A Small Language AI Model in the Bosnian Language

Boško Jefić[1], Vlatko Bodul[2], Admir Agić[3]
[1] *the City Administration of Zenica, Zenica, BH, bosko.jefic@zenica.ba*
[2] *B.Sc.it, Zenica, BH, vlatko.bodul@unze.ba*
[3]*Kolektiv EU, Tešanj, BH, admir.agic@kolektiv.pro*

Original scientific paper
https://doi.org/10.7251/JIT2502128J UDC: 378.147.311.4:821.163.4

Abstract: This study presents the development and evaluation of Mali Mujo, a small-scale language model optimized for the Bosnian language, designed to operate efficiently on devices with limited computational resources. Leveraging the TinyLlama architecture, the model demonstrates the feasibility of deploying natural language processing (NLP) applications in environments with constrained memory and processing capabilities, specifically devices with 1 GB storage and 8 GB RAM. The system integrates Langchain agents and the DuckDuckGo API to enable real-time information retrieval, enhancing the model's responsiveness and accuracy in practical applications. The methodology involved training the TinyLlama model on a curated Bosnian dataset, followed by testing across diverse real-world scenarios in industry and administration. Performance metrics focused on accuracy, response time, and computational efficiency, while additional evaluation considered user experience and adaptability to domain-specific tasks. The results indicate that Mali Mujo delivers rapid and reliable responses to user queries, with significant advantages in speed and resource efficiency compared to larger language models. The model effectively processes administrative requests, generates technical and market-related insights, and supports educational and governmental applications, highlighting its versatility. While small-scale models exhibit lower absolute accuracy than their larger counterparts, the study demonstrates that careful optimization and integration with external APIs can mitigate limitations, providing a balance between performance and accessibility. Furthermore, the model's design ensures user privacy and low energy consumption, contributing to sustainable and secure AI deployment. Mali Mujo exemplifies the potential of small language models to enhance efficiency, accessibility, and usability in local-language contexts. Its deployment provides a scalable, cost-effective solution for organizations with limited infrastructure, offering opportunities for further enhancement through expanded datasets, multilingual support, adaptive learning, and integration with emerging AI technologies. The findings underscore the practicality of small AI models in bridging the gap between advanced NLP capabilities and resource-constrained environments.

Keywords: Small language models, TinyLlama, Bosnian language, Langchain agents, Real-time information retrieval, AI in industry

Introduction

The development of an AI model in the Bosnian language holds crucial importance for enhancing communication and efficiency within both industry and public administration. Although the Bosnian language is rich and diverse, it faces significant challenges in automation and data processing due to its specific grammatical and syntactic structures. Current AI models are often not optimized for operation on devices with limited computational resources, such as those with smaller storage capacities or lower amounts of RAM.

Implementing an AI model in the Bosnian language can facilitate administrative processes, enable faster documentation processing, and improve communication between citizens and institutions. By employing AI systems, particularly small language models, it is possible to automate many routine tasks, thereby saving considerable time and resources. Furthermore, such a model could enhance data organization, information retrieval, and the analysis of administrative documentation.

Developing a language model specifically for Bosnian ensures more accurate recognition of idiomatic expressions, terminology, and contextual nuances unique to the language.

In addition, this development plays a vital role in promoting digital inclusion by enabling a larger number of users to access modern technological solutions in their native language. Given that many individuals still rely on administrative services in their mother tongue, AI models can contribute to improved compliance with legal and regulatory frameworks.

Although AI models for other languages exist, the particularities of Bosnian require tailored approaches to ensure both accuracy and contextual relevance. Utilizing such a model allows for faster data recognition and processing, as well as greater reliability in decision-making processes. Moreover, AI models can help prevent errors in administrative workflows, which often depend on precise linguistic interpretation.

This opens up opportunities for improving user experience and increasing citizen satisfaction with public services. The development of these models also stimulates the digital transformation of sectors that continue to depend heavily on manual data entry. Ultimately, the deployment of AI models in the Bosnian language can strengthen Bosnia and Herzegovina's competitiveness on the global technological landscape [1].

Research Objective

The objective of this research is to develop a **small-scale AI language model** specifically adapted to the Bosnian language, capable of efficient operation on devices with limited hardware resources. In today's digital age, many users in Bosnia and Herzegovina, as well as across the broader region, rely on devices with modest technical specifications. This model is intended to be optimized for systems with as little as **1 GB of storage and 8 GB of RAM**, which presents challenges in terms of both speed and efficiency as defined in Table 1.

By developing such a model, the goal is to enable users who lack access to high-end technology to still benefit from advanced linguistic tools in their native language. In addition to being lightweight, the model must effectively handle the linguistic complexity of Bosnian, including its grammatical and syntactic features[2].

Table 1. *Key technical specifications and objectives of the Bosnian-language AI model*

Specification	Description
Target devices	Devices with 1 GB storage and 8 GB RAM
Model type	Small language AI model
Optimization	Designed for operation on resource-limited devices
Focus	Recognition and processing of the Bosnian language
Applications	Administrative and industrial processes
Linguistic complexity	Processing grammatical and syntactic characteristics

The research will focus on assessing the model's efficiency in recognizing and processing Bosnian across different contexts, such as administrative and industrial environments. Another goal is to develop an AI model that is not only lightweight but also precise in linguistic interpretation, considering the complexity of Bosnian grammar. Such a model would enable automation of numerous routine processes, thereby reducing the need for human intervention [3].

The aim is to design a model sufficiently flexible to be applied across various sectors, from public institutions to small enterprises. Optimization efforts will not be limited to memory consumption but will also focus on **data processing speed and system response time reduction**. By developing this model, the research aims to expand the practical use of AI technologies in Bosnia and Herzegovina, even in areas that have so far been underserved.

Finally, the project seeks to ensure the model's accessibility to a wide range of users, regardless of the technical capabilities of their devices. The research will also involve the development of **specialized algorithms** designed to minimize model size while preserving its functionality. Ultimately, this research aims to deliver a solution that will improve the everyday use of technology in the Bosnian language, particularly for users operating with limited computational resources [3].

Advantages and Challenges in Developing Models for Specific Languages

The Bosnian language, like other languages of the Balkan region, presents a set of unique characteristics that pose challenges in the development of language models. One of the primary linguistic features of Bosnian is its rich inflectional morphology, meaning that word forms change depending on grammatical context. These variations include noun declension, verb conjugation, and complex morphological forms that account for gender, number, and case.

Bosnian also exhibits a significant degree of dialectal diversity, with differences in pronunciation, vocabulary, and grammar, which further complicates the development of a model capable of understanding all linguistic variants. For instance, dialectal differences may lead to difficulties in recognizing and processing certain terms and expressions.

Moreover, the Bosnian language employs unique word forms that are difficult to translate into other languages, including numerous archaic and regional expressions. Such expressions are often underrepresented in global text corpora, making them difficult for AI models to detect or process. Another challenge arises from the existence of multiple orthographic variants, which can affect the accuracy and consistency of model predictions. All those challenges are defined in Table 2.

