DOI 10.2478/jazcas-2025-0037

TRAINING OF LARGE LANGUAGE MODEL MISTRAL ON SLOVAK LANGUAGE DATA

PETER BEDNÁR[1] – MAREK DOBEŠ[2] – RADOVAN GARABÍK[3]
[1]Faculty of Electrical Engineering and Informatics of the Technical University of Košice, Košice, Slovakia
[2]Centre of Social and Psychological Sciences, Slovak Academy of Sciences, Košice, Slovakia
[3]Ľ. Štúr Institute of Linguistics, Slovak Academy of Sciences, Bratislava, Slovakia

BEDNÁR, Peter – DOBEŠ, Marek – GARABÍK, Radovan: Training of large language model Mistral on Slovak language data. Jazykovedný časopis (Journal of Linguistics), 2025, Vol. 76, No. 2, pp. 433–451.

Abstract: This study investigates the adaptation of the Mistral 7B large language model for the Slovak language, addressing the limited availability of high-quality open-source models for low-resource languages. While commercial models like GPT-4 and Claude exhibit strong Slovak proficiency, their proprietary nature restricts transparency and customization. To overcome this, we fine-tuned the open-weight Mistral 7B model using the Araneum Slovacum VII Maximum corpus (5.3 billion tokens), creating a specialized Slovak variant, Mistral-SK-7b. The training, conducted on the Leonardo supercomputer (10,000 GPU hours), yielded significant improvements: the fine-tuned model generates grammatically correct and contextually coherent Slovak text, eliminating the errors (code-switching, repetition loops, and lexical interference from other languages) observed in the original Mistral-7B-v0.1. The resulting model, released under the Apache 2.0 license, provides a publicly accessible resource for Slovak NLP applications while preserving the base model's multilingual capabilities. Our work demonstrates the feasibility of adapting state-of-the-art LLMs for linguistically underrepresented languages and underscores the role of open models in promoting digital language preservation.

Keywords: large language models, Mistral, computational linguistics, Slovak language, Natural Language Processing, model fine-tuning

1. INTRODUCTION

1.1 Large Language Models in Natural Language Processing

Large language models (LLMs) are deep neural networks designed to learn and generate text in natural language. They achieve this through extensive training on vast amounts of textual data, allowing them to discern and create patterns that resemble human language. The training process involves analyzing numerous texts, from which the models build statistical relationships among letters, tokens, words, sentences, and larger text structures. As a result, LLMs are capable of producing coherent and contextually appropriate text.

The architecture of LLMs typically involves several layers of neural networks, each layer responsible for processing different aspects of the text. The initial layers might focus on identifying patterns at the letter or word level, while the deeper layers understand sentence structure and context.

One of the critical advancements in LLMs is the use of transformers, a type of neural network architecture that allows for better handling of long-range dependencies in text (Vaswani et al. 2017). Transformers have revolutionized the field by enabling models to consider the entire context of a sentence or paragraph, rather than processing text sequentially. This innovation has significantly improved the performance of LLMs in various natural language processing tasks, such as translation, summarization, and question answering (Devlin et al. 2019).

Training LLMs requires massive computational resources and access to extensive text corpora. For example, OpenAI's GPT-3 model was trained on hundreds of billions of words from diverse sources, including books, articles, and websites (Brown et al. 2020). The scale of data and computation involved in training these models allows them to capture a wide range of linguistic nuances and knowledge.

1.2 Accessibility and Limitations of Large Language Models

The most extensive language models currently available, such as GPT-4 and Claude, are primarily accessible only to the companies that develop them. These companies provide public access to their models under specific terms and conditions. For instance, OpenAI's GPT-4 is accessible through an API or a web based chat interface, but usage is governed by pricing, rate limits, and other restrictions set by OpenAI (OpenAI 2023). Similarly, Anthropic's Claude is another example of a powerful language model available to users under the conditions defined by its creators (Anthropic 2023).

One significant drawback of this proprietary model approach is the sudden switch to the lack of transparency. While some of the first generative LLMs were often rather transparent (including a description of the training data, training algorithms and parameters used for training), the landscape of the most performant LLMs changed to completely opaque. The broader community lacks detailed knowledge about the architecture, limitations, and training data of these models. This opacity can hinder academic research and independent verification of the models' capabilities and biases (Bommasani et al. 2021). Without access to the underlying details, it's challenging to fully understand how these models operate, their potential biases, or the robustness of their outputs. It is often not even possible to find out the proportion of multilingual data in the models, which makes it difficult to meaningfully compare performance across the models in less resourced languages (such as Slovak).

Because of these limitations, there is a growing movement towards the availability of open models. These open-source models can be downloaded,

modified, and used by anyone under creative licenses. This openness facilitates a more inclusive and collaborative approach to model development and usage, allowing for community-driven improvements and adaptations.

The availability of open models also democratizes access to advanced language processing capabilities, enabling smaller organizations and individual developers to leverage state-of-the-art tools without prohibitive costs. These open models can be fine-tuned for specific tasks, making them highly adaptable to various applications, from academic research to commercial use.

1.3 Multilingual Large Language Models and Slovak Language

While most language models are predominantly trained in English, there is a growing number of models that support multiple languages, including Slovak. Proprietary models such as GPT-4, Claude, and Gemini have shown good proficiency in Slovak due to extensive training on diverse multilingual datasets (OpenAI 2023; Anthropic 2023; Google 2024). These models benefit from the substantial resources and computational power available to their developers, enabling them to achieve high-quality outputs across various languages.

