst, with SARI scores of 33.56 and 33.60. These results reflect the inability of standard encoder-decoder models to generalize under low-resource scenarios. Zero-shot models demonstrate slight performance improvements, with scores ranging from 33.92 (Llama-3.2-3B) to 40.02 (Llama-3.3-70B). Notably, performance on the semi-manual test set remains stable or slightly improves across all zero-shot models. This suggests that zero-shot models are not particularly sensitive to test set construction and may rely on generic rewriting patterns that generalize equally across both test types.

Domain-specific models, including kazLLM and Sherkala, achieve higher SARI scores in the 42.86–45.80 range, and show relatively small gaps between the two test sets. This indicates better stability and improved simplification quality when models are pretrained on Kazakh data.

KazSim models achieve the highest scores overall. KazSim (Llama-3.3-70B) reaches 56.38 on the automatic test set and 48.42 on the semi-manual set. Other KazSim variants (Qwen2 and Llama-3.2-3B) also outperform all baselines and zero-shot models. Although there is a drop in SARI when moving to the manual test set, KazSim maintains a clear advantage, showing that task-specific supervision enables better generalization.

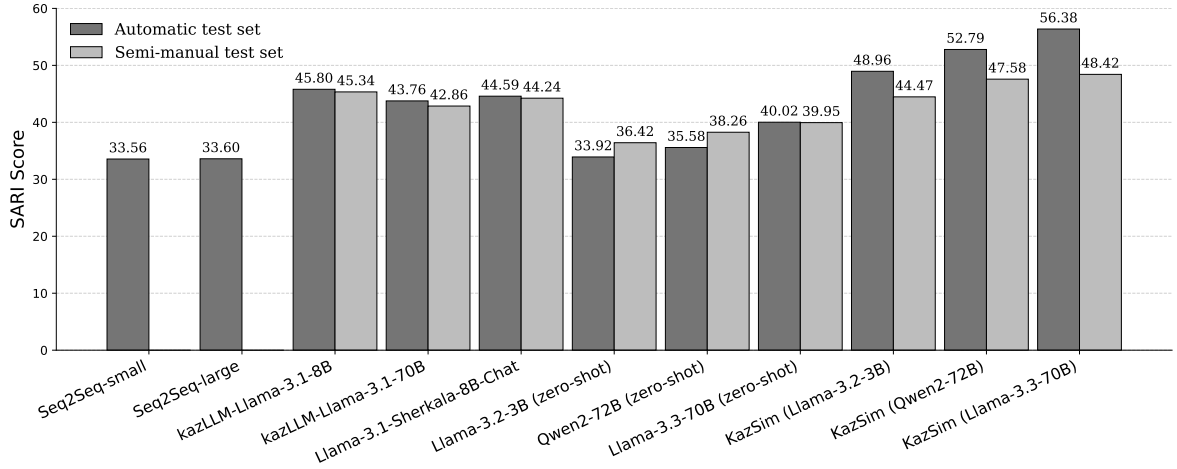

Figure 4. Results of Sari score of various models for two different datasets.

 doi:10.20944/preprints202506.1947.v1

Table 6. Comparison of model performance under English and Kazakh instruction prompts. Metrics reported on the semi-manual test set include BLEU, SARI, and F1.

Model	English prompt			Kazakh prompt		
	BLEU	SARI	F1	BLEU	SARI	F1
Sherkala-Llama-3.1-8B	17.08	44.23	83.55	15.78	46.15	82.66
kazLLM-Llama-3.1-70B	16.34	42.85	83.72	16.22	42.30	83.62
kazLLM-Llama-3.1-8B	17.09	45.34	82.78	15.85	46.42	82.48
Llama-3.3-70B (zero-shot)	4.53	39.95	72.08	8.26	40.73	76.42
KazSim (Llama-3.3-70B)	20.32	48.42	84.27	21.03	48.78	84.39

