universität wien

MASTERARBEIT | MASTER'S THESIS

Titel | Title

A Hybrid System for Serbian Natural Language Processing with Large Language Models

verfasst von | submitted by

Ivan Cvetanović BSc

angestrebter akademischer Grad | in partial fulfilment of the requirements for the degree of

Master of Science (MSc)

Wien | Vienna, 2026

Studienkennzahl lt. Studienblatt | Degree programme code as it appears on the student record sheet:	UA 066 921
Studienrichtung lt. Studienblatt | Degree programme as it appears on the student record sheet:	Masterstudium Informatik
Betreut von | Supervisor:	ao. Univ.-Prof. Mag. Mag. Dr. Werner Winiwarter

Acknowledgements

I want to express my gratitude to Werner Winiwarter for supporting and guiding me throughout my Bachelor's thesis, practical courses, and now Master's thesis.

I would also like to thank the native Serbian speakers who volunteered their time to take part in the evaluation study. Their judgements were essential to the human assessment in this work.

I also want to thank my family and close friends for the continuous support and patience throughout my studies.

Finally, I would like to acknowledge the financial support of the Republic of Austria and the computing resources provided by the Faculty of Computer Science at the University of Vienna during my studies.

Abstract

Serbian is a low-resource language for NLP, underrepresented by current tools and models, which are mainly built for and evaluated on English. This thesis investigates the limitations of current tools, fine-tunes an open-weight large language model for Serbian NLP, and develops a unified web application that integrates these capabilities in a hybrid system. The model is fine-tuned on an instruction-formatted Serbian dataset of 24,601 examples across five tasks, and is then evaluated through automatic benchmarks, task-specific metrics, and a user study with native speakers. The choice of evaluation method substantially affects the measured performance. The same base model scored ≈34% under log-probability scoring but ≈76% under chat-based generation on the identical benchmark. Fine-tuning improved performance on multiple task-specific metrics and on instruction following, but the study found that native speakers did not prefer the fine-tuned outputs overall, and for summarization they strongly preferred the base model's outputs. Automatic metrics and human judgement diverged substantially. Although the automatic summarization metric improved, the raters judged the fine-tuned summaries worse, because they were shorter and less faithful. The work contributes a fine-tuned open-weight Serbian model and a locally self-deployable application, but its central finding is methodological. Detailed testing of low-resource languages requires multiple complementary methods, since conclusions depend heavily on the chosen method.

Kurzfassung

Serbisch ist für die maschinelle Sprachverarbeitung (NLP) eine ressourcenarme Sprache, die von aktuellen Werkzeugen und Modellen unterrepräsentiert wird, da diese hauptsächlich für das Englische entwickelt und daran evaluiert werden. Die vorliegende Arbeit untersucht die Grenzen aktueller Werkzeuge, stimmt ein großes Sprachmodell (LLM) mit offenen Gewichten für die maschinelle Sprachverarbeitung des Serbischen fein ab und entwickelt eine einheitliche Webanwendung, die diese Fähigkeiten in ein hybrides System integriert. Das Modell wird auf einem im Instruktionsformat vorliegenden serbischen Datensatz mit 24.601 Beispielen über fünf Aufgaben hinweg feinabgestimmt und anschließend mittels automatischer Benchmarks, aufgabenspezifischer Metriken und einer Nutzerstudie mit Muttersprachlern evaluiert. Die Wahl der Evaluationsmethode beeinflusst die gemessene Leistung erheblich. Dasselbe Basismodell erreichte auf demselben Benchmark etwa 34 % bei Log-Wahrscheinlichkeits-Bewertung, jedoch etwa 76 % bei chatbasierter Generierung. Die Feinabstimmung verbesserte die Leistung bei mehreren aufgabenspezifischen Metriken sowie beim Befolgen von Anweisungen. Die Studie ergab jedoch, dass Muttersprachler die feinabgestimmten Ausgaben insgesamt nicht bevorzugten und bei der Zusammenfassung die Ausgaben des Basismodells deutlich vorzogen. Automatische Metriken und menschliches Urteil wichen erheblich voneinander ab. Obwohl sich die automatische Metrik verbesserte, beurteilten die Bewertenden die feinabgestimmten Zusammenfassungen als schlechter, da sie kürzer und der Quelle weniger treu waren. Die Arbeit liefert ein feinabgestimmtes serbisches Modell mit offenen Gewichten und eine lokal eigenständig betreibbare Anwendung. Ihr zentraler Befund ist jedoch methodischer Natur. Eine detaillierte Prüfung ressourcenarmer Sprachen erfordert mehrere komplementäre Methoden, da die Schlussfolgerungen stark von der gewählten Methode abhängen.

Contents

List of Tables

List of Figures

List of Algorithms

Acronyms

ABSA Aspect-Based Sentiment Analysis.

API Application Programming Interface.

ARC AI2 Reasoning Challenge.

ASR Automatic Speech Recognition.

BCS Bosnian–Croatian–Serbian.

BLEU Bilingual Evaluation Understudy.

CER Character Error Rate.

GPU Graphics Processing Unit.

JSON JavaScript Object Notation.

LDA Latent Dirichlet Allocation.

LLM Large Language Model.

LoRA Low-Rank Adaptation.

MMLU Massive Multitask Language Understanding.

MT Machine Translation.

NER Named Entity Recognition.

NLP Natural Language Processing.

PIQA Physical Interaction: Question Answering.

POS Part of Speech.

ROUGE Recall-Oriented Understudy for Gisting Evaluation.

SRL Semantic Role Labelling.

TF-IDF Term Frequency–Inverse Document Frequency.

VRAM Video Random-Access Memory.

WER Word Error Rate.

1. Introduction

1.1. Motivation

Alongside the advancement of Large Language Models (LLMs), Natural Language Processing (NLP) has also achieved rapid development [VSP+17] and can now handle tasks such as grammar correction, translation, sentiment analysis and summarization across different domains. This progress is, however, concentrated on high-resource languages, like English, while many other languages remain underserved [JSB+20].

1.2. Problem Statement

Serbian is one such language and poses a great challenge for NLP because of its complexity: it has seven cases, three genders, complex verb conjugation, relatively free word order and digraphia (both Cyrillic and Latin scripts) [MAM23]. Furthermore, the scarcity of training data compounds these difficulties, making Serbian a valuable stress-test for LLM adaptability [AU25].

1.3. Aim and Research Questions

The overall aim is to provide a unified web application, offering an extensive pipeline for analysing the Serbian language with the help of LLMs and rule-based systems. Furthermore, an additional fine-tuned model dedicated to these processing tasks, which will be rigorously evaluated, is also planned. The first question to be covered is if the fine-tuning improves performance on automatic benchmarks versus the base model. The second question is whether native speakers prefer the outputs of the fine-tuned model. Finally, the trade-offs of deploying a fine-tuned model versus the task-specific classifiers will also be examined.

1.4. Contributions of the Thesis

The thesis delivers both a system and an empirical investigation. The system itself offers a broad set of Serbian NLP tasks, such as morphological, syntactic, semantic and pragmatic analyses. It also offers grammar correction, sentiment, translation, hate-speech detection, summarization, topic modelling and semantic role labelling. The single Flask-based interface offers multiple visualizations, like a word cloud, a nested donut chart, a heatmap and an interactive dependency tree. Beyond analysis, the system supports both Latin

and Cyrillic input and can switch between local and remote models, depending on what the user chooses.

An open-weight model (Qwen3) was fine-tuned on a purpose-built Serbian multi-task dataset, consisting of 24,601 instruction-formatted entries. The dataset covers entries for training summarization, grammar correction, translation, aspect-based sentiment and sentiment. The data construction methodology leveraged Wikipedia-derived Serbian text [ŠP26] and teacher-model-generated examples. The fine-tuned model matched or improved on the base model's automatic scores on the tasks it was trained on, while leaving the general-knowledge benchmark essentially unchanged, which is covered in detail in Chapter 5.

Evaluating the instruction-tuned Serbian models by log-probability scoring compared to chat-style generation produced sharply different results: 34.3% versus 75.6% accuracy for the same model on the same benchmark, meaning that the standard log-probability harness severely underestimates the performance of instruction-tuned models such as the base Qwen3-14B-Instruct.

Finally, a blind, controlled human evaluation was conducted, comparing the base and fine-tuned models on generative tasks. The judging was done by native Serbian speakers, who rated the output quality, with inter-annotator agreement also reported.

All these contributions span engineering the application, modelling the fine-tune, and evaluating the benchmark and the user study. The overarching finding is methodological: for a low-resource language like Serbian, the conclusions drawn about a model's quality depend heavily on how it is evaluated (by log-probability or chat-based scoring, and by automatic metrics or human judgement), so a reliable assessment requires multiple complementary methods.

1.5. Thesis Structure

Chapter 2 provides the background: why the linguistic properties of Serbian make it a challenge for NLP, the principles of LLMs and fine-tuning, the tasks the system addresses, the existing tools for Serbian that the system relies on, the earlier language models for Serbian and the usual approach to low-resource instruction tuning, and how instruction-tuned models are evaluated.

Chapter 3 presents the evolution of the web application from a morphological-analysis tool into a unified multi-task platform, and identifies its current limitation, the dependence on a general-purpose model, which is one of the motivations for the thesis.

Chapter 4 describes the fine-tuning process: what model was selected and why, the construction of the multi-task training data, training infrastructure and configuration, and the final deployment of the resulting model.

Chapter 5 shows how the automatic benchmark evaluation was realized, including the log-probability versus the chat-based methodology finding and the task-by-task comparison of the base and fine-tuned models.

Chapter 6 reports on the results of the native-speaker user study, while Chapter 7 discusses the findings and limitations, and Chapter 8 concludes.

2. Background and Related Work

2.1. The Serbian Language as an NLP Challenge

Serbian is a South Slavic language spoken by an estimated twelve million speakers worldwide [MAM23], and is categorized as a "low-resource" language for NLP, not because it lacks speakers, but because annotated data and dedicated tools have historically been scarce [MAM23, JSB+20]. This scarcity, combined with the language's complexity, makes it a demanding task for adapting language models [AU25].

Serbian is a morphologically rich language: it has three genders, nouns inflect for seven cases, verbs conjugate for multiple tenses and aspects, and a single lemma can appear in many distinct word forms [MAM23]. This increases data sparsity, making the models see fewer examples of each form during training, which is exactly the kind of problem that more training data and adaptation are meant to mitigate.

Serbian also has relatively free word order, so the grammatical roles are carried by morphology and not by position, which many models rely on [MAM23]. This reliance on positional regularities, which works well on languages like English, is why syntactic tasks like dependency parsing and role labelling are harder to do for the Serbian language.

The next difficulty is digraphia. Serbian is written in both Latin and Cyrillic scripts and the same text can appear in either script [MAM23]. This doubles the surface variation and requires transliteration handling for components that are specialized in only one script, which will be covered in Chapter 3.

Apart from the script, Serbian also has two standard pronunciations, called Ekavian and Ijekavian, and both of them are officially correct. The difference between them comes from the old Common Slavic vowel *jat*. The same word is written with *e* in Ekavian, but with *(i)je* in Ijekavian, for example Ekavian млеко / *mleko* and Ijekavian млијеко / *mlijeko* ('milk') [Ale06]. Ekavian is the standard used around Belgrade, while Ijekavian is used in western Serbia, Montenegro and Bosnia. For NLP this adds another layer of variation on top of the digraphia problem, since the same word can be written in several ways. This is a problem because the scarce Serbian training data does not cover all of these forms equally.

These properties, combined, mean that general-purpose models, predominantly trained on mainstream languages, are a poor fit out of the box for Serbian, motivating the dedicated resources surveyed later in this chapter and the fine-tuning approach in this thesis [AU25].

2.2. Large Language Models and Fine-Tuning

Modern NLP is built around the Transformer architecture, which replaced recurrence with self-attention and enabled training at scale [VSP+17]. Models like BERT learned general-purpose language representations from large unlabelled corpora [DCLT19], and autoregressive models scaled this further, showing that a single model could perform multiple tasks [B+20].

The mechanism our web application relies on is the following: instruction tuning trains a model to follow natural language task descriptions, such that one model can perform grammar correction, translation, summarization and more from prompts alone, without task-specific architectures [WBZ+22, OWJ+22]. This is why a single fine-tuned model is a practical solution for the backend of a multi-task application.

During the fine-tuning process, two approaches were used and compared. Full fine-tuning updates all model weights and can achieve good task adaptation but is very expensive in memory and compute. Parameter-efficient methods, such as Low-Rank Adaptation (LoRA), freeze the model weights and train small low-rank matrices, drastically reducing the memory and compute necessary while also achieving good fine-tune quality [HSW+22]. This quality can be further improved by utilizing the full fine-tune.

In this work, LoRA was used during the initial exploratory phase in order to test the impact of the datasets used on the quality of the output. Based on the results, the final model was trained with full fine-tuning, updating all model parameters and achieving stronger results.

Qwen3 [Y+25], which belongs to an open-weight instruction-tuned model family, was chosen for the fine-tuning. The rationale behind this is that its open weights allow local deployment and full control over the training data, which matters for low-resource languages, where data provenance and on-premise serving are concerns. The exact rationale is detailed in Chapter 4.

2.3. NLP Tasks Covered in This Work

Our application covers several NLP tasks for the Serbian language. This section defines each of them and their approaches. Table 2.1 gives a worked input/output example for each of the five generative tasks.

Grammar correction. Detecting and correcting grammatical, orthographic, and agreement errors while also preserving the core structure and meaning of the sentence. In the past, rule-based and statistical systems were used to tackle this, while today it is dominated by neural and LLM approaches [BYQ+23]. Grammar correction is especially demanding in the Serbian language, since a single wrong case ending or gender/number mismatch is a grammatical error, and Serbian has seven cases and three genders, so the space of possible errors is large. Unlike in English, where many errors are lexical or word-order based, a model correcting Serbian has to track morphosyntactic agreement across the sentence.

Translation. Machine translation is the automatic conversion between the source and

the target language, in this case between Serbian and English. The translation process for Serbian is complicated due to the fact that it is very context-dependent regarding the morphology and that it has free word order. A single sentence with the same meaning can be written in multiple ways, while still retaining the same translation to English. Meaning often depends on the case and clause structure rather than position. Furthermore, a particle like "da" can have multiple functions which the translator needs to detect and properly use according to other words in the sentence. The approach used in this thesis is Transformer-based [VSP+17].

Sentiment analysis. The process of classifying the polarity of the input among positive, negative, and neutral [Liu12], here using a Transformer-based model [BEACC22]. The real challenge in the case of Serbian is the use of mixed polarity, combined with the fact that most current sentiment models are trained on multilingual social-media text, which does not transfer well to Serbian, especially if longer texts are used [Liu12, BEACC22].

Aspect-based sentiment analysis (ABSA). ABSA is responsible for identifying the aspect terms of the input text and assigning sentiment to each one, such that one sentence can carry positive and/or negative sentiment towards specific aspects. ABSA can be decomposed into aspect extraction and aspect polarity classification. The extraction step is typically harder and involves finding the terms. The classification step is just like sentiment analysis, but applied to the terms found. Aspect extraction is particularly hard in free-word-order, morphologically rich languages because the positions of the terms vary and can be in an inflected form, so the usual positional heuristics fail [PGP+16].