In addition to these closed models, there are also open-source models supporting Slovak, such as Gemma-3-4b-it (Google 2025) or Llama-3.3-70B-Instruct (Meta 2025). These open models, however, still face several challenges and exhibit notable deficiencies in their Slovak language capabilities. One significant factor contributing to this discrepancy is the limited availability of publicly accessible Slovak text data. The scarcity of high-quality, diverse training data in non-English languages, including Slovak, constrains the model's ability to learn and generalize effectively (Blasi et al. 2022).

Moreover, the development of open models often lacks the extensive fine-tuning and quality control processes that proprietary models undergo. Fine-tuning for specific languages, such as Slovak, requires significant time, expertise, and computational resources, which may not be as readily available in the open-source community (Touvron et al. 2023). As a result, the Slovak outputs from these open models may not yet match the quality of those produced by their closed counterparts.

To address these gaps, we present the following contributions:

- Slovak language adaptation: We fine-tune the Mistral 7B model – a state-of-the-art open-weight LLM – using the Araneum Slovacum VII Maximum corpus (5.3B tokens), creating Mistral-SK-7b, the first publicly available Slovak-tuned model of this scale.
- Performance validation: Our qualitative evaluation demonstrates that the fine-tuned model eliminates grammatical errors and code-switching artifacts prevalent in the base Mistral-7B-v0.1, achieving near-native fluency in Slovak text generation.
- Resource-efficient methodology: By leveraging the Leonardo supercomputer (10k GPU hours), we optimize the training process while preserving the mod-

el's multilingual capabilities, providing a blueprint for adapting LLMs to other low-resource languages.

- Open-access release: The model is distributed under the Apache 2.0 license, enabling community use and further research in Slovak NLP applications.

2. RESOURCES

2.1 Decision on Model Training Strategy for Slovak Language

In our research, we faced a critical decision regarding the development of a Slovak language model: whether to train a native model from scratch or to fine-tune an existing model using Slovak text data. Training a model from the ground up demands an enormous amount of training data and computational resources. This process involves collecting and preprocessing a vast corpus of Slovak text, as well as dedicating significant time and computational power to the training process.

Given these substantial requirements, we opted to utilize an existing pre-trained model and fine-tune it with Slovak-specific data. This approach leverages the comprehensive training that the original model has already undergone, which includes learning general language patterns and structures (Radford et al. 2019). Fine-tuning enables us to adapt the model specifically to Slovak, using a smaller, more manageable dataset while still achieving high-quality results (Howard – Ruder 2018).

2.2 Mistral 7B Large Language Model

Mistral is a family of state-of-the-art language models in several sizes. The smallest version Mistral 7B (Mistral AI, 2023) has 7.3 billion parameters and is designed to offer high performance despite its relatively compact size. The model leverages advanced architectural features such as Grouped-Query Attention (GQA) and Sliding Window Attention (SWA) to enhance both its speed and ability to handle long sequences efficiently.

A key feature of Mistral 7B (Mistral 7B v0.1 that we use in our research) is its accessibility under an open-source Apache 2.0 license, which encourages broad usage and adaptation. This stands in contrast to many proprietary models that restrict usage through licensing terms. The open-weight nature of Mistral 7B means it can be fine-tuned for specific applications, providing flexibility for developers and researchers to tailor the model to their needs. On the other hand, the composition of training data (including the amount of texts in specific languages) has been deliberately kept undisclosed by the authors, thus limiting its transparency.

We decided to utilize the Mistral 7B model for several key reasons. Firstly, Mistral 7B is an open model, which allows us to fine-tune it with our selected data. The open-access nature of Mistral 7B, makes it an ideal candidate for customization and specific applications. This flexibility is crucial for adapting the model to handle Slovak text effectively.

Secondly, the model's parameter count is suitable for training with the computational resources and data we have available. Mistral 7B, with its 7.3 billion parameters, strikes a balance between complexity and manageability, enabling efficient training without requiring the extensive resources needed for much larger models like GPT-3, which has 175 billion parameters (Brown et al. 2020), and an efficient inference even on low end hardware.

Thirdly, Mistral 7B has been apparently trained only on sparse Slovak data. Starting with a model that lacks heavy prior exposure to Slovak allows us to monitor the training process closely and observe the development of structures responsible for understanding and generating Slovak text. This real-time insight into the model's learning process can be invaluable for refining and optimizing our fine-tuning approach (Radford et al. 2019).

Here are some key aspects of the model's architecture and parameters:

- Transformer Architecture: Mistral 7B is based on a decoder-only Transformer architecture. This structure is widely used in modern language models for its effectiveness in handling sequential data and generating coherent text.
- Grouped-Query Attention (GQA): This mechanism allows for faster inference and reduces memory usage by efficiently managing how queries, keys, and values are processed in the attention layers. It optimizes the attention calculation, making it suitable for large-scale models.
- Sliding Window Attention (SWA): SWA enables the model to handle longer context windows more effectively. It uses a sliding window approach to maintain a fixed-size cache, allowing the model to process sequences up to 128K tokens in length theoretically, although it typically uses an 8K context during training.
- Byte-fallback BPE Tokenizer: This tokenizer ensures that all characters are represented within the model's vocabulary, preventing out-of-vocabulary issues which are common in language models.
- Model Parameters: Mistral 7B contains approximately 7.3 billion parameters. It includes 32 attention heads, each with a dimension of 128, and a total hidden size of 4096. The model also uses rotary position embeddings to encode positional information effectively.