To assess prompt sensitivity, we compare model outputs when using either English or Kazakh instruction prompts for the same simplification task. Results are summarized in Table 6. Overall, model performance remains relatively stable across prompt languages, though minor variations are observed. For domain-specific models (e.g., kazLLM and Sherkala), Kazakh prompts yield slightly higher SARI scores, indicating better alignment with simplification objectives when instructions are provided in the target language. Sherkala, for instance, improves from 44.23 to 46.15 in SARI with a Kazakh prompt. Zero-shot performance is more sensitive to prompt language. Llama-3.3-70B sees a notable increase in BLEU (from 4.53 to 8.26) and F1 (from 72.08 to 76.42) under Kazakh instructions, suggesting that instruction-following behavior improves when prompts are better aligned with the target output language. The proposed KazSim model remains robust under both conditions, achieving the best results across all metrics. Performance is slightly higher with Kazakh prompts, reaching a SARI of 48.78 and an F1 of 84.39, confirming the benefit of aligning the prompt language with the generation task. These findings suggest that while multilingual LLMs can generalize across instruction languages, prompt language alignment can further improve output quality particularly for zero-shot settings and domain-specific models trained on Kazakh data.

5. Conclusion

This work presents a comprehensive study of sentence-level text simplification for Kazakh, a low-resource and morphologically rich language. To support training, we construct a parallel simplification dataset by first identifying complex sentences using a combination of frequency-based and morphological features. For each selected complex sentence, a corresponding simplified version is generated using LLMs enabling scalable data creation without full manual annotation. In addition to the test set, we also created a semi-manual test set of complex–simple sentence pairs for Kazakh text simplification for real practical evaluation purposes. The proposed model, KazSim, is trained via instruction tuning on top of various Llama-3.3 and Qwen2 backbones and evaluated alongside a diverse set of baselines, including classical Seq2Seq models, Kazakh domain-specific LLMs, and zero-shot instruction-following models.

We first evaluate all models on the automatically constructed test set, derived from the same pipeline used for training data generation. Standard Seq2Seq models perform poorly, with BLEU scores below 0.01 and negligible ROUGE values. These results highlight the limited capability of classical encoder–decoder architectures to handle simplification in low-resource settings without external supervision or pretraining.

Zero-shot models show limited improvements, with Llama-3.3-70B achieving the highest BLEU (5.53) and SARI (40.02) in this category. However, length ratios remain high (3.17–5.38), indicating uncontrolled output length and overgeneration. Precision and recall are also lower than those of task-tuned models, confirming the limitations of zero-shot approaches in structure-sensitive tasks like simplification. Domain-specific models such as kazLLM and Sherkala produce more fluent and compact outputs, with BLEU scores around 20 and F1 scores exceeding 83. Among these, kazLLM-70B reaches the highest BLEU (21.52) and recall (86.65), with a length ratio of 1.06, indicating stronger alignment with reference length. KazSim outperforms all baselines across the board. The best configuration,

 doi:10.20944/preprints202506.1947.v1

KazSim based on Llama-3.3-70B, achieves the highest BLEU (33.5), SARI (56.38), and F1 (87.56), while maintaining a near-optimal length ratio of 0.98. Other KazSim variants also show strong performance, confirming the benefits of instruction tuning on task-specific data.

We further evaluate all models on a semi-manually created benchmark designed to reflect more natural simplification patterns. zero-shot models continue to under-perform. BLEU scores remain low ranging from 0.29 to 4.53 and length ratios remained high, indicating persistent over-generation. Domain-specific models such as kazLLM-Llama-3.1-8B, kazLLM-Llama-3.1-70B, and Sherkala-8B demonstrate stable behavior, with BLEU scores between 16.35 and 17.09 and F1 scores exceeding 82. The best-performing configuration KazSim based on Llama-3.3-70B achieves a BLEU score of 20.33, F1 of 84.27, and maintained a balanced length ratio of 0.99.

Overall, these results confirm that strong simplification performance in low-resource settings cannot be achieved through scale or domain adaptation alone. While general-purpose and domain-specific LLMs produce fluent outputs, they struggle with structure, length control, and simplification-specific alignment. In contrast, KazSim, fine-tuned with instruction-level supervision on training data, consistently yields better output quality across both evaluation settings.