Summarization. Summarization is the process of condensing the text to its essential content while preserving the meaning. Two approaches were used: extractive and abstractive summarization. TextRank [MT04] was used for extractive and a Transformer model for abstractive [HBI+21] summarization. Extractive methods select only the most important sentences while deleting the rest, whereas abstractive methods generate new text that may rephrase the original content or fuse information together. The extractive method used in the thesis is language-independent and does not need any training data; however, it cannot fuse the information together or rephrase a sentence. Abstractive methods are more fluent, but risk producing unfaithful text [MNBM20]. Abstractive summarization for Serbian is constrained by the scarcity of high-quality Serbian training data.

Hate speech detection. The detection of hateful and toxic content and its assignment to different categories, handled by a dedicated classifier rather than a generative model [SW17, KMS+25].

Semantic role labelling (SRL). The process of identifying who did what to whom [GJ02]. SRL assigns roles (agent, patient, etc.) to constituents relative to the predicate, which is also outside the scope of fine-tuning.

Table 2.1.: Example input and expected output for each of the five core generative tasks.

Task	Example input (Serbian)	Expected output
Grammar correction	Постојале су и поебне полицијске снаге. *(There were also special police forces — with a spelling error.)*	Постојале су и посебне полицијске снаге. *(There were also special police forces.)*
Translation (sr→en)	Паприкаш је јело са месом и кромпиром. *(Paprikash is a dish with meat and potatoes.)*	Paprikash is a dish with meat and potatoes.
Sentiment	Smatram da je povređen čl. 106. i 107. *(I consider Articles 106 and 107 to have been violated.)*	negative
Aspect-based sentiment (ABSA)	gledao ga pre godinu, fantastičan film, često ga se setim :D *(watched it a year ago, fantastic film, I often remember it :D)*	*film* → positive
Summarization	Основани су 12. марта 1911. године као крикеташка екипа. Први пут су постали прваци 1924. године. [. . .] *(Founded on 12 March 1911 as a cricket team. They became champions for the first time in 1924. [. . .])*	Фудбалски клуб Аустрија Беч је аустријски фудбалски клуб из града Беча. Били су 24 пута прваци и 26 пута освајачи купа. *(FK Austria Wien is an Austrian football club from the city of Vienna. They were champions 24 times and cup winners 26 times.)*

2.4. Existing Resources and Tools for Serbian

The resources available for Serbian exist, but are scattered, and each was built for a different purpose than the one pursued in this thesis. On the analysis side, CLASSLA-Stanza is an automatic annotation pipeline for South Slavic languages that performs tokenization, lemmatization, POS tagging, named-entity recognition and dependency parsing for Serbian, including non-standard web text [TL23]. It only labels existing text with linguistic structure and does not generate or transform it, so it provides no instruction-response data and can act as a preprocessing aid at best. On the data side there are large raw corpora. CLASSLA-web collects South Slavic web text by iterative crawling of national domains and adds document-level genre and topic metadata [KPRSL26], and a related pipeline cleans Wikimedia dumps into plain-text corpora for the same languages [ŠP26]. Both are suitable for pretraining or continued pretraining, but they are unlabelled and carry no task supervision and no instruction-formatted pairs, and the web crawl additionally contains a growing portion of machine-generated content that is a quality risk if used as training material.

The task-specific resources are closer to the target tasks, but each one covers only a single task. ParlaSent provides parliamentary sentences manually annotated with sentiment labels together with a fine-tuned classifier [MRL24], yet the labels are discrete and the released model is discriminative rather than generative, the domain is limited to parliamentary proceedings, and the annotation is sentence-level rather than aspect-based.

XL-Sum supplies article-to-summary pairs scraped from BBC News in both Serbian scripts [HBI+21], but it is a single-task and single-domain dataset whose Serbian portion is a thin slice and which contains no instructions. The multilingual encoders are similar. XLM-R (XLM-RoBERTa) is an encoder-only masked language model covering around a hundred languages including Serbian [CKG+20], and XLM-T extends it with further pretraining on multilingual tweets for social-media sentiment classification [BEACC22]. Both output labels rather than free text and are not instruction-tunable in a generative sense, so they can back the classification subtasks of the pipeline but cannot themselves provide a generative model. For evaluation, the Serbian LLM benchmark suite supplies the test material used in this thesis [dat24].

Taken together, these resources are either analysis-oriented, raw and unlabelled, restricted to a single task and domain, or classification-only. They each cover a separate piece of the problem, but none of them is an instruction-tuned generative model, and none unifies grammar correction, translation, sentiment and aspect-based sentiment analysis, and summarization under one model that follows natural language prompts. To our knowledge, such a unified generative model for Serbian is still missing at this point. This thesis aims to change this by building a single open-weight instruction-tuned model that can be deployed locally and that spans these tasks, reusing the resources above where they fit while supplying the instruction-formatted Serbian data that they lack.

2.5. Language Models for Serbian and Low-Resource Instruction Tuning

Apart from the task-specific resources described in the previous section, there are also a few language models made directly for Serbian and the closely related Bosnian–Croatian–Serbian (BCS) languages. One of them is BERTić, an ELECTRA-based transformer trained on around eight billion tokens of Bosnian, Croatian, Montenegrin and Serbian web text, which improves part-of-speech tagging, named-entity recognition and similar tasks over multilingual models [LL21]. However, just like XLM-R, it is a discriminative model. This means it only labels text rather than generating it, so it can handle the classification subtasks but cannot follow instructions or produce free text. On the generative side there is YugoGPT, an open-weight 7B base model for BCS, which outperformed general-purpose models like Mistral and Llama 2 on Serbian [Gor23]. However, it is a single general-purpose base model, not one that was specialized and evaluated on a fixed set of Serbian generative tasks with matching training data.

A broader approach for bringing instruction following to under-served languages is massively multilingual instruction tuning. For example, Okapi instruction-tunes large language models in 26 languages, using translated instruction data and reinforcement learning from human feedback [LNN+23]. The Aya project goes even further and covers around a hundred languages, with an instruction-tuned multilingual model and a large, partly human-made instruction set [ÜAY+24]. The problem with these efforts is that they spread the model and the data over many languages at the same time, usually with

machine-translated or general instructions. This way, they cover a lot of languages, but they do not go deep into any single low-resource language.

This thesis takes a different approach. Instead of a general-purpose base model like YugoGPT, an encoder-only classifier like BERTić, or a small per-language part of a multilingual model like Aya or Okapi, it fully fine-tunes an open-weight model on Serbian data for five generative tasks, written in both scripts. The model is then evaluated with both automatic benchmarks and a native-speaker study. What sets this work apart from the models above is this specific combination: a locally deployable, multi-task Serbian generative model that works in both scripts and is evaluated by native speakers.

2.6. Evaluation Methodology for Instruction-Tuned Models

The evaluation of LLMs requires automatic metrics that compare the output of the model to the reference answers. The choice of these metrics depends on the task to be analysed. For example, machine translation uses the BLEU metric, which measures n-gram overlap between the model's translated output and one or more reference translations [PRWZ02]. For summarization, the standard metric is ROUGE, in particular ROUGE-L, which analyses the longest common subsequence between generated and reference summaries [Lin04]. For classification tasks such as sentiment and aspect extraction, performance is measured using the F1 score and accuracy. These metrics are typically computed using standard evaluation harnesses, which run the model over a benchmark dataset and score the outputs automatically, allowing for comparison between the models.

Instruction-tuned models complicate the evaluation picture, because there are two ways to obtain an answer from them on a benchmark. The first one is log-probability scoring, where the harness presents each candidate answer and selects the option to which the model assigns the highest probability, without letting the model generate any text freely. The second is the chat-style evaluation, where the model produces a response to a given prompt, which is then parsed and the predicted answer is extracted. These two methods can produce very different answers for the same model on the same benchmark, especially for instruction-tuned models like the one used here. The reason behind this is that instruction-tuned models are optimized to expect prompts and to give proper answers to them. They do not automatically know what the user wants as an output, so log-probability scoring can underrepresent the model's capabilities. This discrepancy is empirically examined in Chapter 5, where the same base model is shown to score lower under log-probability scoring than under generation-based evaluation.

While automatic metrics are necessary, they are insufficient for judging the quality of generated text, because they reward surface overlap with references and can miss fluency, faithfulness and adequacy. Therefore, they were additionally complemented with human evaluation, especially for open-ended generative tasks such as grammar correction, translation and summarization. Since human evaluation is subjective, its reliability must be quantified using the inter-annotator agreement. The agreement is commonly measured with statistics like Cohen's or Fleiss' kappa, and then interpreted using established agreement bands, like the slight/fair/moderate/substantial/almost-perfect scale [LK77].

Human evaluation conducted in this thesis, including its agreement analysis, is presented in Chapter 6.

This thesis evaluates the fine-tuned model using both approaches: automatic benchmarks in Chapter 5 and a native-speaker study in Chapter 6. By doing so, the thesis surfaces the methodological pitfall in the standard automatic evaluation for Serbian instruction-tuned models, namely that log-probability scoring cannot completely capture the performance of the model, and argues that generation-based and human evaluation give a more faithful picture.

3. System Evolution: From Analysis Tool to Unified Platform

This chapter's purpose is to show the evolution of the web application from a simple linguistic-analysis tool into a unified multi-task platform. The system was developed incrementally in the author's previous work and was divided into three stages, where each stage added a new layer of capability onto the previous one. The chapter will present the foundation, first extension, second extension, resulting architecture and finally the limitation that motivates the thesis.

These three stages build directly on the author's prior coursework: the foundation was developed in the author's Bachelor thesis [Cve24], the first extension during Praktikum 1 [Cve25], and the second extension during Praktikum 2 [Cve26]. This chapter consolidates that earlier work into a single account and updates it to match the deployed system, since several components were revised after those reports were written. The original contribution of this thesis, the fine-tuning of a specialized Serbian model and its evaluation, begins in Chapter 4 and does not overlap with the prior coursework.

3.1. Foundation: Morphological, Syntactic, and Semantic Analysis

The foundation is a web application for linguistic analysis of Serbian, built with the Flask micro-framework, made to tackle scattered Serbian NLP functionality behind a single interface. While there are tools that offer analyses for individual tasks, to our knowledge none of them combine morphological, syntactic and semantic analyses with visualization in one place – a situation similar to the fragmented resource landscape for Serbian [MAM23]. The application accepts both Latin and Cyrillic scripts. Cyrillic is transliterated to Latin and then passed on to the downstream components, addressing the digraphia challenge noted in Chapter 2. For non-Serbian speakers, the application can generate a random sample sentence drawn from a pool of Serbian folk tales.

The core analysis is morphological: input is tokenized, which means that the whole input is separated into multiple entries, where each entry is a word or a non-word token such as punctuation or a number. Afterwards it is lemmatized, which is a process of bringing the word back to its base form. Finally, part-of-speech tagging is applied, which is a process of categorizing each word of the input to its grammatical role, like noun, verb, adjective, etc. All of the above is done using CLASSLA, a pipeline for South Slavic languages [TL23]. Figure 3.1 shows the resulting per-token analysis for a sample sentence.

Original Word	Translation	Lemma (Base Form)	Local Definition	Online Definition	Word Type	Number	Person	Case	Gender	Head	Dependency Relation	Named Entity
Car	car	car	['a male monarch or emperor (especially of Russia prior to 1917)', 'the male ruler of an empire']	http://serbiandictionary.com/translate/цар	NOUN	Sing	/	Nom	Masc	3	nsubj	O
ga	he	on	['he', 'him']	http://serbiandictionary.com/translate/он	PRON	Sing	/	Acc	Masc	3	obj	O
primi	receive	primiti	['get something; come into possession of', 'receive a specified treatment (abstract)', 'receive willingly something given or offered', 'give an affirmative reply to; respond favorably to', 'register (perceptual input)', 'have or give a reception', 'bid welcome to; greet upon arrival', "express willingness to have in one's home or environs", 'contain or hold; have within', 'experience or feel or submit to', 'choose and follow; as of theories, ideas, policies, strategies or plans', 'take on titles, offices, duties, responsibilities', 'take hold of so as to seize or restrain or stop the motion of']	http://serbiandictionary.com/translate/примити	VERB	Sing	3	/	/	0	root	O
i	and	i	['and']	http://serbiandictionary.com/translate/и	CCONJ	/	/	/	/	5	cc	O
obeća	promise	obećati	/	/	VERB	Sing	3	/	/	3	conj	O
se	yourself	sebe	['oneself', 'itself']	http://serbiandictionary.com/translate/себе	PRON	/	/	Acc	/	5	compound	O
da	yeah	da	['yes', 'let me/us/there', 'I hope', 'that', 'in order to', 'to', 'if', 'who', 'than', 'what a']	http://serbiandictionary.com/translate/да	SCONJ	/	/	/	/	10	mark	O
će	want	hteti	['feel or have a desire for; want strongly', 'have need of']	http://serbiandictionary.com/translate/хтети	AUX	Sing	3	/	/	10	aux	O
mu	he	on	['he', 'him']	http://serbiandictionary.com/translate/он	PRON	Sing	/	Dat	Masc	10	obl	O
dati	give	dati	['cause to have, in the abstract sense or physical sense', 'transfer possession of something concrete or abstract to somebody', 'convey or reveal information', 'give as a present; make a gift of', 'be the cause or source of', 'give or supply', 'transmit (knowledge or skills)', 'bestow, especially officially', 'consent to, give permission', "yield to another's wish or opinion", 'have a beginning characterized in some specified way']	http://serbiandictionary.com/translate/дати	VERB	/	/	/	/	5	csubj	O
.	.	.	/	http://serbiandictionary.com/translate/.	PUNCT	/	/	/	/	3	punct	O

Figure 3.1.: Per-token morphological, syntactic and semantic analysis produced by the foundation web application: each word is shown with its lemma, English gloss, local and online definitions, part of speech, morphological features (number, person, case, gender), dependency head and relation, and named-entity tag. Reproduced from the author's Bachelor thesis [Cve24].

CLASSLA is additionally responsible for semantic and syntactic layers. Named entity recognition (NER) is performed using the IOB2 tagging scheme over person, location, organization, and miscellaneous categories [TL23]. One limitation observed is that date expressions were not recognized as entities by the pipeline. Dependency parsing produces a structure of Universal Dependencies relations [dMMNZ21], displayed per token as head index plus relation label. The whole input and each individual lemmatized word are translated to English via googletrans library [goo20]. It is important to note that certain words lose context in single-word translations. For example, the word *da* is always translated as an adverb, but it can also work as a functional particle. This is why sentence-level translations are provided, too. It is important to note that the googletrans library was later replaced by neural machine translation (see Table 3.1). The system also links each word to an English gloss using a local Serbian WordNet file of roughly 2,500 synset entries.[1] In addition, an online dictionary is also linked for each word and provides conjugation and declension tables, supporting non-Serbian speakers.[2] Unfortunately, even though the website is still operating, it does not provide definitions anymore. The database seems to be disconnected.

Dependency structures previously processed are rendered as interactive graphs with PyVis. Words are used as nodes and dependency relations as edges [PUL20] (Fig. 3.2).

During the testing of a nineteenth-century folk tale, a misassignment of word types and dependency relations on rare archaic items was exposed, showing early signs that the tool has a limit on data it encountered during the training, since it did not know what *stogod* ('anything') and *zacenis* ('to gasp for breath') mean. All in all, this foundation established a modular architecture and the analysis capabilities. However, it did not perform any text generation yet. Everything that follows builds upon this foundation.

3.2. First Extension: Speech Input and Discourse-Level Analysis

The goal of the first extension is to broaden the input modalities and move towards discourse-level processing. The additions are: speech input, two types of summarization, topic modelling and new visualizations for improved interpretability.