2.3 Training Data for Slovak Language Model

For training our Slovak language model, we are using the Araneum Slovacum VII Maximum web corpus, a state-of-the-art web corpus of Slovak language. The corpus forms a part of the ARANEA family of web corpora for two dozen languages, compiled by the Ľudovít Štúr Institute of Linguistics, Slovak Academy of Sciences, and the UNESCO Chair in Plurilingual and Multicultural Communication, Comenius University in Bratislava (Benko 2024). The size of the

Slovak corpus is approximately 5.3 billion tokens[1], 4.4 billion words, 280 million sentences, 106 million paragraphs, 14 million documents (i.e. web pages), offering a substantial volume of text necessary for effective model training, and at the time we started to train the model, it was the largest Slovak language corpus available to us for the training.

The corpus is built upon data that are being collected from 2011, with the latest collection in October 2023, thus offering reasonably up to date information, but also a vast amount of texts from recent history. The corpus is deduplicated on a document (i.e. webpage) and a paragraph level, and filtered for "reasonable quality" language (i.e. without texts lacking diacritics). Moreover, it is automatically lemmatized, POS tagged and morphosyntactically analyzed (features that are not used in LLM training).

Table 1 lists selected currently existing big or open-access corpora for Slovak. There is a significant overlap in the two web corpora in the table, the third one (ARANEUM + HPLT + FineWeb 0.1) is a union of existing three big Slovak web datasets, deduplicated on paragraph level and is currently the biggest existing general language corpus of Slovak (Garabík 2025; Webový korpus slovenčiny 2025).

corpus	*size [million tokens]*	*license*	*type*
Araneum Slovacum VII Maximum	5300	MIT/CC-0	web corpus
HPLT 2.0 Slovak Corpus v0.4	4090	CC-0	web corpus
ARANEUM + HPLT + FineWeb 0.1	104720	MIT/CC-0/ODC-by	web corpus
wiki-2019-08	50	CC-BY SA	Wikipedia & Necyklopédia
Corpus of Court Proceedings 3.0	11974	exempt from copyright	specialized
Corpus of Legal Texts 1.9	45	exempt from copyright	specialized
prim-11-public-all	1859	unavailable	national corpus

Table 1: List of selected big Slovak corpora. *Size* is in linguistic tokens (i.e. words + punctuation); *license* is the license of the package and annotation, not the content.

In addition to leveraging these existing corpora, we have initiated negotiations with various institutions in Slovakia that may have access to additional digitalized Slovak texts. These efforts aim to further expand our dataset, thereby enhancing the robustness and accuracy of our model. Access to a broader range of texts will be particularly valuable for fine-tuning and validating the model across diverse linguistic contexts and domains.

By combining publicly available linguistic resources with potential new sources of digitalized texts, we aim to develop a comprehensive and high-quality Slovak

[1] By a "token" here we mean either a word, number or a punctuation character, as usual in corpus linguistics. We feel the need to explicitly point this out, given a different usage of the term "token" in the context of LLMs.

language model. This approach not only maximizes the use of existing data but also opens opportunities for collaboration and further research in the field of computational linguistics.

2.4 Computational Resources for Model Training

To facilitate the training of our Slovak language model, we secured a grant from the National Supercomputing Centre Slovakia[2], which provides us with GPU computing access on the European supercomputer Leonardo. Leonardo is one of the most powerful supercomputers in Europe, ranking among the top in the Top500 list. It features NVIDIA Ampere A100 GPUs, each equipped with 64GB of high-bandwidth memory (HBM2e). This configuration is particularly advantageous for training large language models due to its high memory capacity and efficient data handling capabilities (HPC Cineca 2023).

Leonardo's architecture includes both Booster and Data-centric modules. The Booster module comprises BullSequana X2135 blades, each with a single Intel Xeon 8358 CPU and four NVIDIA A100 GPUs. This setup provides substantial computational power and memory, essential for handling extensive training datasets and complex model architectures. The Data-centric module, featuring Intel Sapphire Rapids CPUs, complements the Booster module by supporting a wide range of applications with high computational demands.

2.5 Virtual Environment and Dependencies

To run our Slovak language model effectively on the Leonardo supercomputer, we set up a specific environment to utilize the available computational resources efficiently. Here's an overview of the necessary environment and dependencies:

Module Loading:

We begin by loading essential modules that provide the necessary software environment for deep learning tasks. This includes a specific Python version and CUDA for GPU acceleration. The relevant modules are:

- profile/deeplrn: A module that provides an optimized environment for deep learning applications.
- python/3.11.6--gcc--8.5.0: This module loads Python version 3.11.6, compiled with GCC 8.5.0, ensuring compatibility with high-performance computing applications.
- cuda: CUDA is essential for leveraging the GPU capabilities of the Leonardo supercomputer, which uses NVIDIA A100 GPUs with 64GB of HBM2e memory.

[2] https://nscc.sk

Python Virtual Environment:

To manage dependencies, we create a Python virtual environment. This isolated environment helps to avoid conflicts between different package versions and system-wide Python packages.

Package Installation:

Within the virtual environment, several key Python packages are installed to support the model training and execution:

- wheel and setuptools: These are fundamental packages for building and distributing Python packages.
- torch and accelerate: PyTorch is a popular deep learning framework, and Accelerate helps in optimizing the training process across multiple GPUs.
- datasets: This library provides easy-to-use tools for loading and preprocessing datasets.
- transformers: Essential for working with transformer-based models, such as our language model.
- flash-attn: Optimized attention mechanisms for faster model training.
- sentencepiece: A tokenizer that handles the text preprocessing needed for our language model.
- bitsandbytes: Useful for optimizing memory usage during model training.
- tensorboardX: Allows us to visualize and track the training process using TensorBoard.