We also evaluate the effect of prompt language by comparing performance under English and Kazakh instructions. While the differences are generally small, models show slightly better results when prompted in Kazakh. This trend is more visible for zero-shot and domain-adapted models, which benefit from alignment between instruction and output language. KazSim remained stable across both settings, confirming its robustness to prompt variation. These findings suggest that prompt formulation plays a role in model behavior, especially in multilingual setups where instruction language may influence generation quality.

Future directions include expanding the dataset with more diverse simplification cases and exploring controlled-generation settings such as step-wise simplification or simplification conditioned on target length. We also plan to extend KazSim to multilingual scenarios and incorporate preference-based evaluation to support fine-grained feedback.

Author Contributions: Conceptualization, A.T. and G.T.; methodology, A.T.; software, A.T., G.T.; validation, A.T., G.T. and I.U.; formal analysis, A.T.; investigation, G.T.; resources, A.T.,G.T.; data curation, A.T.,G.T.; writing—original draft preparation, A.T.; writing—review and editing, A.T.,G.T.; visualization, A.T.,G.T.; supervision, A.T.,G.T.; project administration, A.T.,I.U.; funding acquisition, I.U. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Committee of Science of the Ministry of Science and Higher Education of the Republic of Kazakhstan under grant number AP19680575.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The dataset consisting of Kazakh complex–simple sentence pairs, which is released in this repository: https://github.com/a-toleu/KazSim/tree/main

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. John Carroll, Guido Minnen, Yvonne Canning, Siobhan Devlin, and John Tait. Practical simplification of English newspaper text to assist aphasic readers. In *Proceedings of the AAAI-98 workshop on integrating artificial intelligence and assistive technology*, pages 7–10, Madison, WI, 1998.
2. George A. Miller. WordNet: A Lexical Database for English. In *Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*, 1994. https://aclanthology.org/H94-1111/
3. Matthew Shardlow. Out in the Open: Finding and Categorising Errors in the Lexical Simplification Pipeline. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 1583–1590, Reykjavik, Iceland, May 2014. European Language Resources Association (ELRA). https://aclanthology.org/L14-1403/