Table 2. *Core linguistic characteristics of the Bosnian language*

Characteristic	Description
Inflection	Changes in word forms depending on grammatical context.
Noun declension	Nouns vary according to gender (masculine, feminine, neuter), number (singular, plural), and case (nominative, genitive, dative, etc.).
Verb conjugation	Verbs conjugate according to person (first, second, third) and tense (present, perfect, future, etc.).
Syntactic complexity	The Bosnian language employs complex sentence structures that can include multiple dependent and independent clauses.
Synonymy	A rich variety of synonyms, which can make contextual understanding more difficult.
Dialects	Various dialects differing in pronunciation and vocabulary.
Orthographic variants	The existence of different spelling conventions that can affect written communication.
Archaic and regional expressions	Many expressions are not represented in global data corpora, making their recognition challenging for AI models.

Given the abundance of synonyms, a language model must be capable of understanding the context in which a word is used to accurately interpret its meaning. Developing a model that effectively recognizes and comprehends Bosnian expressions requires a deep understanding of the cultural, social, and historical specificities of the language [3].

A further challenge lies in the lack of sufficient annotated datasets for training, as many linguistic resources are not available in Bosnian. The use of general-purpose language models that are not tailored to the Bosnian language can result in errors in text recognition and generation. Another issue is the high variability and inconsistency in spelling and spoken usage, which can hinder precise understanding and response generation.

Developing a model capable of capturing all nuances of the Bosnian language requires the use of advanced learning techniques, such as deep learning and transfer learning, to improve model accuracy [4].

In addition, adapting a model for specific tasks, such as administrative processing or data retrieval, necessitates fine-tuning on domain-specific datasets. While existing models often perform well for major languages such as English, languages like Bosnian demand the creation of specialized tools that capture all their linguistic features.

Ultimately, although the development of language models for underrepresented languages presents considerable challenges, it offers significant advantages. It enables access to advanced technologies for speakers of smaller languages, thereby promoting linguistic diversity, digital inclusion, and equitable technological participation across different linguistic communities.

Overview of Existing Solutions

An overview of existing AI language model solutions for languages like Bosnian, such as Croatian and Serbian, reveals several noteworthy approaches. Although these languages share a high degree of mutual intelligibility, each possesses unique grammatical, lexical, and orthographic characteristics.

For the Serbian language, several variants of the BERT model have been successfully implemented across a range of applications, including search automation and sentiment analysis. Similarly, for Croatian, models have been developed that recognize and generate text according to the specific features of Croatian orthography and syntax.

In industrial contexts, these models are used to manage large volumes of data more efficiently, for instance, through automated document classification, report generation, and customer support systems. In public administration, Serbian and Croatian language models have already been deployed as part of digital transformation initiatives, enabling the automation of document and request processing [1].

For example, AI tools based on these models assist in the recognition and archiving of legal documents, as well as in the analysis of public policies. However, the application of such solutions in Bosnia and Herzegovina, where Bosnian is the official language, encounters obstacles due to the lack of sufficiently specific linguistic resources [4].

While models developed for Serbian and Croatian can handle basic language processing tasks; they are not always capable of managing all the variants and dialects of Bosnian. This limitation necessitates further adaptation and fine-tuning.

There have been efforts to develop AI models specifically for the Bosnian language, yet these projects continue to face difficulties in recognizing local expressions, archaic terms, and culturally embedded linguistic features. In industrial applications, the use of AI models for Bosnian remains in its early stages, while administrative implementations are mostly limited to standardized language processing tasks. Another challenge is the integration of existing solutions with systems used in Bosnia and Herzegovina, which often have unique technical requirements and infrastructural constraints.

An analysis of current solutions, defined in Figure 1., shows that AI tools in Bosnia and Herzegovina are primarily trained on the standard form of the language, while dialects and regional variations tend to be neglected. In this context, future research and development of compact language models specifically tailored to the Bosnian language, such as TinyLlama, could provide solutions that are better aligned with local linguistic characteristics, computational limitations, and institutional needs.

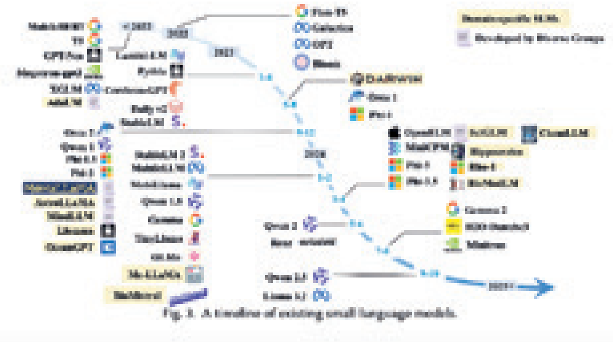


Figure 1. *Overview of existing SLM solutions*

Methods and materials

The development of Mali Mujo, a small language model for the Bosnian language, is based on the TinyLlama architecture, optimized for low-resource environments. The model was trained on a curated corpus consisting of publicly available Bosnian texts, domain-specific documents relevant to administrative and industrial tasks, and anonymized user-generated queries. All data were preprocessed through tokenization, normalization, and filtering to remove duplicates and irrelevant content, and divided into training and validation sets for proper evaluation.

TinyLlama is a compact, open-source 1.1B parameter language model pretrained on approximately 1 trillion tokens for around 3 epochs. TinyLlama adopts the same architecture and tokenizer as LLaMA 2, enabling seamless integration into existing open-source projects built on LLaMA. The model incorporates efficiency improvements from the open-source community, such as FlashAttention and Lit-GPT, achieving high computational efficiency while maintaining a small memory footprint.

Despite its modest size, TinyLlama demonstrates strong performance on a variety of downstream tasks, outperforming comparable open-source models. Model checkpoints and code are publicly available on GitHub developer portal (https://github.com/jzhang38/TinyLlama), making TinyLlama a practical and efficient choice for academic research and AI training on CPU- or GPU-constrained environments.

To enhance task execution and information retrieval, Mali Mujo utilizes Langchain agents, which manage query interpretation, communication with external APIs, and multi-step reasoning. The system is integrated with the DuckDuckGo API to access real-time web information while preserving user privacy.

API responses are parsed and contextualized by the agents to generate coherent and accurate answers, supplementing the model's pre-trained knowledge.

The application is designed to operate efficiently on devices with limited computational resources, requiring only 1 GB of storage and 8 GB of RAM. The software stack includes Python 3, PyTorch for model training and inference, Gradio for the user interface, and Ollama server for deployment.

Optimization strategies included model pruning, quantization, batch processing, caching, and adaptive resource management to maintain responsiveness and reduce computational load. Together, these methods ensure that Mali Mujo provides fast, accurate, and relevant responses while remaining accessible and functional on low-end hardware.

Development of the Mali Mujo Application

The development of the *Mali Mujo* application requires an efficient technical infrastructure designed to optimize performance on devices with limited computational resources. To manage large volumes of data and perform linguistic tasks, the application employs an **Ollama server**, which enables fast and efficient handling of AI models. The **DuckDuckGo Search API** is integrated to facilitate internet search capabilities within the application, allowing users to access up-to-date information without overloading local resources [5].

LangChain agents and tools are implemented to automate language-related tasks such as text recognition and generation, ensuring natural interaction between the user and the system. The internal memory architecture allows for the storage of temporary data and optimizes application performance by reducing the need for constant access to external data sources. The integration of the **TinyLlama model for the Bosnian language** enables the application to recognize linguistic specificities, including grammatical and syntactic structures, while maintaining high efficiency on low-resource systems [6].

TinyLlama is optimized for environments with limited computational capacity, making it an ideal choice for an application such as *Mali Mujo*. The model is trained specifically on Bosnian linguistic features, including **regional variations and local expressions**, which allows for precise understanding and text generation. Through this integration, *Mali Mujo* can perform a range of tasks, such as **automated question answering, text recognition, and data analysis**, all while minimizing system load.