This setup ensures that our environment is optimized for the high computational demands of training large language models, leveraging the full capabilities of the Leonardo supercomputer.

2.6 Memory Management

The computational power of a supercomputer enables handling models that are both computationally and memory intensive. This capability is essential for running advanced language models, but it also complicates the programming process. Several key considerations must be addressed to optimize the model's performance on such a powerful system.

Firstly, decisions must be made to ensure the model can run in parallel across multiple cores. This parallelization is crucial to leverage the supercomputer's full potential and reduce training times significantly. Techniques such as data parallelism and model parallelism are employed to distribute the workload effectively across the available cores. Data parallelism involves splitting the dataset into smaller chunks, which are then processed simultaneously by different cores. Model parallelism, on the other hand, involves splitting the model itself so that different parts of the model are processed by different cores (Dean et al. 2012).

Secondly, efficient communication between cores is paramount. The interconnects used in supercomputers like Leonardo, such as NVIDIA Mellanox HDR 200 Gb/s InfiniBand, provide high bandwidth and low latency, which are critical for minimizing the communication overhead between cores. Effective communication strategies ensure that data transfer between cores does not become a bottleneck, thereby maintaining high computational efficiency.

Lastly, maximizing the memory usage is essential to handle large models. The GPUs in Leonardo, such as the NVIDIA A100 with 64GB of HBM2e memory, allow for large portions of the model to be loaded into memory at once. This capability reduces the need for frequent memory swaps, which can significantly slow down the processing. Techniques like gradient checkpointing can also be used to save memory during training by storing only a subset of intermediate states and recomputing them as needed during the backward pass (Chen et al. 2016).

2.7 Training Process

The training program for the Mistral 7B model is designed to efficiently utilize available computational resources and optimize the training process. Here is an overview of the training script we use.

Environment Setup:

We start by setting up the working directories and loading the necessary modules and datasets. The WORK_DIR, OUTPUT_DIR, DATASET_DIR, and BASE_MODEL_DIR are defined to organize where the datasets and models are stored and where the outputs will be saved.

Model and Tokenizer Initialization:

The train function begins by loading the pre-trained Mistral 7B model using the AutoModelForCausalLM class from the Hugging Face library. This model is configured to use bfloat16 precision and optimized for low CPU memory usage. The AutoTokenizer is also loaded to handle tokenization of the text data. The original Mistral tokenizer was used.

Data Preparation:

The tokenizer's padding token is set to the end-of-sequence token to handle padding efficiently. The training and validation datasets are packed using the pack function to ensure that the context length is fully utilized with fragments from one or more documents. This approach is customary in training large language models to maximize the usage of the context window.

Training Configuration:

Our script sets several key training parameters:

- Batch Size and Gradient Accumulation: The per-device training batch size is set to 2, with gradient accumulation steps set to 8. This means that gradients are accumulated over 8 batches before performing a weight update. With 1 node and 4 GPUs, the effective mini-batch size was $2 \times 8 \times 4 = 64$.

- Training Steps: The total number of training steps is calculated based on the available GPUs, batch size, gradient accumulation steps, and context length.
- Save and Evaluation Steps: The model is configured to save checkpoints and perform evaluations at intervals determined by dividing the total training steps.

Training Arguments:

The TrainingArguments class is used to specify various training parameters such as the optimizer (adamw_bnb_8bit), learning rate, learning rate scheduler (cosine), warmup steps, evaluation strategy, and others. These settings are crucial for managing the training process, ensuring that it runs smoothly and efficiently.

Trainer Initialization and Training:

The Trainer class is instantiated with the model, training arguments, tokenizer, and datasets. The DataCollatorForLanguageModeling is used to handle data collation without masking (as it is not required for causal language modeling).

The training process is started with the trainer.train method, which optionally resumes from the last checkpoint if available.

The training ran successfully, with loss diminishing as seen in Figure 1. The total training time was approximately 10 thousand GPU hours.

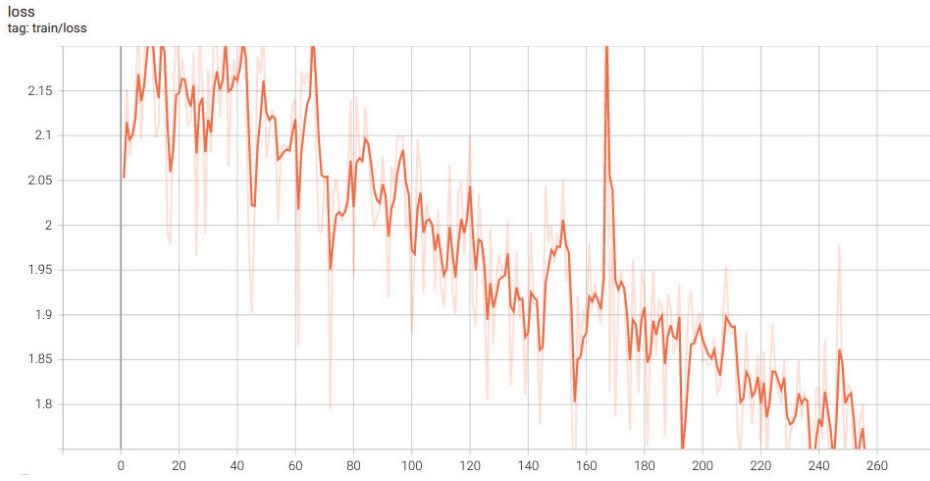

Figure 1: Loss function across the training. X-axis: epochs, Y-axis: loss value.