Speech input is implemented using OpenAI's Whisper model (medium), which is a Transformer-based Automatic Speech Recognition (ASR) model trained on 680,000 hours of multilingual data [RKX+23]. The users are offered an option to either record their voice directly via their device's microphone or upload an audio file. After that, the model outputs the transcribed text into the input field and the analysis begins. The evaluation of this module was done on 1,000 entries of JuzneVesti-SR v1.0 corpus [RL22], delivering a word error rate (WER) of 45.9% and a character error rate (CER) of 21.6%. CER was consistently lower than WER, indicating that the output was phonetically close but diverged at the word level. A more detailed analysis of the dominant errors showed that the model struggled to differentiate between the dialects, it repeated words, and it deleted

[1] Each row pairs a word with a WordNet synset offset and an English gloss. A `http://serbiandictionary.com/` lookup was also linked, but it is no longer functional.

[2] The conjugation and declension tables are linked from PONS, `https://en.pons.com/`.

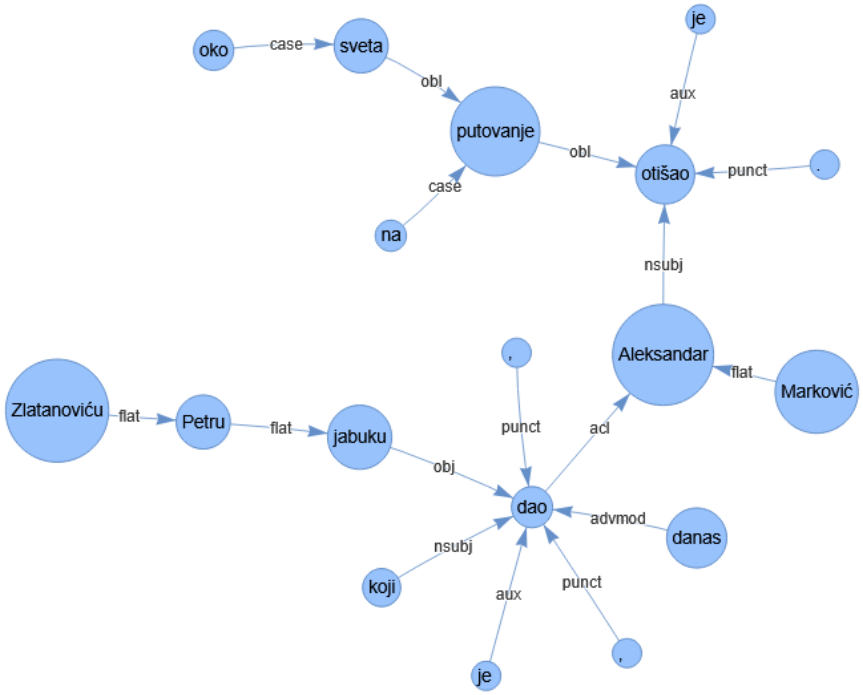

Figure 3.2.: Dependency structure of a Serbian sentence rendered as an interactive graph with PyVis [PUL20]: words are nodes and Universal Dependencies relations [dMMNZ21] are the labelled edges. Reproduced from the author's Bachelor thesis [Cve24].

short function words and proper names. This reflects Serbian's low-resource status compared to Whisper's reported performance on high-resource languages. Figure 3.3 shows the error distribution per file.

Extractive summarization shortens the input text by taking its top ranked sentences and deleting others. The input sentences are vectorized with TF-IDF [MRS08], a cosine similarity graph is built and then the sentences are ranked with PageRank [PBMW99]. They are ranked in the manner of TextRank [MT04], meaning that the top ranked sentences form the summary. This technique is language-independent and requires no annotated training data. This is a good fit for a low-resource language. However, it comes at the cost of being unable to rephrase or fuse the information, reducing the cohesion in the process. Abstractive summarization was first implemented with an mT5 model fine-tuned on the XL-Sum dataset [XCR+21, HBI+21], with the input transliterated to Cyrillic, since the model performed slightly better on that script. An initial informal evaluation of thirty snippets revealed several failure modes where the model would output hallucinated boilerplate, fabricated facts, over-generalized phrasing, and truncation for short inputs [JLF+23]. The output was fluent but unfaithful. This component was later

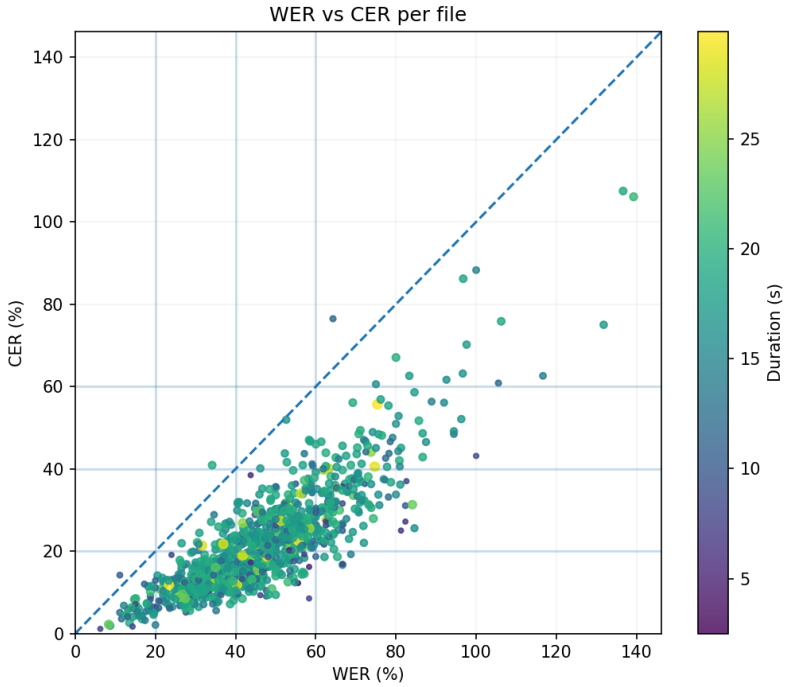

Figure 3.3.: Word error rate (WER) versus character error rate (CER) for the Whisper speech-to-text module, evaluated on 1,000 JuzneVesti-SR utterances [RL22]; each point is one file, coloured by its audio duration. Points lie below the diagonal, i.e. CER is consistently lower than WER, confirming that the transcripts are phonetically close but diverge at the word level. From the author's Praktikum 1 [Cve25].

moved onto the LLM backend (Table 3.1), so abstractive summaries are now produced by prompting the general-purpose model, which exhibits the same fidelity problems and is an early but concrete example of the general-purpose model limitation that Sect. 3.5 will cover.

Topic modelling uses Latent Dirichlet Allocation (LDA) [BNJ03] over a document-term matrix built with scikit-learn's CountVectorizer [PVG+11]. The input is first lowercased, the punctuation is stripped and a filter is applied that removes common Serbian stopwords, after which the most probable words per topic are displayed (Algorithm 1). Beyond the analytical tasks, three new visualizations were added, including a word cloud for frequencies [HLLE14] (Fig. 3.4), a heatmap of named-entity types across sentences (Fig. 3.5), and a nested donut chart relating universal to language-specific POS tags (Fig. 3.6). After all of these additions, the system accepted speech and analysed discourse; however, its new generative component, the abstractive summarizer, was an off-the-shelf model with documented problems.

Figure 3.4.: Word-cloud visualization of the most frequent tokens in an analysed Serbian text; larger words occur more often. From the author's Praktikum 1 [Cve25].

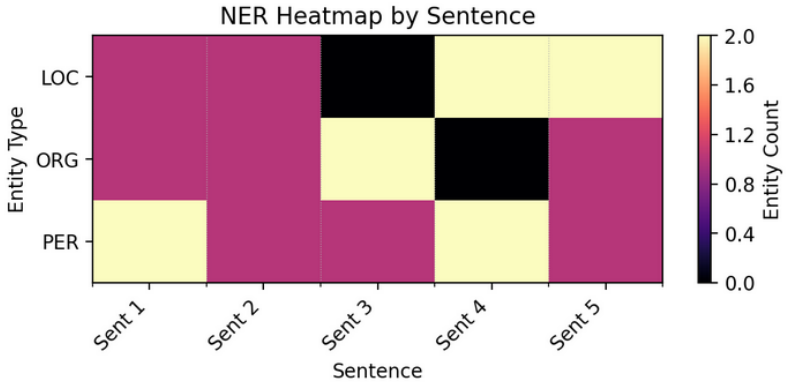

Figure 3.5.: Heatmap of named-entity types (person, organization, location) across the sentences of an analysed text; cell intensity encodes the entity count per sentence. From the author's Praktikum 1 [Cve25].

Algorithm 1: Topic modelling with Latent Dirichlet Allocation.

Input: Input text x; number of topics k; Serbian stopword list S
Output: The most probable words per topic

lowercase x and strip punctuation
remove from x every token contained in S
$DTM \leftarrow \text{CountVectorizer}(x)$ `// document-term matrix`
fit LDA with k topics over DTM
foreach *topic* $t \in \{1, \ldots, k\}$ **do**
    display the most probable words of topic t
end

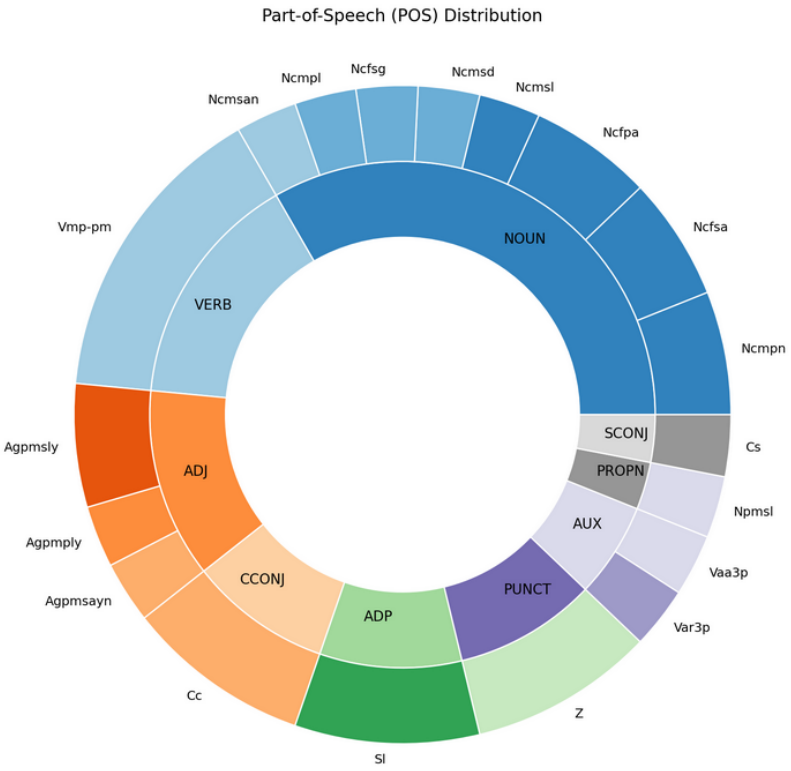

Figure 3.6.: Nested donut chart relating the universal part-of-speech tags (inner ring) to the finer language-specific CLASSLA tags (outer ring) for an analysed Serbian text. From the author's Praktikum 1 [Cve25].

3.3. Second Extension: Generative Language Tasks

The second extension's main focus was to improve the semantic and generative tasks powered by a locally deployed LLM. The model chosen for backend deployment is Llama 3.1 8B [G+24], served through the Ollama framework [Oll23] without external APIs and with acceptable performance on local hardware. The five new additions include: grammar correction, sentiment analysis, hate-speech detection, aspect-based sentiment analysis and semantic role labelling. For grammar correction, the approach is purely prompt-based. The model is instructed to act as a Serbian proofreader in the Ekavian dialect, to preserve the word order and meaning and to output only the corrected sentence if necessary. The generation is near-deterministic, meaning that a temperature near zero was used, since correction demands consistency and not creativity [HBD+20]. Tests on higher temperatures produced unstable results, while tests with temperatures set to zero produced locally optimal but not globally optimal results, hence the value of 0.05 was used in the end. Observed behaviour on this setting showed that the model is strong on spelling, agreement, verb forms and multiple errors at the same time. However, it also showed that sometimes, it does not follow the constraints mentioned in

the prompt. Precisely these failures are what task-specific fine-tuning later addresses. Sentiment analysis utilizes the pretrained twitter-XLM-RoBERTa classifier (the cardiffnlp `twitter-xlm-roberta-base-sentiment` model) [BEACC22, CKG+20] that runs locally, both on the whole input and on the sentence level. The model proved reliable on clearly polarized sentences. However, it also showed recency bias for sentences with mixed polarity inputs, taking the polarity of the last part of the sentence. Hate speech detection uses a multilingual XLM-RoBERTa hate-speech model that takes the most probable category from its per-category confidence scores, flagging the input unless that category is the appropriate (non-hateful) one, with additional keyword-based overrides for explicit violence and slurs [CKG+20, SW17]. The results presented a weakness of systematic over-flagging. Neutral or positive statements were flagged as hateful if they used keywords connected to nationality, race, religion, etc. The detection of overt discrimination remained accurate. These two tasks deliberately use classifiers instead of LLMs, a design decision the thesis later evaluates during the comparison of the fine-tuned model against the dedicated classifier. ABSA is a hybrid pipeline: CLASSLA dependency parsing extracts the candidate aspect nouns [TL23], which are then passed together to an LLM in a single prompt that assigns a polarity and confidence score to each aspect, followed by a rule-based pass that refines the polarities using Serbian sentiment cues and negation [PGP+16]. This approach ensured the correct separation of multiple aspects with conflicting sentiments. However, abstract nouns and temporal expressions occasionally get incorrectly parsed as aspects. Aspect extraction is hard in free-word-order languages (Sect. 2.1). SRL also utilizes the hybrid approach – CLASSLA produces an indexed token sequence and candidate predicates consisting of verbs and auxiliaries. The LLM is then prompted to return strict JSON frames with a role inventory of agent, patient/theme, recipient, time, location, source, destination, cause, purpose, manner, instrument and other [GJ02, PGK05]. Additional normalization rules were added in order to improve the output, where temporal adverbs are mapped to Time, causal preposition *Zbog* to Cause, and the Location is split into Source/Destination by the governing preposition. One limitation was how this system handled the word *biti*, since it sometimes yielded redundant frames. Instrumental phrases sometimes get mislabelled as patient/theme and the role spans can extend beyond their correct boundaries. Algorithm 2 and Algorithm 3 summarize these two hybrid pipelines. By this stage, the web application utilized analysis and generation across many tasks, but the generative capabilities depended on prompting a general-purpose model whose Serbian competence does not always deliver highly dependable and accurate results, which is the situation that Sect. 3.5 diagnoses.

Algorithm 2: Aspect-based sentiment analysis (CLASSLA + LLM hybrid).

Input: Input sentence x; instruction-following LLM
Output: Aspect–polarity pairs with confidence scores
parse x with the CLASSLA dependency parser
$A \leftarrow$ candidate aspect nouns extracted from the dependency parse
prompt the LLM once with (A, x) to assign a polarity and confidence score to every aspect
refine the polarities with a rule-based pass (Serbian sentiment cues and negation)
return the aspects with their polarities `// conflicting aspects stay separate`

Algorithm 3: Semantic role labelling (CLASSLA + LLM hybrid) with role normalization.