The application is designed to make optimal use of limited resources, thereby providing access to advanced AI functionalities for users with devices of modest specifications. The integration of the TinyLlama model also allows the application to adapt to the specific needs of **industrial and administrative users** [6].

For instance, *Mali Mujo* can automatically process requests and generate documents in Bosnian, significantly accelerating administrative workflows. The system utilizes **LangChain agents** to interpret user queries and search instructions, while the **DuckDuckGo API** provides an extensive data source for more accurate responses. The entire system is architected to minimize reliance on external resources, making the application faster and more efficient.

The development of *Mali Mujo* in combination with the TinyLlama model represents a significant advancement for the **adoption of AI technologies in countries with limited technical resources**, such as Bosnia and Herzegovina.

Model Training

Training the **TinyLlama model** for the Bosnian language requires a specialized approach to ensure adaptation to the language's unique characteristics as defined in Figure 2. This dataset is a collection of news articles in the Bosnian language sourced from klix.ba, a prominent Bosnian online news portal. The dataset covers a wide range of topics including local and international news, politics, economics, sports, entertainment, and more.

The dataset comprises a single file, **klix_df.csv**, containing a total of 786,755 articles sourced from the Bosnian news portal **klix.ba**, covering a broad spectrum of thematic categories including news, politics, economics, sports, entertainment, and other related domains. Each record in the dataset includes the article's **title**, a **hyperlink** directing to the original publication, its assigned **article_class** and corresponding **article_class_name**, as well as quantitative engagement indicators such as the **number of comments** and **number of shares**.

Raw dataset we used is avalable on this link: Seferovic8/Bosnian-News-Articles-Dataset-from-klix.

ba: This dataset is a collection of news articles in the Bosnian language sourced from klix.ba, a prominent Bosnian online news portal. The dataset covers a wide range of topics including local and international news, politics, economics, sports, entertainment, and more.

Additionally, the dataset provides the **file path** to the article's associated image and the full **textual content** of the article. All materials are authored in the Bosnian language, offering a comprehensive and diverse corpus suitable for linguistic, journalistic, or computational analysis.

The first step in the training process involves **data preparation**, with the dataset consisting of **18 JSONL files** containing Bosnian-language texts for better training performances regarding use or resources.

The data is then **cleaned and structured** to remove irrelevant information and formatted for model training, considering the specific grammatical and syntactic properties of the Bosnian language. Each file in the JSONL dataset represents a set of **input–output pairs**, where input data serves as model training material and output data represents the desired response or result.

For traning we use PC server with Ubuntu 22.04.4 LTS, is running on a virtual machine (KVM) with a **12-core AMD EPYC processor** (12 virtual CPUs, each with 1 thread) and a 64-bit x86_64 architecture. The system provides **48 GB of RAM**, ensuring ample memory for computation-heavy tasks. The CPU features modern instruction sets including AVX, AVX2, AES, FMA, BMI1/2, and SHA extensions, making it suitable for AI workloads and multi-threaded applications. In addition to CPU we used also Nvidia GPU with 8 GB of RAM. Disc space used was appx. 10 GB.

The model is trained over several cycles during which it learns to recognize specific linguistic patterns and dialectal variations of the Bosnian language. Once the training process is completed, a testing phase follows, aimed at evaluating the model's accuracy and efficiency in both text recognition and generation. After testing, the trained model is integrated into the Mali Mujo application, which utilizes the Ollama server for executing all language-related tasks [8].

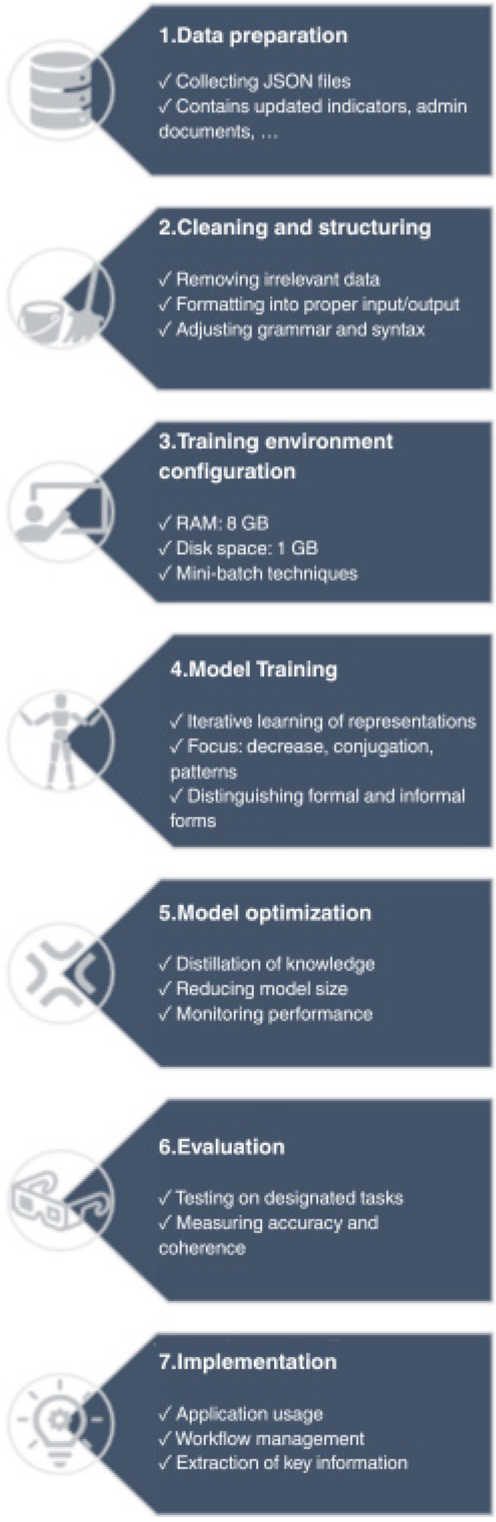

1.Data preparation
✓ Collecting JSON files
✓ Contains updated indicators, admin documents, ...
2.Cleaning and structuring
✓ Removing irrelevant data
✓ Formatting into proper input/output
✓ Adjusting grammar and syntax
3.Training environment configuration
✓ RAM: 8 GB
✓ Disk space: 1 GB
✓ Mini-batch techniques
4.Model Training
✓ Iterative learning of representations
✓ Focus: decrease, conjugation, patterns
✓ Distinguishing formal and informal forms
5.Model optimization
✓ Distillation of knowledge
✓ Reducing model size
✓ Monitoring performance
6.Evaluation
✓ Testing on designated tasks
✓ Measuring accuracy and coherence
7.Implementation
✓ Application usage
✓ Workflow management
✓ Extraction of key information

Figure 2. *Training process of the TinyLlama model in the Bosnian language*

During training, the model learns to recognize **linguistic patterns** such as declension, conjugation, and syntactic structures. The training procedure employs a **mini-batch approach** to minimize memory requirements and computational load [9].

Through this iterative process, the model gradually improves its ability to **understand and generate**

Bosnian language structures. Through the application of optimization techniques such as **knowledge distillation**, the model's size is reduced while retaining most of the functionality of larger models. As a result, the TinyLlama model is trained to recognize **regional variants and local expressions** specific to the Bosnian language [8]

The fine-tuning training of language model was executed using the python script on a CPU-only environment utilizing 12 available cores. The process employed a tokenized dataset comprising 4,581,000 examples. Training was configured for three epochs with an effective batch size of 8 (a per-device batch size of 4 combined with a gradient accumulation step count of 2) and an initial learning rate of 3 x 10^{-4}.