The unusual spike in training loss around Epoch 160 likely stems from a combination of unclipped gradients and optimizer dynamics. As gradient clipping was not used, large gradients could have caused abrupt updates to the model's weights, temporarily destabilizing training. This effect might have been amplified by the 8-bit

Adam optimizer (adamw_bnb_8bit), which – while memory-efficient – can sometimes exhibit higher variance in gradient updates. Additionally, the spike could reflect a batch of unusually noisy or challenging samples, creating a brief divergence before the model recovered. Notably, the loss quickly resumed its downward trend, suggesting the overall training process remained robust.

3. EVALUATION OF THE MODEL

To assess the performance of large language models (LLMs), various metrics have been developed or adapted from existing machine translation metrics. These metrics can broadly be classified into qualitative and quantitative categories. Quantitative metrics, such as perplexity, BLEU (Bilingual Evaluation Understudy; Papineni et al. 2002) and ROUGE (Recall-Oriented Understudy for Gisting Evaluation; Lin, Och, 2004), provide numerical evaluations of language model outputs based on their accuracy, fluency, and relevance to a reference text. On the other hand, qualitative metrics involve human evaluations that focus on aspects like coherence, creativity, and context appropriateness. While these metrics have been instrumental in evaluating LLMs, they face significant limitations when applied to languages other than English. Most metrics, particularly quantitative ones, rely on linguistic features or reference datasets that are heavily biased towards English, making it challenging to accurately gauge the performance of LLMs in non-English contexts (Bender et al. 2021; Ruder et al. 2021). Moreover, qualitative evaluations often require native speakers and culturally relevant contexts, which are not always readily available or well-defined for many languages (Blasi et al. 2022). Often, existing evaluation datasets are just machine translated from English.

We have not performed any thorough quantitative evaluation (apart from perplexity) of our model, because of lack of any relevant Slovak language metrics. Such a battery of general tests is being prepared by a wide community of researchers, and we plan to evaluate and compare our model, once the test suite is ready. We performed simple tests evaluating the percentage of out-of-lexicon words (based on comprehensive lexicographic and morphological databases), but our model already saturated the accuracy, being virtually indistinguishable from corpus samples of natural language.

Based on our (admittedly limited) experiments with the model, we conclude that it retains its original capability of in-context learning. For example, when presented with a prompt describing a bilingual dictionary and short bilingual list of Slovak-English (or English-Slovak) word pairs, the model outputs the correct equivalent of the following word, followed by a surprisingly accurate bilingual glossary. However, the results for other languages apart from English are very inaccurate. The model also lost the ability to freely switch the output language to

that of the prompt and keeps switching to Slovak, after a few words in the language of the prompt.

For comparison, we include several examples of text generation of the original Mistral-7B-v0.1 model and our mistral-sk-7b model, with the prompt marked with underline.

We used greedy search for these examples, which picks a token with the highest probability as the next token, and no repetition penalty (reflected in the first and last output of the Mistral-7B-v0.1 model).

The output from the original Mistral-7B-v0.1, we mark grammatical and lexical (but not factual errors nor texts in English) errors with text *in italics*:

> Kde bolo, tam bolo, bol raz jeden kráľ. A ten kráľ mal tri dcéry.[3] A dcéry *maly tri* synov. A *synov maly* tri dcéry. *maly tri* synov. A *synov maly* tri dcéry. *maly tri* synov. A *synov maly* tri dcéry. *maly tri* synov. A *synov maly* tri dcéry.
>
> Veľké jazykové modely sú novým trendom v umelej inteligencii. Sú zaujímavé najmä[4] *pre to*, že sú vytvorené na základe veľkých množín textov, ktoré sa vyskytujú v rôznych jazykových *korpúzach*.
>
> Na základe dostupných informácií vieme, že Ľudovít Štúr sa narodil[5] 28. 10. 1815 v obci Hlbonké nad Váhom.
> ## Ľudovít Štúr
> Ľudovít Štúr was born on 28 October 1815 in the village of Hlboké nad Váhom. He was the son of a Lutheran pastor, Ján Štúr, and his wife, Anna, née Kollár.
>
> Sociálna psychológia je pododbor psychológie, ktorý sa zaoberá[6] vzťahmi medzi *ludmi*. Sociálna psychológia sa *zabýva* vzťahmi medzi *jednotlivci*, skupinami, organizáciami a širším spoločenským kontextom.
>
> Slnko je v znamení[7] raka
> # The Sun is in the sign of Cancer
> The Sun is in the sign of Cancer.
> The Sun is in the sign of Cancer.

The output from mistral-sk-7b (remarkably, there are no grammatical errors in this output):

> Kde bolo, tam bolo, bol raz jeden kráľ. A ten kráľ mal tri dcéry. Najstaršia bola princezná, ktorá sa volala Zlatovláska. Druhá bola princezná, ktorá sa volala Ružovláska. A tretia bola princezná, ktorá sa volala Blonďatá.
> Veľké jazykové modely sú novým trendom v umelej inteligencii. Sú zaujímavé najmä preto, že dokážu spracovať veľké množstvo dát a vytvoriť z nich zmysluplné výstupy.

[3] Once upon a time, there was a king. And that king had three daughters.

[4] Large language models are a new trend in artificial intelligence. They are interesting chiefly

[5] According to available information we know that Ľudovít Štúr was born

[6] Social psychology is a subfield of psychology dealing (with)

[7] The Sun is in the sign (of)

Veľké jazykové modely sú založené na princípe transformácie prirodzeného jazyka na abstraktný jazyk, ktorý je ľahšie spracovateľný počítačom.