Input: Input sentence x; LLM; role inventory $R = \{$agent, patient/theme, recipient, time, location, source, destination, cause, purpose, manner, instrument, other$\}$
Output: Normalized semantic-role frames
$(T, P) \leftarrow$ CLASSLA: indexed tokens T, candidate predicates P (verbs, auxiliaries)
foreach *predicate* $p \in P$ **do**
  prompt the LLM to return a strict JSON frame for p using roles from R
end
`/* Normalization rules applied to the returned frames */`
map temporal adverbs to Time
map the causal preposition *Zbog* to Cause
split Location into Source / Destination by the governing preposition
return the normalized frames

3.4. Architecture of the Unified Web Application

This section consolidates the system as it stood at the start of the thesis work. It is a Flask application consisting of multiple separate modules organized into a single input pipeline (Fig. 3.7). The data flow is as follows: the user inputs text, uploads an audio file, or records their voice directly, which is then transliterated and preprocessed with the CLASSLA pipeline. Then the selected tasks are run, the results are aggregated and their visualizations are processed and shown on the screen.

The app contains three primary component types responsible for the core analysis and generative tasks, alongside supporting utilities such as speech input and transliteration.

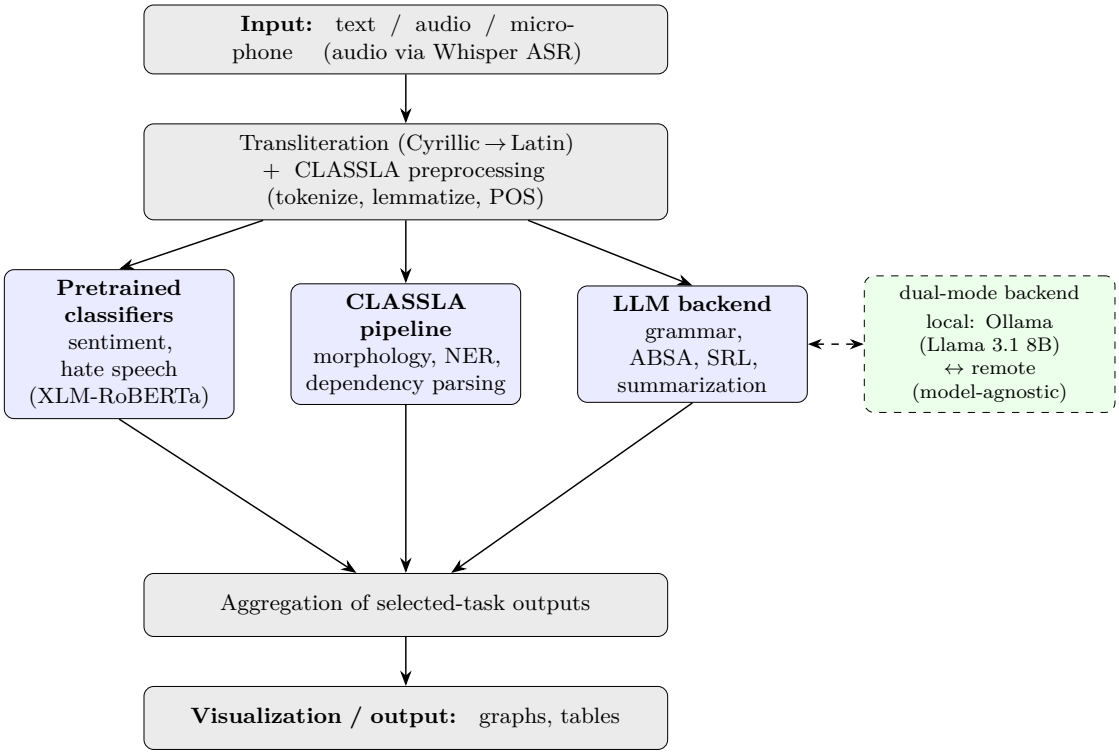

Figure 3.7.: Architecture of the unified web application. Input (text, audio, or microphone, the latter two via Whisper ASR) is transliterated (Cyrillic→Latin) and preprocessed with CLASSLA, then dispatched to three component types (the CLASSLA linguistic pipeline, pretrained XLM-RoBERTa classifiers, and a generative LLM backend) whose outputs are aggregated and visualized. The backend is model-agnostic through a dual-mode local/remote swap: the general-purpose Llama 3.1 8B (served locally via Ollama) marks the seam where the fine-tuned model of Chapter 4 is later plugged in.

The first one is CLASSLA, a pipeline for linguistic analysis, the second one is a dedicated pretrained classifier for sentiment and hate speech, and the third one is the LLM backend for generative tasks. It is important to note that several earlier components were progressively moved onto the LLM backend, since they proved more reliable.

The backend offers a dual-mode: the user can utilize both local and remote processing of the input, which can be switched in the options tab of the app. Its purpose is to offer a solution for decoupling the app from any specific model, which is what precisely allows the fine-tuned model to be plugged in without any architectural change. All in all, the architecture is model-agnostic, but the output quality is not and is bounded by the model's competence.

Table 3.1.: Tasks in the unified web application mapped to their serving component/model and processing type, as of the consolidated architecture described in this section (the system at the start of the thesis work). Translation and abstractive summarization are shown transitioning to the neural and LLM backends (from `googletrans` and mT5, respectively).

Task	Component / model	Processing type
Tokenization, lemmatization, POS	CLASSLA	Linguistic pipeline
Named-entity recognition	CLASSLA	Linguistic pipeline
Dependency parsing	CLASSLA	Linguistic pipeline
Speech-to-text	Whisper (medium)	Neural ASR
Translation	`googletrans` → neural MT	Machine translation
Extractive summarization	TF-IDF + PageRank	Graph-based (TextRank)
Abstractive summarization	mT5 (XL-Sum) → LLM backend	Generative
Topic modelling	LDA (scikit-learn)	Statistical
Grammar correction	LLM backend (Llama 3.1 8B)	Generative (LLM)
Sentiment analysis	twitter-XLM-RoBERTa	Pretrained classifier
Hate-speech detection	multilingual XLM-RoBERTa	Pretrained classifier
Aspect-based sentiment (ABSA)	CLASSLA + LLM	Hybrid
Semantic role labelling (SRL)	CLASSLA + LLM	Hybrid

3.5. The Motivating Limitation: Dependence on General-Purpose Models

Every generative task in the application depends on a general-purpose instruction-tuned model with no Serbian specialization. As observed in Sect. 3.2 and Sect. 3.3, the consequence of this was unfaithful abstractive summaries, grammar correction with unintended meaning, gender or dialect changes, violation of explicit rules in the given prompts, and aspect and role errors connected to the model's limited knowledge of Serbian. These observations match the documented pattern, which is the fact that predominantly English-trained models underperform on morphologically rich, low-resource languages [JSB+20, AU25]. The constraints can be stated in the prompt but cannot be enforced, meaning that prompting alone cannot solve this problem reliably. The underlying Serbian competence is fixed at what the pretraining provided. The alternative to this problem is the adaptation of open-weight models, which can be fine-tuned on Serbian multi-task data in order to specialize them while retaining local deployment and data control [WBZ+22, HSW+22, Y+25]. Thanks to the dual-mode of the application, a specialized model can replace the general-purpose one without any system changes. Addressing this limitation is the central contribution of the thesis, which aims to construct Serbian multi-task training data, fully fine-tune an open-weight model and evaluate the results with automatic benchmarks and native speakers, covered in the next chapter.

4. Fine-Tuning a Specialized Model

The purpose of this chapter is to describe the construction of the dedicated Serbian LLM that addresses the limitations mentioned in Chapter 3. The scope includes the reasoning behind the choice of the base model, construction of the training data, progression from parameter-efficient to full fine-tuning, the infrastructure and training settings, required engineering and the deployment inside the web application. The central commitment is to specialize an open-weight model for Serbian multi-task use while having a locally deployable pipeline.

4.1. Model Selection and Rationale

The backbone of the system had to be an open-weight model, not a closed API model. This allows for full control over the training data and process, reproducibility and the ability for local deployment without sending Serbian text to remote services [AU25]. This decision rules out the strongest closed models and points towards open instruction-tuned models.

A strong fit for the use case and the available local hardware is Qwen3, which is an open instruction-tuned model with a permissive licence and parameters spanning from 0.6B to 235B [Y+25]. One of the deciding properties of the model is strong multilingual pretraining, which covers 119 languages and dialects, a significant upgrade over its predecessors, and a unified instruction-following design [Y+25]. As an instruction-tuned model, OpenPipe/Qwen3-14B-Instruct[1] already follows the prompt-based approach for given tasks, hence fine-tuning only specializes the existing behaviours rather than teaching the whole output format from scratch.

During the fine-tuning process, two models were used: a small 4B model and a larger 14B model, both dense variants. First, a smaller model was used to run the benchmarks and compute the results as a prototype phase. Afterwards the model was fine-tuned multiple times and the results were compared until a reasonable improvement was achieved. The reason behind this is that smaller models are faster to train. This also allowed testing whether a model so small would be good enough for advanced tasks. The 14B fine-tuned model was selected in the end because of its overall strength, while the 4B served only as a fast exploratory prototype. The per-task base-versus-fine-tuned comparison for the 14B model is reported in Chapter 5.

[1] A fine-tune-friendly instruction variant of Qwen3-14B with a non-thinking-by-default chat template: `https://huggingface.co/OpenPipe/Qwen3-14B-Instruct`.

4.2. Training Data Construction

The goal of the construction of the training data is to assemble a single instruction-formatted dataset covering all generative tasks the model must perform. 24,601 examples across all five tasks were made: summarization with 6,000, grammar correction with 6,000, translation with 5,000, ABSA with 4,000 and sentiment with 3,601. As decided earlier, hate-speech detection and semantic role labelling were excluded from the scope and from the data construction process. The reason behind this is the fact that the existing classifiers already do an adequate job.

Every example uses a unified format, which consists of an instruction-style prompt-response pair defined in such a manner that the model learns to deal with different tasks through differing instructions rather than task-specific heads [WBZ+22]. The reasoning behind this usage of pairs was to mirror the prompt templates inside the web application, such that the training and deployment match.

The examples used in the datasets came from existing Serbian corpora, rule-based injection and from distillation by a stronger teacher model[2] Qwen2.5-72B-Instruct-AWQ, which was then converted into unified instruction format. This approach can be considered as knowledge distillation and synthetic instruction generation [HVD15, WKM+23]. For the translation set, OPUS-100 pairs [ZWTS20] were combined with teacher-distilled translations of Serbian Wikipedia text. Sentiment consisted of SentiComments.SR [BCN20] and ParlaSent (BCS, i.e. Bosnian–Croatian–Serbian) [MRL24]. Summarization set consisted of Serbian Wikipedia entries and used the article lead as the summary for the body. Grammar correction set utilized rule-based error injection over the summarization text and included synthetic errors like case-ending swaps, diacritic stripping and punctuation (the full rule set is given in Appendix A.2). ABSA used SentiComments.SR comments and synthetic reviews annotated by the teacher model. The data was additionally processed with Python scripts to remove duplicates, validate the format, and create a held-out split used for testing. All evaluation and user-study data was kept separate from this training data. Each task used multiple prompt phrasings in both Latin and Cyrillic, in order to reinforce the model against the digraphia problem. The full pipeline and the per-task composition are shown in Fig. 4.1 and Table 4.1. Appendix A.6 gives the same composition with per-task sources.

4.3. From LoRA to Full Fine-Tuning

The distinction between full fine-tuning and LoRA is the area of effect over the weights. LoRA freezes the base weights and trains small low-rank adapters, which is cheap, fast and delivers the results [HSW+22]. Full fine-tuning updates all the weights, which is slower and requires more compute, but delivers better results. The work began with LoRA, because its memory efficiency ensured fast testing of the training data. As soon as satisfactory results were achieved, the work was moved to full fine-tuning.

[2] A larger, stronger model (the *teacher*) generates the training data that is then used to train the smaller target model.

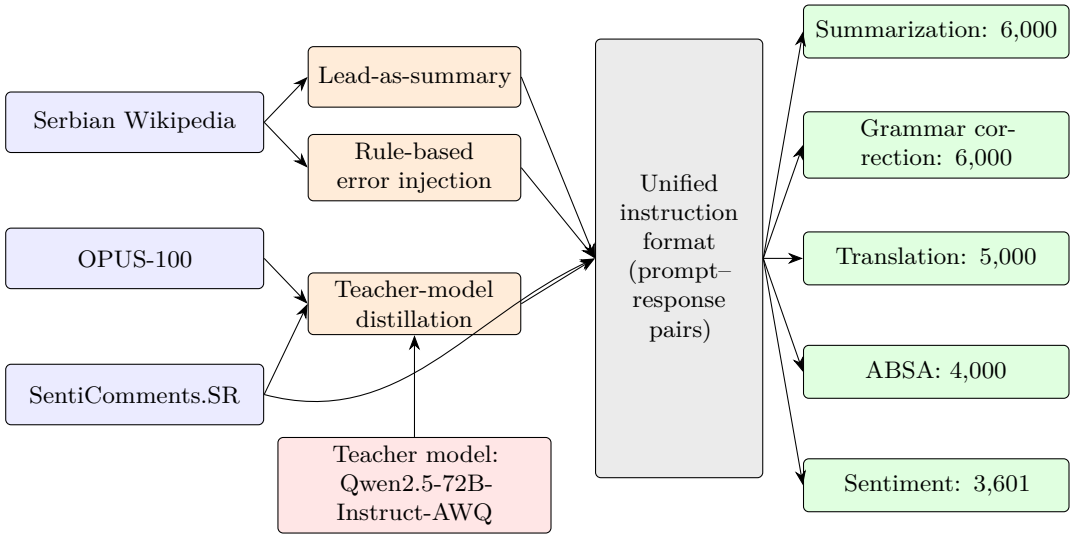

Figure 4.1.: Construction of the instruction-tuning dataset. Source corpora are processed per task (lead-as-summary, rule-based error injection, and teacher-model distillation with Qwen2.5-72B-Instruct-AWQ) then converted to a unified instruction (prompt–response) format, yielding five task datasets totalling 24,601 examples. The diagram is schematic and shows the primary sources only (ParlaSent, which also feeds the sentiment set, is omitted for clarity), and the translation set combines OPUS-100 pairs with teacher-distilled Serbian Wikipedia text.

Table 4.1.: Composition of the instruction-formatted fine-tuning dataset: per-task example counts, share of the total, and primary data source. Synthetic examples were distilled from the Qwen2.5-72B-Instruct-AWQ teacher model.

Task	Examples	Share (%)	Primary source / construction
Summarization	6,000	24.4	Serbian Wikipedia (lead as summary)
Grammar correction	6,000	24.4	Rule-based error injection
Translation	5,000	20.3	OPUS-100 + teacher distillation
ABSA	4,000	16.3	SentiComments.SR + synthetic reviews
Sentiment	3,601	14.6	SentiComments.SR + ParlaSent (BCS)
Total	**24,601**	**100.0**	

The initial phase used LoRA for the prototype task adaptation, and trained the adapters on general capabilities and individual tasks. The config utilized rank r=32, $\alpha = 64$, dropout=0.05 and targeted all seven projection matrices: q, k, v, o, gate, up and down. LoRA trained quickly and cheaply, but the multi-task output was of a lesser quality than desired, motivating the move to the full fine-tune [BPGO+24].

The platform used for LoRA and full fine-tuning was University of Vienna's NVIDIA DGX Spark desktop supercomputer, featuring a GB10 Grace Blackwell Superchip with 128 GB of unified memory on CUDA 13.1. This provided enough headroom to fully fine-tune the 14B model without sharding. The final multi-task model was produced by full fine-tuning, which gave stronger and more consistent performance across all five tasks. However, the LoRA results are also worth mentioning: the LoRA-tuned 4B model reached a sentiment accuracy of 73% on ParlaSent, compared to the cardiffnlp classifier's 69.8%, and this trade-off is discussed further in Chapter 7. This phase sets an empirical basis for both the full fine-tuning and the deployment choices.