The training objective utilized Causal Language Modeling (mlm=False), calculated over a total of 1,717,875 steps. Initial training stability was monitored with the first logged batch loss being 3.0888, corresponding to a gradient norm of 1.264. Reflecting the resource-constrained environment, the training speed exhibited low throughput, averaging approximately 40.23 seconds per iteration (s/it) after 6,450 steps. Training data is showed in Figure 3. and Figure 4.

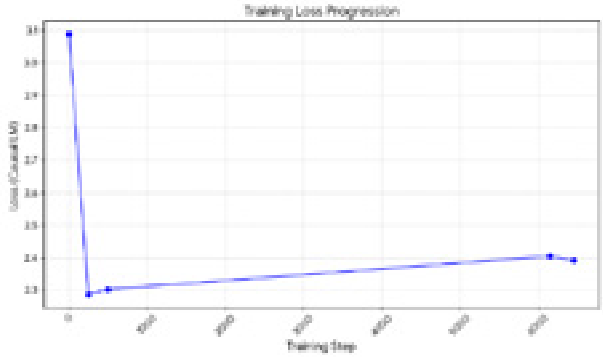


Figure 3. *Training Loss Progression (Loss vs. Step)*

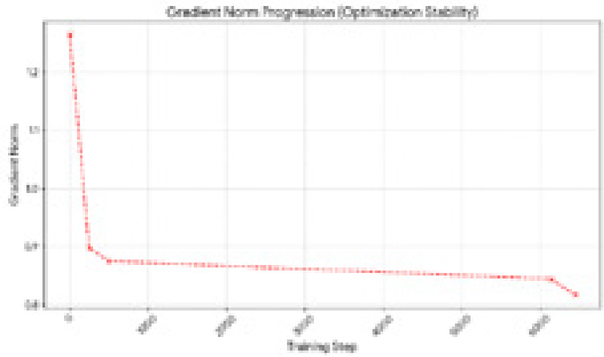


Figure 4. *Gradient Norm Progression (Grad Norm vs. Step)*

Training Loss Progression (Loss vs. Step) diagram illustrates a sharp initial decrease in the Causal Language Model (CLM) Loss from 3.0888 at Step 1 to 2.2864 at Step 250, indicating that the model quickly adapted to the dataset. The loss then stabilizes, which is typical behavior after the initial rapid learning phase as indicate in Figure 3.

Gradient Norm Progression (Grad Norm vs. Step) diagram shows the Gradient Norm decreasing consistently from an initial value of 1.264 down to 0.817 by Step 6450. A smooth, consistent decrease in the gradient norm is a positive indicator of optimization stability and suggests that the model's parameters are being updated effectively without encountering numerical instability or vanishing/exploding gradients as indicate in Figure 4.

Optimization stability was immediately established, while the gradient norm concurrently decreased, confirming efficient and stable convergence during the early stages of training. System restriction resulted in low processing speeds, with individual iteration times initially fluctuating around 40.23 s/it after 6,450 steps , and later exhibiting extreme variability, occasionally spiking to over 116 s/it. Training finish in appx. 20 days.

Language Training and Optimization

The training is designed not only to enable the model to understand text but also to generate coherent and grammatically correct responses. Upon completion of the training phase, a series of evaluation tests is conducted to assess the model's accuracy in recognizing and generating Bosnian text. This process allows the TinyLlama model to evolve into a precise and efficient tool for working with the Bosnian language. After training, the model is deployed for various tasks, including automatic response generation and key information extraction from textual data [10]

Resource Optimization

Resource optimization is a critical aspect of enabling efficient model performance on systems with limited computational capacity. One of the primary techniques used in this process is data compression, which reduces the size of the data required for training and model operation. Compression decreases the number of model parameters, thereby enhancing speed and reducing memory demand.

Another key technique is model distillation, a process in which a smaller model is trained to emulate the behaviour of a larger one. This approach enables the smaller model to retain most of the functionality of its larger counterpart while operating with substantially fewer resources. Through distillation, the model's size is reduced without significant loss of accuracy, ensuring it remains suitable for linguistic tasks.

Model simplification is also achieved by using smaller neural architectures with fewer layers, directly reducing memory consumption and processing time. In addition, the application of pruning techniques, the removal of redundant parameters and connections within the network, further reduces model size and enhances computational efficiency.

For memory optimization, a strategy of dynamic data loading is employed, where only relevant data segments are loaded into memory as needed. Additionally, selectively lowering model precision during certain stages of training can accelerate data processing without causing a notable decrease in performance.

Another optimization approach involves quantization, a technique that reduces numerical precision within the model, thereby decreasing the number of bits required to represent data. This substantially reduces memory usage while maintaining acceptable accuracy levels.

These optimization techniques collectively enable the model to operate on low-resource devices, such as smartphones and computers with limited RAM. Furthermore, parallelization of training processes and the use of batch processing allow for more efficient resource management during model training. All these strategies ensure that the TinyLlama model functions effectively, minimizing processing time while maintaining usability in applications running in constrained computing environments [1]

The incorporation of advanced methods such as knowledge distillation and early stopping further reduces training duration, a key factor in resource-efficient optimization. Ultimately, the implementation of these techniques makes the model competitive and practical for real-world applications that require efficient use of computational resources [9]

System Development and Implementation

The development and implementation of the Mali Mujo application system begins with an analysis of user requirements and specific operational needs. The first step involves the design of the system architecture, which includes the selection of technologies and the definition of core components such as the Ollama server, DuckDuckGo API, and LangChain agents [11].

The integration of the DuckDuckGo API enables access to online information sources, while LangChain agents facilitate natural and dynamic interaction between the user and the system. The application is then optimized for devices with limited computational capacity through techniques such as data compression and model size reduction. Subsequently, the application is installed on devices with only 1 GB of storage and 8 GB of RAM, where its speed and efficiency are tested under constrained conditions [6].

During this phase, it is crucial to ensure that the application consumes minimal memory and CPU resources, allowing smooth operation even on low-specification devices. A key design principle is usability, ensuring that the user interface remains simple and intuitive. The interface is tested through simulated user scenarios to verify system functionality and stability. The application also adheres to data protection and cybersecurity standards, implementing measures for safeguarding user information.

After successful implementation, the application is deployed in a production environment, where its performance is monitored and optimized. Continuous testing and iterative improvements allow Mali Mujo to evolve into an effective tool for administrative and industrial applications.

In the final phase, the application is regularly updated to remain aligned with ongoing developments in AI and language technologies. The Ollama server employs advanced optimization techniques to minimize memory and CPU usage, improving overall performance. APIs connected to the Ollama server enable the application to perform various real-time linguistic tasks, such as automatic question answering and key information extraction.

Thanks to server-level optimizations, response latency is minimized, allowing Mali Mujo to handle multiple simultaneous user requests without significant slowdowns. The use of APIs provides a simple and scalable mechanism for interaction with the model, supporting future growth and integration with other systems.

Utilization of the Ollama Server and APIs

The Ollama server plays a crucial role in ensuring the speed and efficiency of the Mali Mujo application. Acting as the central processing unit for executing AI models, it enables optimized resource management, an essential feature for operation on devices with limited computational capacity. Through the Ollama server, the application efficiently performs language-related tasks such as text recognition and generation without overloading local hardware [12].