Na základe dostupných informácií vieme, že Ľudovít Štúr sa narodil 28. októbra 1815 v Uhrovci. V roku 1830 sa stal študentom evanjelického lýcea v Bratislave. V roku 1833 sa stal študentom evanjelického kolégia v Prešove.

Sociálna psychológia je pododbor psychológie, ktorý sa zaoberá skúmaním vzťahov medzi ľuďmi. Jej predmetom je skúmanie vzťahov medzi ľuďmi, ich správanie, vzájomné ovplyvňovanie, vzťahy medzi ľuďmi a ich prostredím.

Slnko je v znamení Vodnára a je v trigóne s Jupiterom v znamení Rýb a v kvadráte s Uránom v znamení Barana. Merkúr je v znamení Vodnára a je v trigóne s Venušou v znamení Rýb a v kvadráte s Plutom v znamení Kozorožca.

The texts generated by the original Mistral-7B-v0.1 are riddled with numerous grammatical and stylistic errors. They also frequently incorporate words from Czech, sometimes also from Croatian or Slovene[8], and often switch to English, or get stuck in repetition loops. Almost all of these issues are also visible in our examples above. This aligns with previous observations that untrained models often lack coherence and consistency in language-specific outputs (Ruder et al. 2021). In contrast, the second text, produced by the trained model, does not contain any mistakes in the provided example (which was not specifically selected). In fact, during our work with the model, we noticed grammatical mistakes are very rare and not obvious (a kind of mistakes a native speaker would make as well).

This demonstrates the positive impact of targeted training on language models' ability to generate more linguistically accurate and culturally appropriate content (Blasi et al. 2022).

4. CONCLUSIONS AND FUTURE RESEARCH DIRECTIONS

Conclusions

Large language models can be trained for various purposes, including text generation, question answering, and text error checking, among others. These models leverage vast amounts of data to understand and produce human-like text, adapting to different tasks based on their training objectives and data.

[8] After Czech, Polish and Sorbian(s) are the nearest literary languages to Slovak. However, Sorbian is used very infrequently and we would be surprised if any significant amount of Sorbian texts made it into the original training data. On the other hand, Polish is very well represented on the Internet, but the orthography is significantly distinct from Slovak, which leaves Slovene and Croatian (and its sister Latin script languages) as the languages with somewhat compatible orthography for the model to draw from. We observed this behaviour (mixing or switching to a mixture of Slovene and Croatian) with several other available LLMs with low support for Slovak.

In the first phase of our project, we decided to update an existing model for generating Slovak text. Given the nature of our training data, focusing on text generation is the most straightforward approach. Training for text generation involves feeding the model extensive datasets of Slovak text, allowing it to learn linguistic patterns, structures, and vocabulary specific to the language. This foundational step ensures that the model can produce coherent and contextually relevant Slovak text.

Text generation is a crucial capability, serving as the backbone for many applications, including automated content creation, conversational agents, and language translation (Brown et al. 2020). By starting with this fundamental task, we can establish a robust baseline for our model, making it easier to expand its capabilities in the future.

The research presented in the article highlights the significant disparity between commercial and open-source language models in handling the Slovak language. While proprietary models like GPT-4 and Claude have demonstrated strong capabilities in generating and understanding Slovak and other low-resource languages due to extensive multilingual training datasets, open-source models such as Gemma and LLaMA3 lag behind in their understanding and production of the language (Darǧis et al. 2024, Costa-Jussa et al. 2022). This gap is primarily due to the limited availability of high-quality Slovak text data and the lack of extensive fine-tuning in open models.

To address these challenges, we decided to fine-tune the existing Mistral 7B model with Slovak text data. This approach offers several advantages:

- **Open Accessibility**: By using an open-source model, we ensure that the resulting Slovak language model is publicly accessible, promoting further research and application development.
- **Resource Efficiency**: Fine-tuning an existing model is more resource-efficient compared to training a new model from scratch, leveraging pre-existing training on general language patterns.
- **Monitoring and Insights**: Training the model on Slovak data from scratch provides valuable insights into the formation of Slovak linguistic structures within the neural network, contributing to our understanding of language processing.

Our initial focus was on text generation due to the straightforward nature of the available training data. This foundational capability is essential for various applications, including automated content creation and conversational agents. The model is available under Apache 2.0 license at https://www.juls.savba.sk/llm.html.

Unresolved Questions

Several unresolved questions and challenges emerged from our research:

1. **Data Scarcity**: The limited availability of high-quality, diverse Slovak text data remains a significant barrier. There is a need for more comprehensive corpora to improve model training and performance.
2. **Fine-Tuning Strategies**: Determining the most effective strategies for fine-tuning models on low-resource languages like Slovak. This includes identifying the optimal balance between data volume and model performance.
3. **Model Evaluation**: Establishing robust evaluation metrics and benchmarks for Slovak language models to ensure consistent and accurate performance assessment.