4.4. Training Infrastructure

As mentioned in the previous section, the core of the training process was the NVIDIA DGX Spark system, which allowed for full fine-tuning of a 14B model on a single device without parallelism or offloading.

The framework used is Hugging Face Transformers Trainer [WDS+20], along with a model loaded in bfloat16 with SDPA attention (`attn_implementation="sdpa"`) and `use_cache=False`. Each example was built via the tokenizer's chat template, tokenized to a fixed max length, with the prompt portion masked to -100, so that the loss is computed only over the assistant's response.

4.5. Training Methodology

The hyperparameters used in the configuration are the following: max sequence length 2048, per-device batch size 1, gradient accumulation 16, learning rate 1e-5, cosine schedule with warmup ratio 0.03, max grad norm 1.0, Adafactor optimizer, 1 epoch and gradient checkpointing enabled [SS18] (the full configuration, alongside the exploratory LoRA run, is in Appendix A.3). Gradient checkpointing was chosen in order to save memory at the cost of reducing the speed of training. Adafactor was chosen over Adam/AdamW [KB15, LH19] because of its training stability and sublinear optimizer-state memory.

Full fine-tuning took approximately 21.3 hours over 1 epoch, reaching the epoch-mean training loss of 0.3536 (Fig. 4.2). The run completed without activating the watchdog, which automatically kills the process if it detects VRAM usage over 115 GB. This produced the final fine-tuned model used in deployment and evaluation. The complete procedure, including the supervised-target construction with prompt-token masking and the concurrent watchdog that guards the shared device, is summarized in Algorithm 4.

Algorithm 4: Memory-safe full fine-tuning of the 14B model on a single DGX Spark device.

Input: Instruction dataset $D = \{(p_i, r_i)\}_{i=1}^{24{,}601}$; base model M (Qwen3-14B-Instruct); tokenizer with chat template
Output: Fine-tuned model $M^\star$

Procedure `TrainSafely()`
  start `Watchdog()` in a separate thread
  if `Watchdog()` *reports more than* 15 *GB already in use* **then abort**
  Load M in bfloat16; SDPA attention; `use_cache`=False; checkpointing on
  `/* Build supervised targets: learn only the assistant's response */`
  foreach $(p_i, r_i) \in D$ **do**
    $t_i \leftarrow$ tokenize(chat_template(p_i, r_i)), truncated to $L = 2048$
    set every prompt-span token of t_i to -100 `// mask the prompt`
  end
  `/* One epoch; Adafactor; cosine schedule, warmup` 0.03; `LR` 10^{-5} `*/`
  for $s \leftarrow 1$ **to** 1,538 **do**
    accumulate the loss over $K = 16$ micro-batches of size 1
    clip the gradient norm to 1.0
    Adafactor update of *all* weights; advance the cosine schedule
  end
  return $M^\star \leftarrow M$
Procedure `Watchdog()`
  if *more than* 15 *GB of GPU memory is already in use* **then** refuse startup
  while *the run is alive* **do**
    poll GPU memory every few seconds
    if *memory use* $\geq$ 115 *GB* **or** *no step finished in* 600 *s* **then** kill the run
  end

4.6. Challenges and Failures

The earlier attempts at fine-tuning were unstable. The initial configuration using 8-bit AdamW ran into an insufficient-memory problem and crashed the whole platform. This is the reason the switch to Adafactor happened [SS18], because it utilizes sublinear optimizer-state memory, which lowers the memory pressure.

An additional safety mechanism was made: a watchdog. The watchdog is essentially an additional separate thread that polls GPU memory every few seconds and kills the main process instantly, if VRAM reaches 115 GB. It also kills the process if no step completes

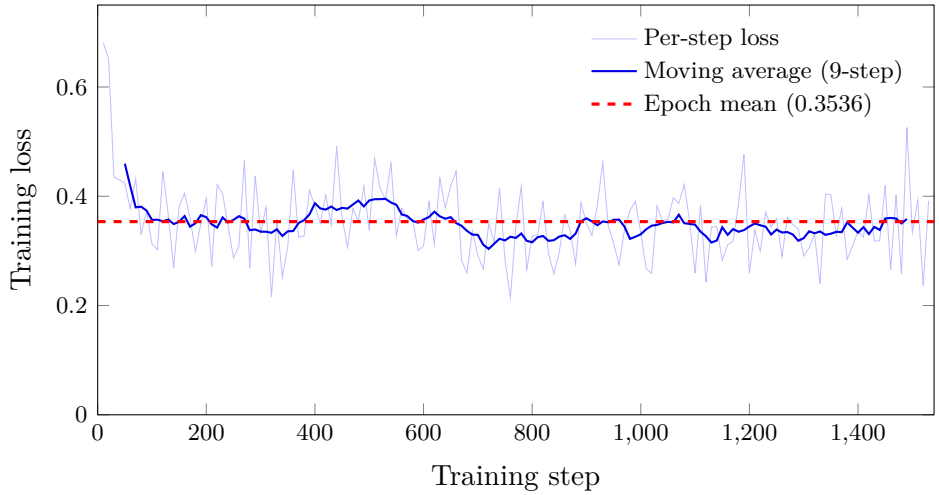

Figure 4.2.: Training loss for the final 14B full fine-tune (1 epoch, 1,538 steps, loss logged every 10 steps, 153 points). The bold line is a centred nine-point moving average of the same series. The dashed line marks the trainer-reported epoch-mean training loss of 0.3536.

in 600 seconds and ensures that the process does not start if more than 15 GB of memory is already in use. This mattered, since a full one epoch run took more than 21 hours and letting it die after 20 hours is expensive. Blocking of other people during their training is also not an option, since if an out-of-memory (OOM) crash happens, there is a big chance the machine will not be able to recover without a physical intervention.

Before a multi-hour run, multiple smoke tests were executed, whereby small slices of data were used to confirm stability end-to-end. Memory levers used were the following: batch 1 and gradient accumulation, bfloat16, gradient checkpointing, bounded sequence length. This setup ensured the full fine-tune could happen at the cost of training speed. It also made the full fine-tuning on a single device practical.

4.7. Deployment

The model is served using vLLM with paged attention [KLZ+23] exposed under the name `fulltune_14b_v3` in bfloat16, max model length 4096 and `-enforce-eager`. In addition, the model runs under a persistent restart wrapper, which is a while loop that relaunches the vLLM 30 seconds after any exit. This setup ensures no transient failures take down the endpoint.

The model is exposed through an OpenAI-compatible endpoint and consumed by the remote mode of the application's dual-mode backend. The web application reaches the endpoint through a tunnelled connection to the serving host. Importantly, this remote host is a self-hosted server (the University's DGX Spark) rather than a third-party API, so the on-premise data-control rationale is preserved. The remote backend can be

selected in settings. By using this setup, one can run stronger models when the local hardware is limiting. In this case the fine-tuned model was served remotely to speed up the comparison process, since the local RTX 3060 with 6 GB of VRAM is slow to produce outputs.

The deployed generative model serves the following tasks: grammar correction, translation, ABSA, and abstractive summarization (the inference prompt templates are listed in Appendix A.1). Although the model was fine-tuned on sentiment, the deployed application uses the dedicated XLM-RoBERTa classifier for sentiment, since it is very lightweight and fast. The fine-tuned model is more accurate, though it takes more time to produce an output, which is why it is a selectable option. With the model built, trained and deployed, the next chapter evaluates it against its base model on automatic benchmarks.

5. Benchmark Evaluation

The purpose of the benchmark is to evaluate the fine-tuned model against its base model. The evaluation was run on generative tasks and on a broad Serbian benchmark. There are two evaluation tracks, each with a separate methodology observation about how instruction-tuned models are scored. Generative and sentiment evaluation sets were checked for overlap with the training data and decontaminated where necessary (see Appendix A.4), so that the end results are not inflated.

5.1. Evaluation Setup

The comparison used two versions of the same model: one fine-tuned and one the base model. The model in question is OpenPipe/Qwen3-14B-Instruct. All the differences in performance are attributed to the fine-tuning alone. The 4B base and fine-tuned models served only as the fast exploratory prototype described in Chapter 4 and are not analysed further here, where the focus is the 14B model.

The five tasks with their datasets and metrics are the following: summarization with 300 items and ROUGE-L [Lin04], sentiment on ParlaSent [MRL24] (600 entries) and SerbMR (the Serbian Movie Review dataset [BNM16], 168 entries) tested on accuracy, translation with 200 entries tested on BLEU [PRWZ02] (chrF [Pop15] was also computed), grammar correction with 200 entries tested on exact matches, and finally ABSA with 200 entries tested on aspect-F1 [PGP+16]. The outputs were produced via a vLLM OpenAI-compatible endpoint and temperature 0 (greedy)[1], with <think> reasoning blocks stripped before the scoring process started.

The benchmark suite utilized the already existing datatab/serbian-llm-benchmark [dat24], which includes BoolQ, ARC (easy and challenge), PIQA, HellaSwag, Winogrande, OpenBookQA, OZ-eval, and ∼40 MMLU-Serbian subjects [HBB+21], amounting to approximately 29,000 multiple-choice and boolean questions in total. Scoring was done by chat-based generation, where the model is prompted to output a single answer (a letter, or a yes/no for BoolQ), which is contrasted in Sect. 5.3.

5.2. Generative Task Results

The fine-tuned model improved across all five tasks it was trained on (Tables 5.1 and 5.2).

Out of all of them, grammar correction had the biggest gains, going from 0.000 to 0.455. The zero in the base model occurred because it always paraphrased or added

[1] Deterministic decoding: at each generation step the model always selects the single most probable next token, making the output reproducible.

commentary instead of returning only the corrected sentence, which is why an exact match failed, even when the correction was reasonable. The usage of additional rules in the prompts did not help. Fine-tuning taught the model how to output the desired format, which directly validates the grammar limitation described in Sect. 3.3. Regarding ABSA, aspect-F1 improved from 0.433 to 0.622, while the sentiment labelling accuracy was near 1 for both models, so the entire gain was in aspect extraction, where precision went from 0.37 to 0.57 and recall from 0.53 to 0.68. This means that the model improved in finding aspects, not in assigning polarity.

Sentiment also improved, with ParlaSent going from 74.2% to 82.7% and SerbMR from 74.4% to 76.8% (Table 5.2). The same table also reports the off-the-shelf cardiffnlp `twitter-xlm-roberta-base-sentiment` classifier [BEACC22] as a baseline: it is reasonable on ParlaSent but collapses on the out-of-domain SerbMR reviews, scoring only 32.7% accuracy, which the fine-tuned model handles far better. This collapse can be seen clearly in the per-class breakdown in Fig. 5.1. On SerbMR the classifier puts almost every review into the *neutral* class and never predicts *positive*, while both LLM versions are much closer to the correct labels. Summarization was a unique case. ROUGE-L improved from 0.072 to 0.120, a small increase that is still low in absolute terms. It is important to note that ROUGE-L only rewards overlap with the gold summary and does not measure faithfulness or completeness, so a low score does not by itself show whether the summary is good or bad. This is examined further in the user study in Chapter 6. Translation remained almost flat, with BLEU going from 61.54 to 61.75 and chrF from 79.3 to 78.9. This outcome is better known as a ceiling effect, meaning that the base model is already strong, leaving almost no room for improvement.

Table 5.1.: Generative task results: base vs. fine-tuned Qwen3-14B-Instruct. Δ is fine-tuned minus base, computed from unrounded scores. n is the number of evaluation items.

Task	Metric	Base	Fine-tuned	Δ	n
Grammar correction	Exact match	0.000	0.455	+0.455	200
ABSA	Aspect-F1	0.433	0.622	+0.189	200
Summarization	ROUGE-L	0.072	0.120	+0.047	300
Translation	BLEU (0–100)	61.54	61.75	+0.21	200

5.3. The Evaluation Methodology Finding

Instruction-tuned models can be scored by their log-probability or by chat-based generation. Log-probability is calculated by comparing the model's assigned likelihood of each candidate answer, without generating any text. Chat-based generation is calculated by parsing the generated answer and comparing it to the reference. This thesis evaluated the same base model both ways on the same Serbian benchmark, and the outcome presented a systematic discrepancy.

Table 5.2.: Three-way sentiment comparison on the decontaminated test sets (percentages). ParlaSent $n = 600$, SerbMR $n = 168$.

	ParlaSent		SerbMR	
Model	Acc.	Macro-F1	Acc.	Macro-F1
cardiffnlp	69.8	69.9	32.7	19.6
Qwen3-14B-Instruct base	74.2	74.2	74.4	67.5
Qwen3-14B-Instruct fine-tuned	82.7	82.7	76.8	75.9

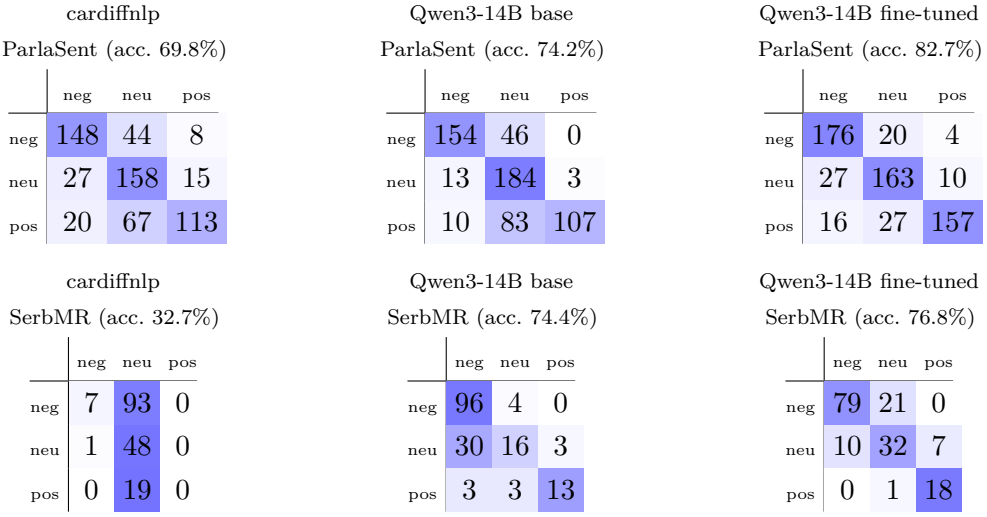

Figure 5.1.: Sentiment confusion matrices (rows are the true label, columns the predicted label, and the shading is row-normalized) on ParlaSent (top, $n = 600$) and SerbMR (bottom, $n = 168$) for the cardiffnlp classifier, the base model and the fine-tuned model. On the out-of-domain SerbMR reviews the cardiffnlp classifier puts almost everything into the *neutral* class and never predicts *positive*, which is the reason for its 32.7% accuracy. Both LLM versions stay close to the diagonal, and the fine-tuned model is the most balanced one.

Under log-probability scoring, the base model Qwen3-14B-Instruct scored 34.3% overall on the Serbian benchmark suite, and under chat-based generation it achieved 75.6%, a difference of more than 40%.

The reason behind the difference between the two tests is how the outputs are penalized. Log-probability penalizes instruction-tuned models because they spread probability mass across many fluent phrasings of an answer, rather than concentrating it on the bare answer token [HWS+21]. Chat-based scoring matches only how the model was trained to respond. Under log-probability scoring, binary tasks from BoolQ achieved good results of 69.9%. However, this did not carry over to the multiple-choice questions, where many subjects like abstract algebra scored below 30%. The model does answer correctly when

allowed to generate, which means that the problem is a signature of a scoring artefact rather than a knowledge gap.