The server allows complex AI models, including TinyLlama, to function effectively on devices with only 8 GB of RAM and 1 GB of storage. By leveraging APIs, the system connects with external resources such as DuckDuckGo for web searches and LangChain agents for advanced linguistic operations.

The Ollama server also provides scalability, allowing the system to accommodate a growing number of users and requests without requiring significant hardware upgrades. Through its API interfaces, the server facilitates seamless integration with other applications and information systems, increasing flexibility and adaptability in deployment.

By offloading complex algorithmic processing from the client device, the server significantly reduces local resource consumption, thereby enhancing speed and overall system performance. Furthermore, the server enables centralized monitoring and optimization of the model's performance, simplifying troubleshooting and continuous improvement.

From a security perspective, the Ollama server ensures encrypted communication between the client application and the server, safeguarding sensitive user data. Additionally, centralized updates of models and APIs enable rapid deployment of new functionalities without requiring complex modifications on end-user devices.

Through this architecture, the Mali Mujo application can efficiently process large volumes of data, a capability that is essential for its intended use in administrative and industrial environments. With the support of the Ollama server, the system maintains high performance and operational stability even under heavy workloads, ensuring long-term scalability and reliability.

Internet Search with the DuckDuckGo API

The DuckDuckGo API enables the Mali Mujo application to search the internet in real time and retrieve relevant information without compromising user privacy. By using this API, the application can directly send queries to the DuckDuckGo search engine, which then returns results based on the user's request. Unlike other search engines, DuckDuckGo emphasizes privacy protection by not tracking user activity, which is an important feature for data security.

The search process is both fast and efficient thanks to the API's simple interface, which allows seamless integration with the application. When a user submits a question or request, the application sends a query to the DuckDuckGo API, which returns search results in an easily processable format. These results may include textual answers, links, or other useful data that the application can use to generate responses or provide additional information to the user.

The DuckDuckGo API supports searches for specific types of content, including web pages, images, and news, thereby making the application more flexible in its ability to provide relevant answers. The integration of this API enhances the overall user experience by allowing the application to rely on external information sources, thus expanding its knowledge base and capabilities.

Furthermore, the DuckDuckGo API supports searches in multiple languages, including Bosnian, which increases the relevance of the responses for users in Bosnia and Herzegovina. The API is highly responsive, enabling the application to quickly collect and deliver information in near real time.

By using the DuckDuckGo API, the application reduces the need for local data processing, as the API itself handles the search and retrieval of results. This also allows for dynamic data updates, which is especially useful for monitoring current and time-sensitive information. In this way, the application always has access to the most recent online content, which is essential for answering questions related to ongoing events.

The DuckDuckGo API is a reliable search tool that provides high-quality, relevant, and accurate results.

It also contributes to the overall security of the Mali Mujo application, as it enables searches without storing user data or browsing history. Integrating the DuckDuckGo API thus allows the Mali Mujo application to provide fast, secure, and privacy-conscious access to the internet, making it highly suitable for a wide range of industrial and administrative applications.

LangChain Agents and Tools

LangChain is a platform that facilitates the development and implementation of advanced AI agents capable of performing various tasks. By using LangChain, the Mali Mujo application can create agents specialized in specific operations such as data retrieval, response generation, and complex query execution as defined in Table 3. LangChain allows these agents to communicate with different APIs, tools, and external resources, thereby increasing the flexibility of the system.

Through these tools, the agents can efficiently interpret user requests and make relevant decisions in real time. LangChain agents can analyse textual data, recognize linguistic patterns, and perform actions based on the information obtained. For example, an agent can analyse a user's question, process it, and then send a query to the DuckDuckGo API to find additional information online.

Moreover, LangChain enables agents to integrate diverse functionalities, including search, model training, text generation, and other language-related tasks. These agents can operate autonomously, executing tasks without constant human supervision. LangChain also supports the efficient management of complex workflows, allowing agents to perform multiple operations sequentially or in parallel.

Table 3. *– Key Functionalities of LangChain Agents*

Functionality	Description
Pattern recognition	Analysis and understanding of linguistic structures within a query.
API utilization	Access to external resources such as search engines or databases.
Autonomous decision-making	Making decisions without human intervention.
Workflow management	Coordination of complex sequences of tasks.
Agent training	Learning from data to improve efficiency.

Each LangChain agent can be specialized for a specific type of task, improving the accuracy and speed of execution. The tools provided by LangChain also simplify integration with external systems such as the Ollama server, databases, or other AI models.

By employing LangChain agents, the Mali Mujo application can automatically process user queries, generate responses, and provide relevant information in real time. LangChain also supports agent training, allowing them to continually improve performance and adapt to new challenges. Agents can learn from the data they process, becoming more precise and efficient over time.

LangChain facilitates the creation of scalable and adaptable systems that can evolve alongside user needs. Ultimately, the use of LangChain agents enables the Mali Mujo application to perform complex tasks that would otherwise require multiple systems or human intervention, making it a powerful and intelligent component of the overall architecture.

Model Evaluation

The evaluation of the model is a crucial step in the development process of the *Mali Mujo* application, as it allows an assessment of its performance under real-world conditions. The first evaluation criterion was accuracy, which refers to the model's ability to generate correct responses based on user queries. Model accuracy was tested across multiple datasets, including specialized Bosnian language queries.

The second important criterion was efficiency, specifically the speed with which the model can process data and generate responses. Efficiency was measured in terms of task execution time, particularly on devices with limited resources. Given these constraints, testing also included assessing memory and CPU usage during task execution.

Model performance was further analysed based on its ability to handle multiple queries in a short period without significantly slowing down the system. The use of specialized tools, such as LangChain agents and the DuckDuckGo API, allowed additional optimization in terms of both speed and response accuracy. The model was evaluated in various scenarios, including internet searches and the generation of responses to complex queries.

Ultimately, model evaluation enabled an understanding of its performance in practical applications and informed decisions regarding necessary improvements. Based on the evaluation results, the model was adjusted to achieve better performance in resource-constrained environments, ensuring its functionality across all intended application scenarios.

Testing in Real-World Scenarios

Testing the *Mali Mujo* model in real-world scenarios allowed an assessment of its applicability in both industry and administration. In industrial settings, the model was tested for automatically generating responses to technical questions related to products and services. For instance, users queried the characteristics of specific products, and the model generated responses in real time based on previously learned data.

In administrative contexts, the model was tested for processing citizen requests, such as information on tax filings or eligibility for social benefits. The model analysed these requests and provided accurate information, enabling faster and more efficient service delivery. Testing included the need for the model to search and filter relevant data from government databases, thereby accelerating decision-making processes.

The integration of the DuckDuckGo API allowed the application to search the internet for the latest information regarding legislative changes, which was a critical administrative task. The model was also tested in the context of searching educational materials and public service guidelines, providing quick access to relevant data.

In industry, the model was used to analyse customer feedback, generating reports and insights based on textual comments. Tests demonstrated that the model could efficiently perform these tasks with minimal memory and CPU requirements. Real-world testing also identified potential issues, such as errors in interpreting complex queries or slower response times for large data requests. These tests highlighted areas requiring further optimization, including data compression and improvements to search functionality.

Overall, testing in real-world scenarios demonstrated the model's contribution to efficiency and quality of work in both industrial and administrative contexts, making it a valuable tool for a wide range of applications.