Future Research Directions

Future research should focus on the following areas to further advance the development and application of Slovak language models:

1. **Data Collection and Curation**: Collaborating with institutions to gather and curate more extensive and diverse Slovak text corpora. This will provide a richer training dataset, enhancing model performance and generalization.
2. **Multilingual Training Techniques**: Exploring advanced multilingual training techniques that can leverage cross-linguistic data to improve performance on low-resource languages. In particular, extensive fine-tuning an existing multilingual model with additional monolingual data might diminish or destroy capabilities in other languages.
3. **Contextual Understanding**: Developing models with enhanced capabilities for contextual understanding and reasoning in Slovak. This includes training on specialized tasks such as question answering and dialogue systems.
4. **Community Engagement**: Encouraging the open-source community to contribute to the development and fine-tuning of Slovak language models. This collaborative approach can drive innovation and improve availability of models.
5. **Evaluation of Generalization Capabilities**: Future work will be to assess whether fine-tuned Slovak models (like Mistral-SK 7B) retain their inherent zero-shot and few-shot learning abilities after task-specific adaptation. Key steps include designing standardized benchmarks for Slovak to test multi-task performance (e.g., translation, sentiment analysis) without task-specific training, quantifying trade-offs between specialization (e.g., NER accuracy) and generalizability (e.g., few-shot prompt adaptability), and comparing fine-tuned models with their base versions to identify how much linguistic flexibility is lost during domain adaptation. This is important for applications requiring dynamic task switching, such as virtual assistants or educational tools. Once a training database for Slovak question answering becomes available, we plan to extend the model's functionality to include this task. Question answering requires the model to understand and generate precise

responses based on the given queries, which involves a more complex understanding of context and information retrieval.

6. **Comparative Human Evaluation of Slovak Language Models:** So far, we performed only very limited qualitative tests of the model. Although we already prepared a set of prompts in Slovak specific to Slovak cultural environment, a thorough evaluation of several Open Source models (including ours) is yet to be performed. The evaluation criteria could include fluency, grammatical correctness, cultural appropriateness, and task adherence. While resource-intensive, this comparative analysis would provide valuable insights into the relative strengths of different approaches for Slovak language processing and establish clearer benchmarks for model quality in this linguistic context.

Once a training database for Slovak question answering becomes available, we plan to extend the model's functionality to include this task. Question answering requires the model to understand and generate precise responses based on the given queries, which involves a more complex understanding of context and information retrieval (Rajpurkar et al. 2016). This extension will enhance the model's versatility, enabling it to handle a wider range of applications, such as educational tools and customer support systems.

In summary, while significant progress has been made in adapting large language models for the Slovak language, continued efforts are needed to address data limitations and improve model performance. The insights gained from this research underscore the importance of developing accessible and high-quality language models as cultural artifacts that preserve and advance linguistic heritage.

Acknowledgments

Part of the Research results was obtained using the high performance computing resources operated by CINECA and awarded within the the National Leonardo access call 2023 by the Centre of Operations, Slovak Academy of Sciences and the Slovak National Supercomputing centre.

This project was supported by the Slovak Research and Development Agency under the Contract no. APVV-22-0414.

This work was supported by the Slovak Research and Development Agency under the Contract no. APVV-22-0370.

This work was supported by DiusAI, a. s.

References

Anthropic (2023): *Claude*. Available at: https://www.anthropic.com/. [accessed 2025-07-24]

BENKO, Vladimír (2024): The Aranea Corpora Family: Ten+ Years of Processing Web-Crawled Data. In: E. Nöth – A. Horák – P. Sojka (Eds.): *Text, Speech, and Dialogue*. TSD 2024

Lecture Notes in Artificial Intelligence, Vol. 15048, Heidelberg: Springer, pp. 55–70. DOI: https://doi.org/10.1007/978-3-031-70563-2_5.

BENDER, Emily M. – GEBRU, Timnit – MCMILLAN-MAJOR, Angelina – SHMITCHELL, Shmargaret (2021): On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? In: *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, New York: Association for Computing Machinery, pp. 610–623.

BLASI, Damian – ANASTASOPOULOS, Antonios – NEUBIG, Graham (2022): Systematic Inequalities in Language Technology Performance across the World's Languages. In: S. Muresan – P. Nakov – A. Villavicencio (Eds.): *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*. Dublin: Association for Computational Linguistics, pp. 5486–5505.

BOMMASANI, Rishi – HUDSON, Drew A. – CARD, Dallas – DURMUS, Esin – SRINIVASAN, Krishnan et al. (2021): On the Opportunities and Risks of Foundation Models. In: *arXiv preprint arXiv:2108.07258*.

BROWN, Tom B. – MANN, Benjamin – RYDER, Nick – SUBBIAH, Mellanie – KAPLAN, Jared D. – DHARIWAL, Prafulla – ... – AMODEI, Dario (2020): Language models are few-shot learners. In: *Advances in Neural Information Processing Systems*, Vol. 33, pp. 1877–1901.

CHEN, Tianqi – XU, Bing – ZHANG, Chiyuan – GUESTRIN, Carlos (2016): Training Deep Nets with Sublinear Memory Cost. In: *arXiv preprint arXiv:1604.06174*.

COSTA-JUSSÀ, Marta R. – CROSS, James – ÇELEBI, Onur – ELBAYAD, Maha – HEAFIELD, Kenneth – HEFFERNAN, Kevin – ... – NLLB Team (2022): No language left behind: Scaling human-centered machine translation. In: *arXiv preprint arXiv:2207.04672*.

DARĢIS, Roberts – BĀRZDINS, Guntis – SKADIŅA, Inguna – SAULĪTE, Baiba – GRŪZĪTIS, Normunds (2024): Evaluating open-source LLMs in low-resource languages: Insights from Latvian high school exams. In: M. Hämäläinen – E. Öhman – S. Miyagawa – K. Alnajjar – Y. Bizzoni (Eds.): *Proceedings of the 4th International Conference on Natural Language Processing for Digital Humanities*, Miami: Association for Computational Linguistics, pp. 289–293.