 doi:10.20944/preprints202506.1947.v1

4. Biljana Drndarević and Horacio Saggion. Towards Automatic Lexical Simplification in Spanish: An Empirical Study. In *Proceedings of the First Workshop on Predicting and Improving Text Readability for target reader populations*, pages 8–16, Montréal, Canada, June 2012. Association for Computational Linguistics. https://aclanthology.org/W12-2202/
5. Matthew Shardlow. A Comparison of Techniques to Automatically Identify Complex Words. In *51st Annual Meeting of the Association for Computational Linguistics Proceedings of the Student Research Workshop*, pages 103–109, Sofia, Bulgaria, August 2013. Association for Computational Linguistics. https://aclanthology.org/P13-3015/
6. Lucía Ormaechea, Nikos Tsourakis, Didier Schwab, Pierrette Bouillon, and Benjamin Lecouteux. Simple, Simpler and Beyond: A Fine-Tuning BERT-Based Approach to Enhance Sentence Complexity Assessment for Text Simplification. In *Proceedings of the 6th International Conference on Natural Language and Speech Processing (ICNLSP 2023)*, pages 120–133, Online, December 2023. Association for Computational Linguistics. https://aclanthology.org/2023.icnlsp-1.12/
7. R. Chandrasekar, Christine Doran, and B. Srinivas. Motivations and Methods for Text Simplification. In *COLING 1996 Volume 2: The 16th International Conference on Computational Linguistics*, 1996. https://aclanthology.org/C96-2183/
8. Advaith Siddharthan. Text Simplification using Typed Dependencies: A Comparison of the Robustness of Different Generation Strategies. In *Proceedings of the 13th European Workshop on Natural Language Generation*, pages 2–11, Nancy, France, September 2011. Association for Computational Linguistics. https://aclanthology.org/W11-2802/
9. Kristian Woodsend and Mirella Lapata. Learning to Simplify Sentences with Quasi-Synchronous Grammar and Integer Programming. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 409–420, Edinburgh, Scotland, UK, July 2011. Association for Computational Linguistics. https://aclanthology.org/D11-1038/
10. Xingxing Zhang and Mirella Lapata. Sentence Simplification with Deep Reinforcement Learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 584–594, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. https://aclanthology.org/D17-1062/, doi:10.18653/v1/D17-1062.
11. Jonathan Mallinson, Rico Sennrich, and Mirella Lapata. Zero-Shot Crosslingual Sentence Simplification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5109–5126, Online, November 2020. Association for Computational Linguistics. https://aclanthology.org/2020.emnlp-main.415/, doi:10.18653/v1/2020.emnlp-main.415.
12. Tannon Kew, Alison Chi, Laura Vásquez-Rodríguez, Sweta Agrawal, Dennis Aumiller, Fernando Alva-Manchego, and Matthew Shardlow. BLESS: Benchmarking Large Language Models on Sentence Simplification. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13291–13309, Singapore, December 2023. Association for Computational Linguistics. https://aclanthology.org/2023.emnlp-main.821/, doi:10.18653/v1/2023.emnlp-main.821.
13. Michael J. Ryan, Tarek Naous, and Wei Xu. Revisiting non-English Text Simplification: A Unified Multilingual Benchmark. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4898–4927, Toronto, Canada, July 2023. Association for Computational Linguistics. https://aclanthology.org/2023.acl-long.269/, doi:10.18653/v1/2023.acl-long.269.
14. Gulmira Tolegen, Alymzhan Toleu, and Rustam Mussabayev. A Finite State Transducer Based Morphological Analyzer for The Kazakh Language. In *2022 7th International Conference on Computer Science and Engineering (UBMK)*, pages 01–06, 2022. doi:10.1109/UBMK55850.2022.9919445.
15. Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. BERTScore: Evaluating Text Generation with BERT. *arXiv preprint arXiv:1904.09675*, 2019. https://api.semanticscholar.org/CorpusID:127986044
16. Fajri Koto, Rituraj Joshi, Nurdaulet Mukhituly, Yuxia Wang, Zhuohan Xie, Rahul Pal, Daniil Orel, Parvez Mullah, Diana Turmakhan, Maiya Goloburda, Mohammed Kamran, Samujjwal Ghosh, Bokang Jia, Jonibek Mansurov, Mukhammed Togmanov, Debopriyo Banerjee, Nurkhan Laiyk, Akhmed Sakip, Xudong Han, Ekaterina Kochmar, Alham Fikri Aji, Aaryamonvikram Singh, Alok Anil Jadhav, Satheesh Katipomu, Samta Kamboj, Monojit Choudhury, Gurpreet Gosal, Gokul Ramakrishnan, Biswajit Mishra, Sarath Chandran, Avraham Sheinin, Natalia Vassilieva, Neha Sengupta, Larry Murray, and Preslav Nakov. Llama-3.1-Sherkala-8B-Chat: An Open Large Language Model for Kazakh. *arXiv preprint arXiv:2503.01493*, 2025. https://arxiv.org/abs/2503.01493

 doi:10.20944/preprints202506.1947.v1

17. Alymzhan Toleu, Gulmira Tolegen, and Rustam Mussabayev. Language-Independent Approach for Morphological Disambiguation. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 5288–5297, Gyeongju, Republic of Korea, October 2022. International Committee on Computational Linguistics. https://aclanthology.org/2022.coling-1.470/
18. Gulmira Tolegen, Alymzhan Toleu, Rustam Mussabayev, Bagashar Zhumazhanov, and Gulzat Ziyatbekova. Generative Pre-Trained Transformer for Kazakh Text Generation Tasks. In *2023 19th International Asian School-Seminar on Optimization Problems of Complex Systems (OPCS)*, pages 144–118, 2023. doi:10.1109/OPCS59592.2023.10275765.
19. Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv preprint arXiv:1810.04805*, 2019. https://arxiv.org/abs/1810.04805
20. Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and Efficient Foundation Language Models. *arXiv preprint arXiv:2302.13971*, 2023. https://arxiv.org/abs/2302.13971
21. Qwen, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 Technical Report. *arXiv preprint arXiv:2412.15115*, 2025. https://arxiv.org/abs/2412.15115
22. Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-Rank Adaptation of Large Language Models. *arXiv preprint arXiv:2106.09685*, 2021. https://arxiv.org/abs/2106.09685