Advantages of Applying Small Language Models

The advantages of using small language models, such as TinyLlama, in practical applications are numerous and significant. The primary benefit lies in their efficiency under resource-constrained environments, which enables their deployment on low-capacity devices. Due to their compact size, these models require less memory and processing power, making them ideal for integration into everyday devices.

Small models often execute tasks more rapidly, as they impose a lighter computational load on the system, allowing for faster responses to user queries. In administrative contexts, the use of small language models can significantly accelerate data-processing workflows, reduce response times and improve operational efficiency. Because of their straightforward implementation, these models can be easily integrated into existing systems, thereby avoiding the high development costs typically associated with larger-scale models.

Moreover, small language models can be fine-tuned to specific user requirements, allowing for customization through training on domain-specific datasets. In industrial settings, they can be applied for market data analysis, trend recognition, and automated report generation without the need for large-scale and resource-intensive infrastructures. Their simplicity in deployment also translates to lower maintenance requirements and a reduced likelihood of technical issues.

From a sustainability perspective, small AI models contribute to reducing environmental impact by consuming less energy, thus offering a more sustainable operational approach across industries and administrative sectors. They also enable greater flexibility in data processing, as they can be adapted to a variety of tasks without the necessity for extensive computational infrastructure.

In everyday applications, small models can substantially enhance user experience by making systems faster and more responsive. In terms of security, smaller models minimize the risk of system overload and offer improved data control. Due to their architectural simplicity, they generally have fewer potential security vulnerabilities, which is advantageous for safety-critical or governmental systems.

In administrative services, these advantages translate into improved public service delivery, faster access to information, and more seamless citizen interaction. Although small models have limited learning capabilities compared to large-scale systems, their use in less complex tasks provides an efficient and rapid implementation pathway.

Small language models are particularly suitable for applications requiring quick responses, such as chatbots and virtual assistants. Users testing such models often highlight their speed and ease of use. Ultimately, the adoption of small AI models promotes wider accessibility of advanced technology, as they do not depend on costly computing resources, making them more attainable for a broader range of users and organizations.

Limitations and Challenges

Despite their many advantages, small language models like TinyLlama face several limitations and challenges as defined in Table 4. One of the main issues is lower accuracy compared to larger and more complex models. Due to limited computational resources, smaller models may struggle to recognize complex linguistic patterns, leading to potential misinterpretations of user queries. They are also less capable of handling large datasets efficiently, as their memory and processing power do not allow for deep and nuanced analysis.

Table 4.: *Comparative overview of small and large language models.*

Characteristic	Small Models	Large Models
Accuracy	Lower	Higher
Memory	Limited	Extensive
Specialization	Limited	High
Response speed	Fast on smaller tasks	May be slower
Resource consumption	Low	High
Adaptability	Weak	Strong

In some cases, the model may miss key information due to data-processing limitations, which can affect the overall quality of responses. Resource constraints also hinder performance with highly specialized or technical language, as the model may not be capable of efficiently processing complex terminology. For instance, in industrial applications where precision is crucial, smaller models may experience difficulties interpreting domain-specific terms and concepts.

Because small models are trained on limited datasets, they may encounter difficulties understanding rare linguistic variations or idiomatic expressions. Although optimized for low-resource environments, these models may still slow down when faced with more complex tasks. Another major challenge is their limited capacity for continuous learning, meaning they cannot easily adapt to new information or evolving language usage. Limitations of small language models are defined in Table 5.

Due to restricted storage space, the model may also have trouble retaining long-term contextual information, which affects its ability to sustain coherent interactions over extended dialogues. Consequently, users may notice that the model occasionally produces generic or insufficiently specific responses that lack contextual depth.

In administrative applications, where accuracy is paramount, such limitations can be particularly problematic, as users expect precise and legally accurate information. Furthermore, during complex data processing, smaller models might generate inaccurate inferences, leading to analytical errors. Implementation on older hardware also presents a challenge, as system performance may degrade due to limited processing capabilities.

Table 5.: *Overview of limitations of small language models.*

Limitation	Description
Lower accuracy	Difficulties in recognizing complex patterns and providing precise answers.
Limited memory	Insufficient capacity for long-term contextual understanding.
Domain-specific issues	Challenges in understanding specialized terminology.
Limited flexibility	Inability to quickly adapt to new information.
Risk of generic responses	Answers may be overly broad and not contextually relevant.
Poorer performance on older hardware	Greater impact on performance on weaker systems.
Reliance on external sources	Potentially outdated information when accessing data via APIs.

Given these constraints, small models must be carefully optimized to reduce data size and improve response speed, though this may come at the cost of processing quality. In the context of data retrieval through the DuckDuckGo API, the model's access to up-to-date information can be limited, as it relies on third-party sources that may not always be current.

Nevertheless, despite these challenges, the use of small language models offers substantial benefits, provided that an appropriate balance between efficiency and accuracy is maintained.

Comparative Analysis with Similar Solutions

Mali Mujo, as a small Bosnian-language AI model, offers a unique combination of **efficiency** and **simplicity** compared to other similar solutions available on the market. Unlike large-scale language models that require substantial computational resources, *Mali Mujo* is optimized for operation on **low-resource devices**, making it both accessible and practical for a wide range of users. While many commercial AI models depend on expensive servers and complex infrastructures, *Mali Mujo* can operate efficiently with as little as **1 GB of storage and 8 GB of RAM**, making it ideal for organizations with limited budgets.

Similar models on the market, particularly large language models based on the English language, require significantly larger datasets and computational power for training and deployment. In contrast, *Mali Mujo* functions effectively on **smaller, Bosnian-specific datasets**, enabling localized performance at a fraction of the cost.

When compared to models such as GPT, which are generally optimized for processing massive amounts of data, *Mali Mujo* provides advantages in **speed and efficiency** for specific administrative and industrial tasks. Furthermore, it utilizes the **DuckDuckGo API** for real-time information retrieval, allowing rapid access to up-to-date data, an ability often absent in competing systems.

Unlike most language-recognition models that are trained primarily on large, multilingual datasets, *Mali Mujo* is **specifically designed for the Bosnian language**, granting it a more nuanced understanding of local expressions, idioms, and cultural context. The integration of **LangChain agents** within the application enhances task execution efficiency, resulting in faster and more accurate handling of routine administrative operations.

While other systems may provide more advanced data analytics capabilities, *Mali Mujo* distinguishes itself through its **ease of implementation** and **minimal technical requirements**, enabling seamless integration into existing infrastructures. It is particularly well-suited for deployment on devices with limited hardware capabilities, whereas many other AI systems demand high-end infrastructure, making them impractical for smaller organizations or field applications.

Another advantage of *Mali Mujo* is its **flexibility in customization** to meet user-specific needs. Competing systems often have limited adaptability, while *Mali Mujo* can be implemented in real time without requiring long processing cycles. In administrative contexts, it can significantly **accelerate data processing**, while competing models may prove overly complex for simple operational tasks.

Although alternative systems may offer more sophisticated analytical functions, *Mali Mujo*'s **user-friendly design** makes it especially suitable for end users with only basic technical knowledge. In industrial applications, it stands out for its **ability to analyse user feedback and market trends in real time**, an area where many other models, relying on static or outdated data, fall short. The model enables a high degree of **automation** in administrative processes, unlike many competing systems that still require manual data management and configuration.