DEAN, Jeffrey – CORRADO, Greg – MONGA, Rajat – CHEN, Kai – DEVIN, Matthieu – MAO, Mark, et al. (2012): Large Scale Distributed Deep Networks. In: *Advances in Neural Information Processing Systems*, Vol. 25.

DEVLIN, Jacob – CHANG, Ming-Wei – LEE, Kenton – TOUTANOVA, Kristina (2019): Bert: Pre-training of deep bidirectional transformers for language understanding. In: *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies*, Vol. 1 (long and short papers), pp. 4171–4186.

GARABÍK, Radovan (2025): Webový korpus slovenčiny ARANEUM + HPLT + FineWeb2. In: *Kultúra slova*, Vol. 59, No. 5, pp. 292–296.

Google (2024): *Gemini*. Available at: https://gemini.google.com/. [accessed 2025-07-24]

Google (2025): *Gemma 3*. Available at: https://huggingface.co/google/gemma-3-4b-it/. [accessed 2025-07-24]

HPC Cineca (2023): *Leonardo HPC System*. Available at: https://leonardo-supercomputer.cineca.eu/. [accessed 2025-07-24]

HOWARD, Jeremy – RUDER, Sebastian (2018): Universal Language Model Fine-tuning for Text Classification. In: *arXiv preprint arXiv:1801.06146*.

LIN, Chin-Yew – OCH, Franz Josef (2004): Looking for a few good metrics: ROUGE and its evaluation. In: *Ntcir workshop*, pp. 1–8.

Meta (2025): *Llama 3*. Available at: https://www.llama.com/docs/model-cards-and-prompt-formats/llama3_3/. [accessed 2025-07-24]
Mistral AI (2023): *Mistral 7B v0.1*. Available at: https://huggingface.co/mistralai/Mistral-7B-v0.1/. [accessed 2025-07-24]
OpenAI. (2023): *GPT-4*. Available at: https://platform.openai.com/docs/models/overview. [accessed 2025-07-24]
PAPINENI, Kishore – ROUKOS, Salim – WARD, Todd – ZHU, Wei-Jing (2002): Bleu: a method for automatic evaluation of machine translation. In: P. Isabelle – E. Charniak – D. Lin (Eds.): *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*. Philadelphia, Pennsylvania: Association for Computational Linguistics, pp. 311–318.
RADFORD, Alec – WU, Jeffrey – CHILD, Rewon – LUAN, David – AMODEI, Dario – SUTSKEVER, Ilya (2019): Language Models are Unsupervised Multitask Learners. In: *OpenAI Blog*. Available at: https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf. [accessed 2025-07-24]
RAJPURKAR, Pranav – ZHANG, Jian – LOPYREV, Konstantin – LIANG, Percy (2016): SquAD: 100,000+ Questions for Machine Comprehension of Text. In: *arXiv preprint arXiv:1606.05250*.
RUDER, Sebastian – PETERS, Matthew E. – SWAYAMDIPTA, Swabha – WOLF, Thomas (2021): Transfer Learning in Natural Language Processing. In: K. Toutanova – A. Rumshisky – L. Zettlemoyer et al. (Eds.): *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, pp. 765–780.
TOUVRON, Hugo – LAVRIL, Thibaut – IZACARD, Gautier – MARTINET, Xavier – LACHAUX, Marie-Anne – LACROIX, Timothée et al. (2023): LLaMA: Open and Efficient Foundation Language Models. In: *arXiv preprint arXiv:2302.13971*.
VASWANI, Ashish – SHAZEER, Noam – PARMAR, Niki – USZKOREIT, Jakob – JONES, Llion – GOMEZ, Aidan N. – KAISER, Lukasz – POLOSUKHIN, Illia (2017): Attention is All you Need. In: *Advances in Neural Information Processing Systems*, Vol. 30, pp. 5998–6008.
Webový korpus slovenčiny ARANEUM + HPLT + FineWeb (2025). Available at: https://www.juls.savba.sk/webskahfcorp.html. [accessed 2025-07-24]

APPENDIX

Slovak support in selected LLMs. If there are multiple versions, we include the version that first supported Slovak, as well as the immediately preceding version (if applicable), and the version with the smallest number of parameters that supports Slovak, as well as the version with strictly smaller number of parameters. The judgement was necessarily subjective, although the level of support was clearly delineated, with no borderline cases.

The table reflects the situation at the time of submitting the final version of the article, the situation at the start of our work was different and much less favourable for Slovak – of the models mentioned in the table, only Llama-2, Phi-1.5, gemma-7b and Mistral-7B-v0.1were available.

model	*provider*	*supports Slovak*
Llama-2-70b-hf	Meta	no (cs)
Llama-3.1-8B	Meta	yes
Mistral-7B-v0.1	Mistral AI	(very) badly
gemma-7b	Google	badly
gemma-2-2b-it	Google	badly
gemma-2-2b	Google	yes
gemma-3-1b-pt	Google	yes
Phi-1.5	Microsoft	no
Phi-3-Small-8K-Instruct	Microsoft	badly
Qwen3-4B	Alibaba Cloud	badly
Qwen3-8B	Alibaba Cloud	yes

Table 2: Slovak support in various smaller Open Source LLMs.

Legend:

- yes – Slovak is fully supported, lexical or grammatical errors only sporadic
- no – Slovak is not supported, model either outputs text in other languages or the output is garbled
- badly – the output is intelligible, clearly in Slovak, but with many grammatical and orthographic mistakes
- (cs) – the output is in Czech, despite the input prompt being in Slovak