LLM leaderboards for low-resource languages like Serbian often rely on log-probability harnesses [BSS+24], which may systematically and severely understate instruction-tuned models. The scoring method must therefore be reported alongside any benchmark score. This discrepancy, and not any single accuracy figure, is the central evaluation finding of this chapter, and it is the main reason the knowledge benchmark is reported at all, since its role here is to expose how strongly the scoring method shapes the result rather than to claim a higher score.

5.4. Knowledge Benchmark Results

This section asks a different question from the generative evaluation. It does not ask whether the model improved on the tasks it was trained on, but whether that task-specific fine-tuning changed its general knowledge. Both models were scored on the full benchmark suite of approximately 29,000 questions using the chat-based generation protocol established in Sect. 5.3, so the two columns are directly comparable. The fine-tuned 14B model reached 74.8%, almost identical to the base model's 75.6%. The benchmark is reported here precisely to establish this result. Fine-tuning on the five generative tasks left the model's broad factual and reasoning ability essentially unchanged. This is the desired outcome rather than a shortcoming, because it shows that specializing the model on Serbian generative tasks did not come at the cost of its general knowledge. The gains reported in Sect. 5.2 are therefore task-local, and they were not paid for by forgetting elsewhere. The fine-tuning data contained no benchmark-style multiple-choice questions, so there was no factual content for the model to learn beyond what the base model already encoded.

The per-category results in Table 5.3[2] make this constancy concrete. The two columns come from one matched re-evaluation on the identical items, and across all 48 categories they track each other closely, with small differences in both directions and no systematic movement. This rules out both a hidden broad gain and a hidden collapse in any single subject. Both models remain weakest on the same low-scoring categories, such as moral scenarios and global facts, a weakness common to LLMs in general. The per-category comparison is visualized in Figures 5.2 and 5.3.

5.5. Qualitative Error Analysis

The scores on their own do not show what the outputs of the two models actually look like. The examples in Appendix A.8 make this clear. This section goes through the main failure mode of each task.

[2]The high-school physics and high-school psychology subsets return identical per-category results within each model (92/150 base, 94/150 fine-tuned) and are the only two subjects with exactly 150 items. This appears to be a duplication in the upstream `datatab/serbian-llm-benchmark` dataset rather than an artefact of the evaluation, and the values are reported as released.

Table 5.3.: Base model `OpenPipe/Qwen3-14B-Instruct` versus the fine-tuned model: per-category accuracy (%) on the full Serbian LLM benchmark suite (48 categories, 28,986 questions). Both models are scored with the identical chat-based protocol (greedy decoding, single-letter parsing for multiple-choice, Da/Ne for boolean).

Category	Base	Fine-tuned	Category	Base	Fine-tuned
BoolQ	86.7	87.5	High school chemistry	72.4	71.9
ARC Challenge	87.1	86.6	High school computer science	80.2	83.3
ARC Easy	94.4	94.7	High school european history	76.2	73.2
PIQA	79.1	77.6	High school geography	81.1	81.6
Winogrande	62.6	63.4	High school mathematics	52.6	55.6
HellaSwag	72.4	70.4	High school microeconomics	75.3	75.7
OpenBookQA	77.3	77.7	High school physics	61.3	62.7
OZ-eval	80.1	80.3	High school psychology	61.3	62.7
Abstract algebra	62.0	55.0	High school statistics	69.3	68.3
Anatomija	68.7	63.4	High school world history	76.4	75.1
Astronomija	86.8	86.2	Human aging	67.0	64.2
College biology	82.4	82.4	Kliničko znanje	75.1	73.2
College chemistry	54.0	53.0	Logičke zablude	73.8	72.4
College computer science	59.6	66.7	Machine learning	51.9	54.7
College mathematics	56.0	55.0	Management	75.8	74.7
College medicine	74.6	72.8	Marketing	82.7	85.0
College physics	66.7	66.7	Miscellaneous	76.7	75.6
Computer security	79.6	77.6	Moral disputes	68.0	68.0
Conceptual physics	77.0	76.2	Moral scenarios	41.8	38.6
Econometrics	50.0	52.6	Osnovna matematika	74.8	77.7
Electrical engineering	64.6	68.8	Philosophy	70.1	69.1
Formalna logika	58.1	58.9	Poslovna etika	71.0	71.0
Global facts	40.0	42.0	Sociology	78.7	78.1
High school biology	87.9	86.9	World religions	77.8	77.2
Overall	**75.6**	**74.8**		all 28,986 questions	

For grammar correction, the main problem of the base model is the format, and not the language itself. It almost always adds an explanation around the corrected sentence („Ispravljena rečenica glasi... Objašnjenje... "), so the exact-match score fails even when the correction is actually right. This is the reason the base model scored 0.000 (Sect. 5.2). The fine-tuned model, on the other hand, returns only the corrected sentence. Its own mistakes are of a different type. Sometimes it under-corrects and leaves one of several injected errors in place, for example, when it restores the diacritic in one word but leaves another word uncorrected, and sometimes it even over-corrects.

For translation, the two models are very close, which matches the almost flat BLEU and chrF. Both of them produce correct and fluent English, and the differences are only in the choice of words or style, for example, when they translate an institution's name differently. This is what a ceiling effect looks like, since the base model is already strong.

For summarization, the fine-tuned model fails through over-compression and, in the worst cases, through hallucination. Besides the S01 example (Fig. 6.4), there is a case

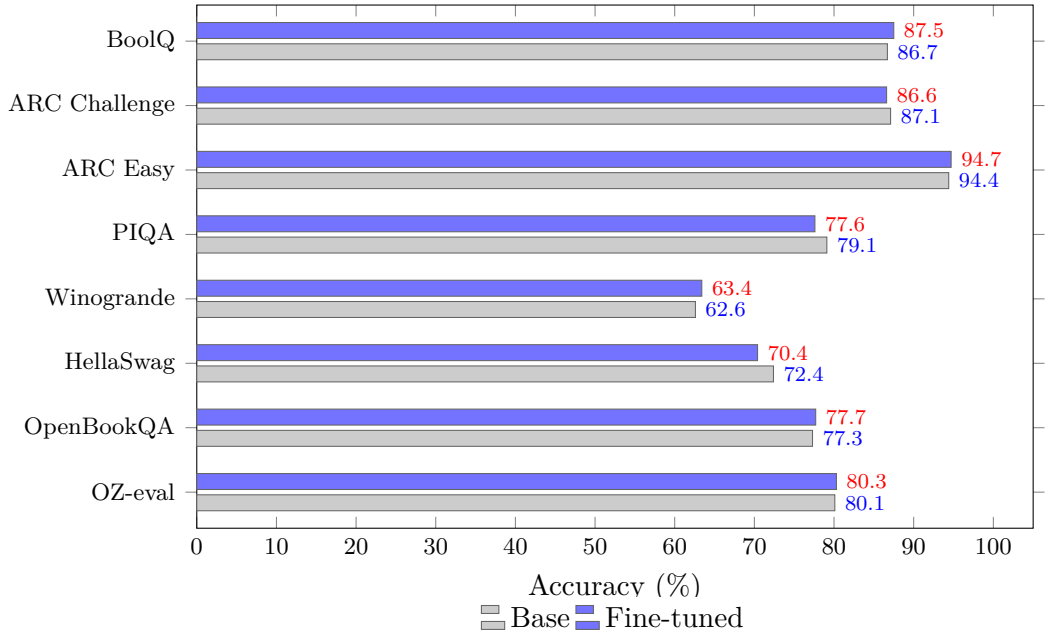

Figure 5.2.: Per-category accuracy on the eight core benchmarks: base (`OpenPipe/Qwen3-14B-Instruct`) versus fine-tuned. Visual companion to Table 5.3.

in the evaluation set where the fine-tuned model reports a 34-year-old trainer setting a plank record, while the source and the reference talk about a 62-year-old former marine. The base summary is longer, but it keeps the facts. This connects to the length difference (Fig. 6.3) and to the strong human preference for the base summaries (Chapter 6).

For sentiment, the fine-tuned model mostly improves on difficult reviews where the base model predicts negative too often. On SerbMR, for items whose gold label is neutral, the base model tends to answer negative, while the fine-tuned model gives the neutral label (Appendix A.8). This is consistent with its higher accuracy and with the confusion matrices in Fig. 5.1.

Finally, on the knowledge benchmark, the errors are basically the same for both models. They are both weakest on the same reasoning-heavy categories, such as moral scenarios and abstract algebra (Table 5.3). This confirms that the fine-tuning changed the output format and the Serbian-specific behaviour, and not the general reasoning.

5.6. Discussion

Fine-tuning significantly improved the generative tasks, especially the format-constrained ones, like grammar correction. It also improved sentiment on both domains, while leaving the general-knowledge benchmark essentially unchanged. Tasks that need specific output formats or Serbian-specific behaviour gained the most from fine-tuning, whereas broad

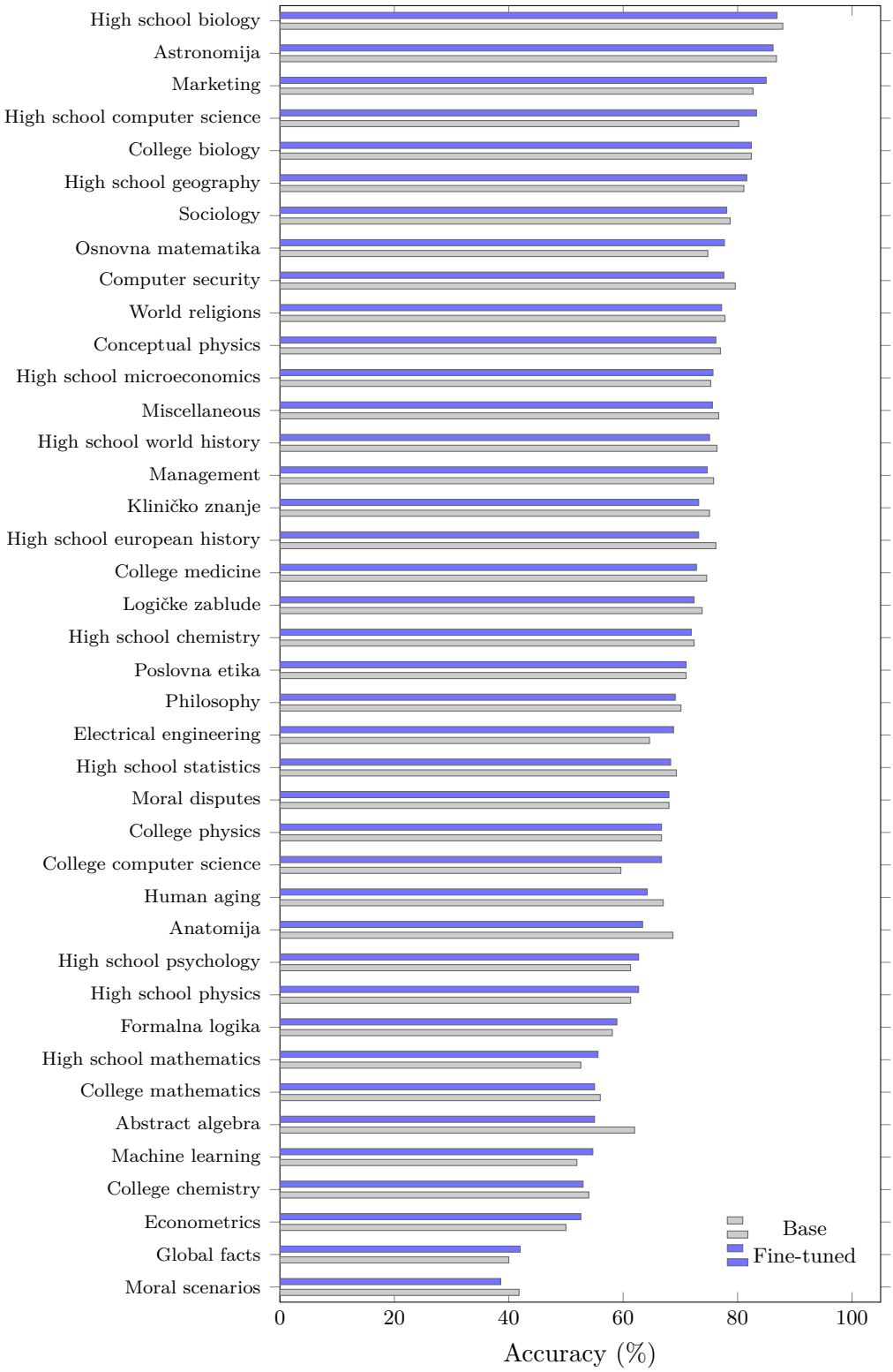

Figure 5.3.: Per-category accuracy on the 40 MMLU-Serbian subjects, sorted by fine-tuned accuracy: base (`OpenPipe/Qwen3-14B-Instruct`) versus fine-tuned. Visual companion to Table 5.3.

factual knowledge, which the fine-tuning data did not target, stayed at the base model's level. A methodological finding showed that automatic metrics must be read carefully, because metrics that reward surface overlap or exact matching, like ROUGE-L and multiple-choice parsing, as well as the off-the-shelf sentiment classifier, can collapse. This shows that performance depends heavily on format and domain, not underlying competence. Specifically for sentiment, the fine-tuned model achieved the highest accuracy and highest robustness in the domain, motivating the addition of a selection option in the app, alongside the existing fast classifier. Automatic metrics cannot fully evaluate the quality of the output, which is the reason behind the user study in Chapter 6.

6. User Study

6.1. Motivation

The automatic metrics mentioned in Chapter 5, like ROUGE, BLEU and exact match, are not sufficient on their own, since they capture surface overlap and not whether the output is natural or useful to the native speaker. In order to complement this automatic evaluation, the decision was made to introduce a study where real humans would assess the results produced by the models. The users were asked whether they preferred the output of a base model or a fine-tuned one, in a blind comparison. The study compares the same two systems, the base and the fine-tuned model, as the automatic evaluation, so both evidence types measure the same contrast.

6.2. Study Design

The study utilized a blind, pairwise comparison, where, for each item, raters are presented with the outputs of both base and fine-tuned models side by side and then they are prompted to decide which one is better. Relative judgements like this tend to be easier and more consistent for raters than absolute scoring when the quality differences are subtle. Additionally, raters do not know which output is which, removing the expectation bias. Both outputs come from models of the same family and size, with the only difference that one of them was fine-tuned, so any preference among the user participants can be attributed only to fine-tuning. Each item had two response types. The first one asked the user if the output of one model was better than the other or if they were the same. The second type was a Likert scale that asked the user to rate the quality dimensions of each variant. Each item presents its two model outputs as "Variant A" and "Variant B", both of which are randomized for each item, with a key kept to decode A/B back to base/fine-tuned. The study covers four generative tasks: grammar correction, translation, ABSA and summarization. Sentiment was evaluated automatically and excluded from human review, because it is a classification task with ground truth, not an open-ended generation that can be preferred by a human judge.

6.3. Task and Item Selection

The study consisted of 36 items in total, 10 of which were grammar correction, 10 for translation, 8 for ABSA, and 8 for summarization. Each item was drawn from held-out evaluation sets, which is the same data used in Chapter 5 and not from the training data. Summarization items came from Serbian XL-Sum data [HBI+21], ABSA from

SentiComments.SR [BCN20] and translation from OPUS-100 [ZWTS20]. Grammar-correction items came from the held-out split of the synthetic error-injection set described in Chapter 4. Evaluation sets were explicitly verified for overlaps with the fine-tuning data, so that the same item could not have appeared in training and evaluation. Items consisted of both Cyrillic and Latin entries, consistently testing the digraphia of the language. The outputs were generated under the same prompts and decoding settings as the benchmark from Chapter 5 in order to ensure consistency. The outputs are shown exactly as the model produced them, including cases where the model added extra text instead of only the correction, and cases where the correction was wrong. No edits or filters were applied in favour of any system, which is a strength.