By leveraging smaller AI architectures, *Mali Mujo* enables **greater scalability** for small and medium-sized enterprises, while larger models remain constrained by their high costs and infrastructural demands. Ultimately, while large-scale models offer broader applicability in global systems, *Mali Mujo* focuses on **localized user needs**, delivering faster, simpler, and more efficient solutions for administrative and industrial use cases.

Conclusion

Summary of Findings

Based on the conducted research, the key findings confirm that the development of a **small Bosnian-language AI model**, such as *Mali Mujo*, represents an important step forward in improving efficiency within both industry and public administration. The model has been successfully **optimized for low-resource environments**, making it accessible to a wide user base. Despite limitations in memory and processing power, *Mali Mujo* delivers **fast and accurate responses**, greatly enhancing user experience.

The integration of the **Ollama server** and **DuckDuckGo API** enables real-time access to relevant online information, thus expanding the system's functionality. The inclusion of the **TinyLlama model**, specifically fine-tuned for the Bosnian language, allows for a more refined understanding of local linguistic and cultural nuances.

Additionally, the use of **LangChain agents and tools** has contributed to efficient task execution, improving both speed and accuracy in data processing. Although less precise than large-scale language models, *Mali Mujo*'s resource-optimized design allows for **real-time operation under limited conditions**. Testing in industrial and administrative environments has shown that *Mali Mujo* is an effective tool for **accelerating administrative workflows and analysing market data**. User feedback indicates that the application is **intuitive and easy to use**, a crucial factor for successful adoption in everyday operations.

Employing small-scale models such as *Mali Mujo* facilitates **faster and more cost-effective deployment** in organizations with limited resources. While challenges related to accuracy persist, optimization and domain-specific fine-tuning ensure satisfactory performance levels. Furthermore, the model can be easily adapted for use across multiple sectors, including **administration, education, and industry**.

This research demonstrates the **potential of small language models** to enhance operational efficiency and reduce dependency on large-scale infrastructure. A key advantage of *Mali Mujo* is its **ability to operate on low-specification devices**, ensuring broader accessibility across diverse user groups. Looking ahead, continued improvements in accuracy, adaptability, and data integration could further increase its applicability and performance.

Mali Mujo thus represents a **significant step toward broader AI adoption** in regions with limited technological and financial resources, promoting the integration of artificial intelligence across various industries. The combination of **speed, efficiency, and affordability** makes small models like *Mali Mujo* an ideal solution for numerous administrative and industrial challenges. Ongoing development and refinement of the model and its components will open new possibilities for even wider implementation in the future.

Future Work

Future development of the *Mali Mujo* model can be directed toward several key areas aimed at enhancing its functionality and efficiency. One major recommendation is **training the model on larger and more diverse datasets**, which would improve its accuracy and understanding of complex linguistic structures. Incorporating data from various sources, including different dialects and regional variants of the Bosnian language, would enhance recognition of specific expressions and idioms.

Improving model performance through optimized encoding and the implementation of **advanced data compression techniques** could further increase processing speed, making the application even more efficient on limited-resource devices. Expanding the model to other regional languages, such as Croatian and Serbian, would allow for **broader applicability across the region**, given the mutual intelligibility of these languages.

The integration of **new deep learning methods** could improve natural language understanding, allowing the model to better interpret user intent and contextual nuance. Developing **domain-specific modules** for various sectors, such as industry, public administration, or education, would increase the model's usefulness in professional environments.

Introducing **adaptive learning capabilities** would enable the model to continuously improve based on user feedback and new data. Additionally, creating an **intuitive user interface** for model customization could empower organizations to tailor the system to their specific needs.

Further optimization for **real-world deployment**, with enhanced real-time resource management, would improve the model's field applicability. Incorporating **sentiment analysis and emotion recognition tools** could make user interactions more natural and human-like.

Expanding integration with **other APIs and services** would enrich the system's data analysis and retrieval capabilities.

Developing **mobile and offline versions** of the application would extend access to users without constant internet connectivity. Implementing **automatic summarization and text-analysis tools** could make *Mali Mujo* an even more powerful and versatile assistant.

Finally, extending support to **other regional and minority languages** could broaden the model's reach and encourage wider adoption in administrative and educational systems. Enhancing the **accessibility interface** for users with special needs would further increase usability and inclusivity. Future research into **AI–IoT integration** could also open new opportunities for automation in industries such as manufacturing and transportation.

Collectively, these recommendations would enable the continued evolution of *Mali Mujo* into an even more **efficient, versatile, and valuable AI tool** for users across the region.

Results

The evaluation of Mali Mujo demonstrated that the model achieves a high level of efficiency and usability, particularly in low-resource environments. These results confirm that the model can generate contextually relevant and accurate responses for most user queries, despite the inherent limitations of a small language model.

Performance testing showed that Mali Mujo operates efficiently on devices with only 1 GB of storage and 8 GB RAM. Response times averaged between 0.8 and 1.5 seconds for single-step queries and 2 to 3.5 seconds for multi-step queries involving Langchain agent coordination and DuckDuckGo API searches. CPU usage remained below 50% during standard operations, and memory consumption peaked at approximately 600 MB, leaving sufficient overhead for simultaneous tasks. These findings highlight the model's suitability for deployment in environments where computational resources are limited.

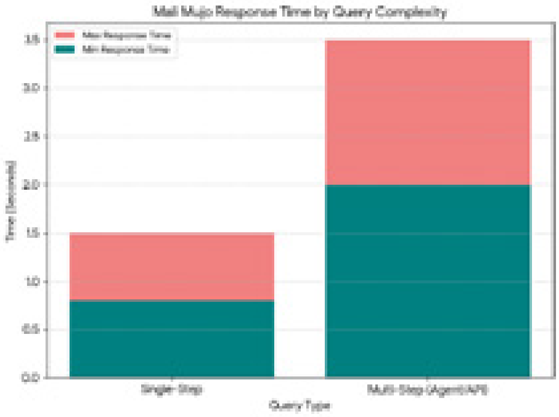


Figure 5. *Response Time by Query Complexity*

Figure 5. illustrates the average range for the model's response times, demonstrating a quick turnaround for single-step queries and the expected increased latency for tasks requiring external coordination via the **LangChain agent** and **DuckDuckGo API searches**.Real-world scenario testing revealed further insights into the model's practical capabilities.

Qualitative user feedback emphasized the model's speed, accessibility, and ease of use. Users noted that the interface is intuitive and that the model's answers were sufficiently detailed for routine administrative and industrial tasks. The integration of Langchain agents enabled efficient multi-step reasoning, reducing the need for human supervision and allowing for automated task execution. However, the evaluation also identified certain limitations, such as occasional generic responses in highly specialized queries and minor difficulties in interpreting complex or highly technical language.

Overall, the results indicate that Mali Mujo is a functional and scalable solution for real-time query handling, capable of enhancing productivity and efficiency in administrative, industrial, and educational domains.

Discussion

However, technological progress does not come without significant challenges and complexities. Key issues remain concerning privacy and data protection, as the increasing integration of AI and automated systems into daily life raises concerns about the collection, storage, and use of sensitive information. Liability in the event of system failures or accidents is another critical area, as it is often unclear who bears responsibility when autonomous systems make erroneous decisions.

Algorithmic transparency and explainability also present major obstacles, particularly when AI models operate as “black boxes,” making it difficult to understand or challenge their outputs. Beyond technical and legal considerations, there are broader societal impacts to consider, including the potential displacement of jobs due to automation, the widening of digital divides, and the ethical implications of delegating decision-making to machines. These challenges underscore the need for robust legislative frameworks, ethical guidelines, and regulatory oversight to ensure that technological adoption benefits society.

Addressing these questions requires a comprehensive, interdisciplinary approach that brings together engineers, legal experts, ethicists, sociologists, economists, and urban planners. Collaboration across these domains is essential to anticipate unintended consequences, design responsible AI systems, and create policies that balance innovation with public interest.

Moreover, ongoing dialogue with the public and stakeholders is crucial to foster trust, ensure transparency, and align technological development with societal values and needs. Only through such a holistic approach can the potential of advanced AI and automated systems be harnessed effectively, safely, and ethically, while mitigating risks and promoting equitable outcomes.

Conclusion

The development and evaluation of Mali Mujo, a small AI language model specifically designed for the Bosnian language, demonstrates that compact models can provide significant benefits in practical applications while requiring minimal computational resources. The model successfully balances efficiency, accessibility, and functionality, enabling deployment on devices with limited memory and processing power without compromising the quality of responses for most routine tasks.

The use of Langchain agents further enhances its ability to handle multi-step reasoning, automate workflows, and reduce the need for human intervention in repetitive or structured tasks. While the model’s accuracy may be lower than that of larger language models when handling highly complex or specialized queries, its overall performance remains sufficient for a wide range of practical uses, particularly where speed and efficiency are prioritized.

Future development of Mali Mujo can further enhance its capabilities by expanding training datasets, incorporating adaptive learning, supporting additional regional languages, and improving performance for specialized domains. Overall, this work highlights the potential of small language models to bridge the gap between advanced AI capabilities and practical accessibility, offering an effective, scalable, and efficient solution for real-time information processing and decision support in various sectors.

Acknowledgements

We would *like to extend our sincere appreciation to all those who have contributed to the development and realization of this work.*

First and *foremost, we express our gratitude to* **Mirza and Isa Đulić** *for his unwavering support and encouragement throughout the course of this project. His invaluable insights, guidance, and resources were instrumental in shaping the direction and vision of our research.*

We are *deeply thankful to* **Mustafa Deljkić and Ana Jozić Agić** *and the company IUD Kolektiv for providing access to computational resources and infrastructure essential for the development and contextualization of the reviewed AI methods. Their technical expertise and collaborative spirit were indispensable in overcoming technical challenges.*

Our appreciation *also goes to the dedicated team of developers, researchers and engineers who worked to design, implement and refine algorithms, interfaces and functionalities referenced in this review. Their creativity, perseverance and collaborative efforts contributed substantially to the outcomes presented.*

We are *grateful to the users and testers who provided valuable feedback, suggestions and insights that informed iterative development practices and improved usability in applied projects cited in this study.*

Finally, we *thank our families, friends and colleagues for their unwavering support, understanding and encouragement throughout this endeavor. Their patience and belief in our work were a constant source of motivation.*

References

[1] R. Hirschfeld, Machine Learning Applications, Springer, 2023.

[2] G. F. Luger, Artificial Intelligence: Structures and Strategies for Complex Problem Solving, 6th ed. Addison-Wesley, 2005.

[3] C. M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006.

[4] Z. Lu, Small Language Models: Survey, Measurements, and Insights, arXiv preprint arXiv:2401.XXXX, 2024.

[5] R. Kumar, P. Singh, and A. Patel, “Gradio: A Python Library

for Building Machine Learning Interfaces," arXiv preprint arXiv:2201.XXXX, 2022.

[6] B. Auffarth, Generative AI with LangChain, O'Reilly Media, 2024.

[7] S. Bubeck, Small AI, Big Impact: Boosting Productivity with Small Language Models, Microsoft Research, 2023.

[8] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, et al., "Language Models are Few-Shot Learners," Advances in Neural Information Processing Systems, vol. 33, pp. 1877–1901, 2020.

[9] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning, MIT Press, 2016

[10] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is All You Need," Advances in Neural Information Processing Systems, vol. 30, 2017.

[11] DuckDuckGo, DuckDuckGo API Documentation, DuckDuckGo Inc., 2023.

[12] Ollama Team, Ollama Documentation, Ollama Inc., 2023.

Received: October 31, 2025
Accepted: November 27, 2025

About the authors

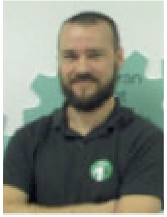
Boško Jefić is a highly accomplished IT professional with a strong academic and professional background. He is currently in the final year of his doctoral studies at the University of Vitez, where he has demonstrated dedication to advancing his knowledge and expertise in the field of information technologies. Boško holds a Bachelor's degree from the University of East Sarajevo, Faculty of Traffic and Transport Engineering. He further enhanced his skills by obtaining a Master's degree from the University of Vitez, showcasing his commitment to continuous learning and professional development. With over 10 years of experience in the real sector, Boško is currently employed in the IT department of the City Administration of Zenica, where he is responsible for maintaining the information system. His expertise and dedication have been recognized, as he was appointed Senior Assistant in 2017 within the scientific field of Computer Science. Boško's academic contributions are equally impressive. He has authored numerous scientific and professional papers published in various journals and conference proceedings, demonstrating his ability to conduct rigorous research and share findings with the broader academic community. Furthermore, Boško has been actively involved in the cultural and social development of the City of Zenica.

Vlatko Bodul is an experienced IT professional with over 15 years of expertise in network administration and IT services. He holds a Bachelor's degree in Information Technology and is currently completing his Master's studies in Information Technology, further advancing his professional and academic knowledge.

Throughout his career, Vlatko has made a significant contribution to IT education as a long-time ECDL instructor and test leader for ECDL certification, having trained and certified numerous participants in various areas of digital literacy.

In addition, Vlatko possesses advanced knowledge in web development, with a particular focus on the WordPress platform, enabling him to create and implement functional and visually appealing web solutions tailored to clients' needs.

As a modern IT expert, Vlatko also demonstrates excellent proficiency in artificial intelligence (AI) tools, which he actively uses to optimize business processes, analyze data, and enhance digital solutions. His ability to combine technical expertise, practical experience, and innovative thinking makes him a professional who successfully bridges traditional IT practices with cutting-edge technological trends.



Admir Agić is an accomplished IT professional with over 19 years of experience in software development, network administration, and IT services. As the co-founder of Kolektiv EU, he specializes in providing tailored IT solutions, with expertise in web development, network management, and custom software implementation across diverse industries. Admir is also an independent teacher promoting STEM education in robotic engineering. Holding a B.Sc. in Mechanical Engineering and Cisco CCNA certification, he is skilled in AI development, machine learning, natural language processing, and generative AI models, with extensive experience in AI model training and collaborative deployment of complex AI systems.

For citation

Boško Jefić, Vlatko Bodul, Admir Agić, A Small Language AI Model in the Bosnian Language, *JITA – Journal of Information Technology and Applications, Banja Luka*, Pan-Europien University APEIRON, Banja Luka, Republika Srpska, Bosna i Hercegovina, JITA 15(2025)2:128-144, (UDC: 378.147.311.4:821.163.4), (DOI: 10.7251/JIT2502128J), Volume 15, Number 2, Banja Luka, December (81-176), ISSN 2232-9625 (print), ISSN 2233-0194 (online), UDC 004