6.4. Rating Instrument and Procedure

Grammar correction used the following dimensions on its Likert scale: errors corrected, meaning preserved and naturalness. Likert scales for translation used faithfulness and fluency. ABSA ones used aspects correct, sentiment correct and no spurious aspects. Finally, summarization used accuracy, coverage and readability. The Likert scales all spanned from one to five, where five was the best grade and one the worst.

Each task included a detailed explanation of how the model was prompted, also mentioning the fact that the model was supposed to return only the corrected sentence, which was not always the case, even with an explicit prompt. This way, the raters could judge the following of the instruction by the model. This neutralizes verbosity bias and does not alter the outputs.

ABSA's structured JSON outputs were rendered into a readable list applied identically to both systems. This way, the non-specialist raters could understand and judge them. Outputs that did not contain valid structured data were flagged and shown as invalid.

The study was done via asynchronous online Google Forms[1], where one item was presented per page, targeting native Serbian speakers (the rating instrument and rater demographics are given in Appendix A.7). The initial recruitment target was 15-20 people, and 16 valid responses were obtained. Participation was voluntary and anonymous: the form opened with a notice stating that, by taking part, respondents consented to their anonymous responses being processed for research purposes. No personally identifying data was collected. Respondents were only asked for anonymous background information (age, native-speaker status, spoken dialect, level of English, and familiarity with NLP), and the dialect and native-speaker breakdown is reported in Appendix A.7.

6.5. Analysis Method

For each task, forced-choice preferences were tested against the chance level of 50%. It was done using a two-sided exact binomial test, to determine whether the model was systematically preferred.

[1]Google Forms, a web-based survey tool: `https://www.google.com/forms/about/`.

Additionally, the Wilcoxon signed-rank test was used for the comparison of Likert ratings, in this case for base vs. fine-tuned, paired per rater, per item and per Likert dimension. This test is a standard and appropriate method for analysing paired observations or ordinal data. Because the dimensions rated within the same rater and item are not fully independent, the resulting p-values are treated as indicative rather than as a strict significance test.

Agreement across the three forced-choice categories (base, fine-tuned and tied) was quantified using Fleiss' kappa, interpreted with the standard agreement bands [LK77].

The randomized A/B tests were decoded back to the base/fine-tuned with the help of a blinding key produced before the analysis. At the end of the study, the users were presented with an attention check question, which, if answered incorrectly, would mark the results as invalid, since the question showed two outputs, one of which was just a random mixture of letters. All 16 received responses passed this question and came from native speakers.

6.6. Results

The results of the user study were unexpected, as seen in Table 6.1. While the fine-tuned model showed clear improvement on the automatic generative-task metrics, it was not preferred by the users. Across all 4 tasks and 16 raters, the base model was in general more preferred, having 268 preferences against 158 preferences for the fine-tuned model, with 150 ties (Table 6.1, Fig. 6.1). For non-tie judgements, 37.1% favoured the fine-tuned model, which is a significant difference for a two-sided binomial test. This contrasts with the automatic evaluation in Chapter 5, where fine-tuning improved multiple metrics.

Table 6.1.: Per-task results of the blind pairwise user study (base vs. fine-tuned Qwen3-14B-Instruct, 16 native-speaker raters). The preference columns give forced-choice counts. "Fine-tuned %" is the fine-tuned share of non-tie decisions. The binomial p is a two-sided exact test against the 50% chance level (ties excluded). κ is Fleiss' inter-rater agreement.

Task	Fine-tuned (n)	Base (n)	Tie (n)	Fine-tuned % (non-tie)	Binomial p	Fleiss' κ
Grammar	48	58	54	45.3	0.3821	0.248
Translation	48	65	47	42.5	0.1319	0.035
ABSA	57	34	37	62.6	0.0206	0.096
Summarization	5	111	12	4.3	4.040×10^{-27}	0.005
Overall	**158**	**268**	**150**	**37.1**	1.099×10^{-7}	**0.222**

Focusing on the per-task judgements as shown in Fig. 6.1, we can see that only the fine-tuned output for ABSA was favoured by the users, reaching 62.6% of the non-tie votes, which is a statistically significant preference ($p = 0.021$), while the summarization results were completely one-sided in favour of the base model, with the fine-tuned summaries

earning only 5 of the 116 non-tie votes, which shows a clear statistical preference. Fine-tuned translation reached 42.5%, showing a slight preference for the base model, while grammar correction reached 45.3%, which cannot be classified as a clear outperformance or underperformance.

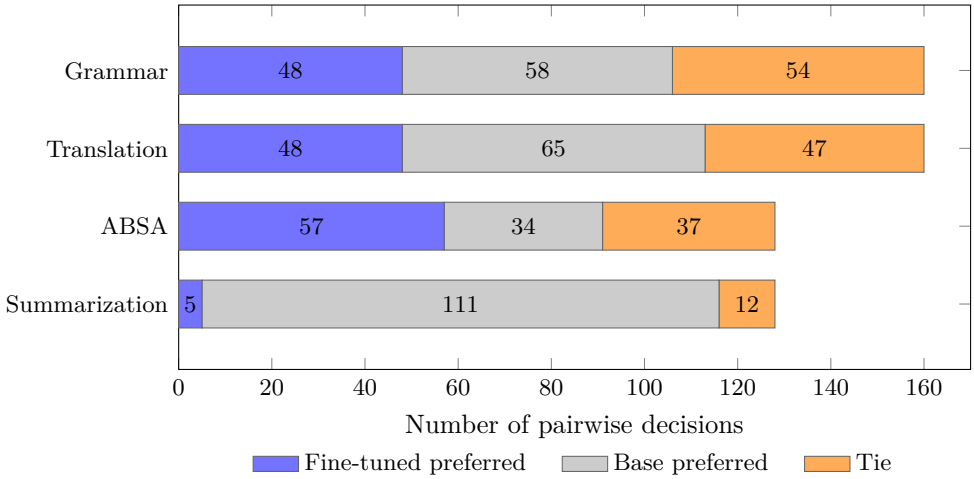

Figure 6.1.: Forced-choice preferences per task in the blind pairwise study (16 native-speaker raters, 576 decisions in total). Each stacked bar shows how many decisions favoured the fine-tuned model, the base model, or neither (tie). The base model was generally preferred, overwhelmingly so for summarization (111 base vs. 5 fine-tuned). ABSA is the only task with a statistically significant preference for the fine-tuned model ($p = 0.021$). The grammar and translation differences are not statistically significant (see Table 6.1).

The results of the per-dimension Likert ratings in Fig. 6.2 further support the previous findings. The outcomes of the Wilcoxon test showed no major favouritism between the models on most tasks, reaching almost identical scores. For the grammar correction, the results pointed to a slight preference for the base model over the fine-tuned one, while the summarization showed the clear dominance of the base model.

Upon further inspection of the outputs of the fine-tuned model for the summarization task, a potential reason for the favouritism was discovered. The fine-tuned model consistently produced shorter summaries than the base model, and these often omitted information, making them unfaithful in certain cases. The base model produced longer summaries, but they encompassed all the core information of the original. This difference in length is shown in Fig. 6.3. On the stored sample outputs, the fine-tuned summaries are on average about a third as long as the base model's summaries.

In Fig. 6.4, we can see an example entry from outside the user study. It clearly showcases the behaviour previously mentioned about the output of the models. Additionally, in this case, we can see that the fine-tuned output misidentified a project about streets in Serbia as concerning streets in Philadelphia and introduced new hallucinated information, while the base summary remained faithful. The fine-tuned model appears to have developed a

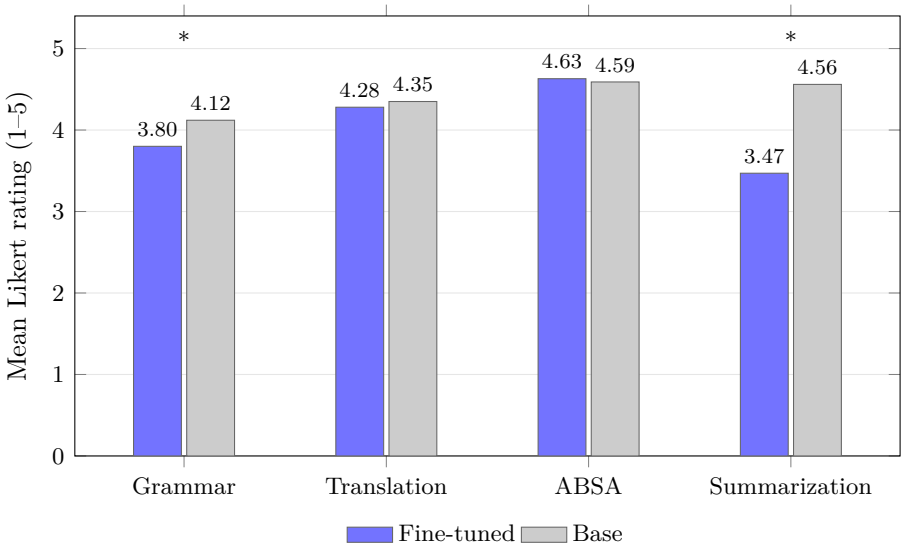

Figure 6.2.: Mean Likert ratings (1–5, higher is better) for fine-tuned versus base outputs, by task. Values are printed above each bar. $*$ marks a statistically significant difference (Wilcoxon signed-rank test: Grammar $p = 0.0001$, Summarization $p < 0.001$). Translation and ABSA are not significant. In both significant cases the base model scored higher.

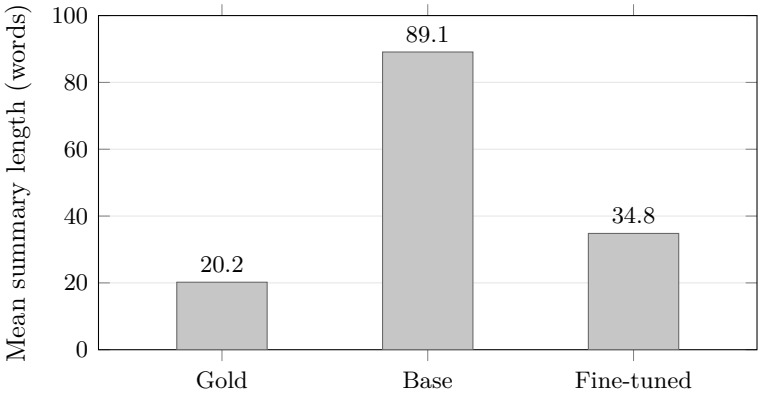

Figure 6.3.: Mean output length (in words) of the base and fine-tuned summaries, together with the gold reference summary, over the 20 stored sample outputs of the summarization evaluation. The base model has on average 89 words, the fine-tuned model 35 (about 2.6 times shorter) and the gold reference 20. In characters the averages are 581 (base), 225 (fine-tuned) and 130 (gold). This supports the over-compression described in Chapters 6 and 7. The fine-tuned model was trained on short Wikipedia leads, so it produces much shorter summaries that leave out content on the longer evaluation texts.

Item S01: summarization example (base vs. fine-tuned model)

Source text:
Он не пева попут Бруса Спрингстина (Улице Филаделфије), већ прикупља историјске податке, фотографије, материјале у жељи да забележи развој града у ком живи. Тренутно је апсолвент Саобраћајног факултета у Београду, а љубав према историји и будуће занимање спојио је и покренуо пројекат. Сајт је, симболично, покренуо у градском архиву. После позитивног реаговања суграђана, Никола сада жели да прошири акцију и на остала места у земљи и повеже их кроз платформу — Улице Србије. Идеја је да на таблама с називом улице стоји и код који би могао да се учита преко мобилног телефона и одведе до сајта и свих информација о улици, њеним значајним становницима и како је изгледала некад.

*English: He does not sing like Bruce Springsteen (*Streets of Philadelphia*), but instead collects historical data, photographs and materials, wishing to document the development of the city he lives in. He is currently a final-year student at the Faculty of Transport in Belgrade, and he has combined his love of history with his future profession by starting a project. He symbolically launched the website in the city archive. After a positive response from fellow citizens, Nikola now wants to expand the initiative to other places in the country and connect them through a platform,* Streets of Serbia*. The idea is that street-name signs would also carry a code that could be scanned with a mobile phone and lead to the website and all information about the street, its notable residents, and how it once looked.*

Base model:
Никола, апсолвент Саобраћајног факултета у Београду, кроз пројекат „Улице Србије" прикупља историјске податке, фотографије и материјале о улицама у Србији. Идеја је да се на таблама с називима улица, преко мобилног телефона, пренесе информација о њиховој историји, значајним особама и претходном изгледу. Пројекат је започет у градском архиву, а Никола сада жели да га прошири на целој територији земље.

English: Nikola, a final-year student at the Faculty of Transport in Belgrade, collects historical data, photographs and materials about streets in Serbia through the project Streets of Serbia*. The idea is that street-name signs would convey, via a mobile phone, information about their history, notable people, and former appearance. The project was started in the city archive, and Nikola now wants to expand it across the whole country.*

Fine-tuned model:
Никола Марковић, 23-огодишњи студент, покренуо је пројекат Улице Филаделфије, који је у суштини виртуелна галерија улица града Филаделфије.

English: Nikola Marković, a 23-year-old student, started the project Streets of Philadelphia*, which is essentially a virtual gallery of the streets of the city of Philadelphia.*

Figure 6.4.: Item S01 (summarization). The base model returns a faithful summary that preserves the core facts of the source, whereas the fine-tuned model returns a much shorter summary that omits most of the content, adds a hallucinated name and age, and misidentifies the topic as the streets of Philadelphia rather than a project documenting streets in Serbia.

bias towards terse output, which helped on tasks that follow a certain format but harmed the summarization, where coverage and faithfulness matter the most. It is important to note that the summarization metric used in automatic evaluation increased from 0.072 to 0.120, yet human raters judged the fine-tuned summaries worse, which is a concrete case of an automatic metric diverging from human quality judgement.

Turning to inter-rater reliability, agreement among raters, measured with Fleiss' kappa, was low overall, only reaching fair values for grammar correction and slight agreement for translation, ABSA and summarization [LK77]. Low agreement is consistent for translation and ABSA, which means that the raters are perceiving the quality of both outputs as similar. It is important to note that the direction of preference was consistent for summarization, even though Fleiss' kappa is low. The reason behind this is that kappa is sensitive to the marginal distribution.

The outputs of the fine-tuned model used for the user study contained two grammar items, where the input was left uncorrected, and one ABSA item, where a malformed structure was produced. These are consistent with mixed-to-negative human scores and demonstrate that real, unedited outputs surface model limitations rather than a sanitized comparison.

6.7. Threats to Validity

The study is based on 16 raters, which is a modest sample for drawing strong conclusions about preferences. However, the result difference achieved for summarization preference is large enough to be considered robust, even at this sample size. The differences for grammar correction and translation can be considered less significant, as they hold limited statistical power rather than true equivalence. A less significant difference should, in this case, be read as "no preference detectable at this sample size" and not as proof that the two models are equivalent in those two tasks. A larger sample might reveal effects this study could not detect.

Fleiss' kappa was low across all tasks, indicating low agreement among raters. Only the agreement on grammar correction achieved "fair" agreement [LK77]. Low agreement means that the individual item preferences are noisy between the users and should not be directly connected to other results. Low agreement for translation and ABSA is consistent with raters perceiving the two outputs as similar in quality. When outputs are close, preferences scatter, which decreases kappa. This means that low kappa on those tasks partly reflects the finding rather than undermining it.

Raters were drawn from a limited, non-random network of native speakers, so they may not represent the broader Serbian-speaking population. Twelve out of sixteen participants reported speaking mainly the Ekavian dialect, three Ijekavian and one mixed, meaning that Ijekavian speakers were underrepresented and results may not showcase the general opinion of all Serbian speakers. Representation across dialects and registers is a recognized challenge in Serbian NLP [AU25].

The evaluation items were constructed with the help of specific sources: news articles were used for summarization [HBI+21] and review-style text for ABSA came from

SentiComments.SR. The findings therefore apply to these domains and registers and may not apply to others like medical, legal, conversational, etc. The summarization finding about the over-compression might not manifest in other text types.

ABSA's structured outputs were transformed from their raw JSON form into a readable list to make it easier for non-technical users to judge them. Even though the transformation was applied identically to both model outputs, it is still a layer between raw model output and what the rater saw and may have influenced the judgements in ways other tasks were not subject to. This means that ABSA results carry an additional caveat that other tasks do not.

Sentiment classification was excluded from human evaluation and assessed only with the automatic benchmark. Beyond that, the study measured the relative preference between the two systems, not absolute quality. Raters were prompted to choose which of the two outputs they thought was better, so the study results say nothing about whether the output meets external quality standards, only the preference. The divergence between automatic metrics and human judgement is itself a limitation of relying on metric-based evaluation alone and is examined as a substantive finding in Chapter 7.

7. Discussion

7.1. Overview

This chapter interprets the results of the automatic evaluation and human study together, analysing what they could mean for the fine-tuning of Serbian LLMs and how such models could be evaluated. The two approaches yielded disagreeing results, which is as interesting as the individual results. The direction this chapter will develop is the following: methodology finding, what fine-tuning improved and harmed, the divergence in human evaluation, deployment trade-offs and the broader lessons learned.

7.2. The Log-Probability Evaluation Gap

The same base model scored approximately 34% under log-probability scoring with the standard evaluation harness [HFK+23]. However, it achieved 75.6% under chat-based scoring on the identical benchmark [dat24], a difference of 41%. The only difference was the scoring method. The model, questions and the answer set were the same, meaning that the gap between the scores is mainly an artefact of evaluation methodology and not model capability. Categories like Boolean questions held up under log-probability scoring while others collapsed. This points to the fact that the scoring method interacted with answer format rather than failing randomly.

Log-probability scoring evaluates a model by which candidate it assigns the highest probability to, whereas chat-based scoring lets the model generate an answer it would actually use. This is a disadvantage for the instruction-tuned models since they are optimized to follow instructions and generate well-formed responses. Their probability mass over raw answer tokens may not reflect that competence [HWS+21], which means that log-probability scoring automatically undermines them. The evaluation method chosen for convenience can therefore misidentify a model's actual capabilities by a wide margin.

Published evaluations of Serbian and other low-resource language models that rely on log-probability scoring may have directly underestimated instruction-tuned models, giving an inaccurate picture of the state of Serbian NLP. The methodological lesson of this thesis is that the methodology must match how models are actually used and that the benchmark scores are not absolute properties of a model but are contingent on the scoring protocol. The documentation and quantification of this gap on a Serbian benchmark is one of the contributions of this work.

7.3. What Fine-Tuning Improved

The measured gains on automatic evaluation were strong on the tasks the model was trained for. The fine-tuning produced large gains on grammar correction, going from zero to 45.5%, and on sentiment classification. A substantial gain was also achieved, especially in aspect extraction, and a modest one in summarization. The general-knowledge benchmark, by contrast, was unchanged: the fine-tuned model scored 74.8% against the base model's 75.6%, because the fine-tuning data targeted task formats and Serbian-language competence rather than factual content. The gains were therefore concentrated in the generative and format-sensitive tasks, which is consistent with the nature of the fine-tuning data.

The fine-tuning process also showed a big qualitative gain. The model showed improvement in adherence to the task format. The fine-tuned model reliably produced clean, task-exact output, whereas the base model frequently added explanations, commentary and preambles. This format-following was real and visible, but as shown in Chapter 6, it did not translate into a human preference on most tasks, especially not on a summarization task, where the strong compression became a liability.

However, even after fine-tuning, the weakest benchmark categories remained reasoning-heavy ones, e.g., moral scenarios, abstract algebra and mathematics. This indicated that the fine-tuning had more influence over output format than over reasoning or factual knowledge. This is consistent with the nature of fine-tuning data, which targeted task formats and Serbian-language competence rather than reasoning ability.

7.4. The Divergence Between Automatic Metrics and Human Judgement

The automatic evaluation and human study did not agree. Automatic metrics indicated strong improvements on fine-tuned tasks, yet native speakers did not prefer the fine-tuned outputs overall and strongly preferred the base model for summarization. On summarization specifically, the automatic metric rose for the fine-tuned model, from 0.072 to 0.120, ranking it as the better of the two, while the human raters strongly preferred the base model, with 111 votes against 5 and a far higher Likert score of 4.56 against 3.47. The metric therefore pointed in the opposite direction to human judgement on the same task. The fine-tuned score of 0.120 is also still low in absolute terms, so its small rise was never strong evidence that the summaries had actually improved. Table 7.1 juxtaposes, per task, the automatic-metric change with the human preference and Likert direction, making the divergence legible at a glance. The metric points to a fine-tuned win on every task where it changes, but the raters concur only on ABSA, and on grammar correction and especially on summarization the metric rises even though the raters still prefer the base model.

Upon further inspection of the fine-tuned model's output, it was discovered that it produced much shorter summaries that omitted information and were sometimes unfaithful to the source. As previously shown in Fig. 6.4, the fine-tuned model mistook

Table 7.1.: Automatic-metric change versus native-speaker judgement, per task (fine-tuned vs. base). "Human pref." is the fine-tuned share of non-tie forced choices. "Likert" gives the mean fine-tuned and base ratings. The metric favours the fine-tuned model on every task where it changes, but raters concur only on ABSA. On grammar correction and especially summarization the metric improves while raters still prefer the base model.

Task	Automatic metric (base → fine-tuned)	Human pref.	Likert (FT / base)
Grammar correction	Exact match 0.000 → 0.455	45.3%	3.80 / 4.12
Translation	BLEU 61.54 → 61.75	42.5%	4.28 / 4.35
ABSA	Aspect-F1 0.433 → 0.622	62.6%	4.63 / 4.59
Summarization	ROUGE-L 0.072 → 0.120	4.3%	3.47 / 4.56

a passing reference for the subject of the text (which was actually about a project documenting streets in Serbia) and added a hallucinated name and age. Unfortunately, the surface-overlap metric used in this project, which was ROUGE-L, missed this. The reason behind this is that ROUGE rewards surface lexical overlap and brevity-driven precision and does not detect unfaithfulness or hallucinated content at all [Lin04, MNBM20, FKM+21]. This means that a shorter summary can score higher on ROUGE while being worse for a human reader.

This difference is a demonstration that automatic generation metrics can mask degradation, and that improvement on a metric does not guarantee improvement in human-perceived quality, especially for tasks like summarization. This is the second time in this work that evaluation methodology determined the conclusion: the first being the log-probability gap. The two of them combined show that conclusions about Serbian LLM quality are highly sensitive to how evaluation is done. The usage of complementary methods should be considered a standard for the robust evaluation of language models for low-resource languages. Preferably, this should be a combination of automatic metrics, generation-based benchmarking and human judgement, since no single method is sufficient on its own, and relying on one can lead to wrong interpretations.

The fine-tuning effect on the output was task-dependent. The usage of format-following constraints helped on tasks like grammar correction, but harmed open-ended ones, like summarization. Therefore, the fine-tuning effect depends on the similarity between the fine-tuning objective and the task. The fine-tuning data should be aligned with the objectives, and aggressive optimization towards specific outputs can degrade tasks requiring completeness. A likely contributing factor for summarization is the training construction itself: the model was trained on short Wikipedia leads as summaries (Chapter 4), which may have biased it towards terse outputs that lose coverage on the longer, news-style texts used in evaluation.

7.5. Deployment Considerations

The fine-tuned model was more accurate at sentiment than the dedicated cardiffnlp classifier. Upon further testing of the classifier, it was discovered that its weakness is the processing of long, out-of-domain reviews, which was not the case for the fine-tuned model [MRL24, BEACC22]. Nevertheless, the deployed web application retains the classifier, since it offers a fast and decently accurate default. The bigger, more accurate models are offered as a selectable option in the app settings.

The final results imply that no model is an optimal choice across all tasks during the deployment. The fine-tuned model is good at format-following and the tasks it was trained on, while the base produces better summaries and dedicated classifiers remain efficient. Therefore, a hybrid approach, which mixes all of them together to achieve the most benefits, seems like the most practical approach, rather than relying on one model. This is what the current architecture adopts.

7.6. Full vs. Parameter-Efficient Fine-Tuning

The usage of LoRA was very effective during the exploratory phase of the thesis, since it allowed for fast fine-tuning on the available hardware. This way, testing of different datasets was made possible in a short period of time [HSW+22]. However, full fine-tuning still produced the strongest results for the final deployed model. The trade-off of LoRA is that, even though it is far cheaper in memory and compute, it still cannot reach the accuracy a full fine-tune would [BPGO+24]. The choice mainly depends on the available hardware for training.

7.7. Limitations of the Work

Several limitations should be kept in mind when interpreting the results. A large part of the training data was produced by a stronger teacher model (Qwen2.5-72B-Instruct-AWQ) and by rule-based synthetic injection rather than gold human annotation. The fine-tuned model therefore partly inherits the teacher's biases and is bounded by its quality, and the synthetic grammar and ABSA data may not fully reflect the errors and review styles found in real Serbian text. The summarization data was built from Wikipedia article leads, which likely biased the model towards overly short summaries.

The experimental scope is also narrow. Only a single model family (Qwen3, at 4B and 14B) was studied, so it is unclear how far the findings generalize to other architectures, and the final 14B model came from a single full fine-tuning run without multiple seeds, so the reported gains carry no estimate of run-to-run variance. Hate-speech detection and semantic role labelling were excluded from the fine-tuning, so the model does not cover every task the application offers.

The evaluation has its own limits. The knowledge-benchmark results rely on a third-party dataset (datatab/serbian-llm-benchmark) that is not independently validated and contains at least one duplicated subset (Chapter 5). The automatic generation metrics

used are surface-level and, as the human study showed, can diverge from perceived quality, so the automatic gains should be read with that caveat. Decontamination was applied where overlaps were detected, but full coverage across a 24,601-example training set cannot be guaranteed, and the SerbMR sentiment test set was smaller afterwards. Finally, full fine-tuning and serving require substantial hardware (the NVIDIA DGX Spark used here), so the local-only deployment path is constrained.

The human study was modest in size and drawn from a limited pool of Serbian speakers. Its full set of threats to validity is discussed in the Threats to Validity section of Chapter 6.

8. Conclusion

8.1. Summary of the Work

The objective of the thesis was to investigate the fine-tuning of open-weight LLMs for Serbian multi-task NLP, along with building a web application demonstrating these capabilities for a low-resource language. The outcome was a unified Serbian NLP web application utilizing a hybrid system of modules supporting multiple tasks (grammar correction, translation, sentiment, aspect-based sentiment, and summarization), alongside other supporting modules, with the fine-tuned open-weight model as its intellectual core. The work treated the application as a deliverable, and the fine-tuning and its evaluation as the research contribution.

An open-weight Qwen3 model [Y+25] was fine-tuned on an instruction-formatted, multi-task Serbian dataset, consisting of 24,601 examples across all five tasks. The model was then evaluated both automatically and through a user study. The evaluation combined automatic benchmarking, task-specific metrics, and native-speaker human judgement, which was a deliberate approach.

8.2. Key Findings

Returning to the research questions posed in Chapter 1: the first asked whether fine-tuning improves automatic-benchmark performance, and the answer is task-dependent. It produced large gains on grammar correction (from zero to 45.5%) and sentiment, a substantial gain on ABSA and a modest one on summarization, while translation stayed essentially flat (BLEU 61.5 to 61.8), where the base model was already near the ceiling. It left the general-knowledge benchmark unchanged (74.8% against the base model's 75.6%), since that knowledge was never part of the fine-tuning data. The second asked whether native speakers prefer the fine-tuned outputs, and the answer was largely no: only ABSA was preferred, summarization strongly favoured the base model, and grammar and translation showed no clear preference. The third asked about the trade-offs of deploying a fine-tuned model versus task-specific classifiers, and no single model proved optimal, since the fine-tuned model is stronger on format-following and the tasks it was trained on while the base produces better summaries and the dedicated classifiers remain fast and efficient, motivating the hybrid deployment the application adopts.

The same base model scored ≈34% under log-probability scoring but ≈75.6% under chat-based scoring on the identical benchmark, a difference of ≈41% solely attributable to the evaluation method, with implications for how Serbian LLMs have been assessed so far. The automatic metrics and human evaluation also diverged substantially. Fine-tuning

improved the metrics, yet native speakers did not prefer the fine-tuned outputs overall and strongly preferred the outputs of the base model for summarization, where the automatic metric ROUGE was improved. This shows that metric improvement does not automatically guarantee human-perceived quality.

The fine-tuning achieved large improvements on grammar correction and sentiment, a substantial gain in aspect extraction, and it improved instruction-following and task-format adherence. It did not, however, add general knowledge, since the fine-tuned model matched the base model on the knowledge benchmark. Fine-tuning was therefore not beneficial for every task. Its format-following bias helped constrained tasks but harmed summarization, where coverage and faithfulness suffered. The value that was achieved through fine-tuning was task-dependent. No single model was optimal for all tasks, which motivates the hybrid deployment that routes each task to its strongest component.

8.3. Contributions

The main contributions of this work are:

1. Quantifying the log-probability vs. chat-based scoring gap on a Serbian benchmark, and documenting a clear divergence between automatic metrics and human judgement for the Serbian model outputs.

2. Demonstrating that an open-weight model can be adapted for multiple Serbian NLP tasks, with significant gains on grammar correction, sentiment and aspect extraction.

3. Integrating these capabilities into a usable system for a low-resource language.

4. Clearing the datasets of overlapping data, applying decontamination where necessary, and honestly reporting the results, including the underperformance on specific tasks.

5. Providing a Serbian native-speaker study that complemented and sometimes contradicted automatic metrics.

8.4. Future Work

A larger and more dialectally balanced human study would strengthen the final findings, given this study's modest size and predominantly Ekavian pool. Future fine-tuning could address the over-compression problem the model exhibited by adjusting the fine-tuning data or objective to reward completeness and faithfulness, not just conciseness. The knowledge-benchmark/training overlap should also be verified exhaustively. The exploration of evaluation metrics that detect unfaithfulness and hallucination [JLF+23] is a natural direction for the Serbian language. The approach can also be extended to closely related Slavic languages like Croatian and Bosnian, leveraging their mutual

intelligibility and shared resources as a promising direction. There is also potential to address dialectal and script variation more explicitly in both data and evaluation.

8.5. Closing Remarks

The work shows that open-weight models can be meaningfully adapted for Serbian NLP, but this adaptation requires careful, multi-method evaluation to inspect whether it truly improved qualitatively and quantitatively. The conclusions made about the model quality heavily depend on how the evaluation/study is conducted, and this applies beyond Serbian to low-resource languages in general. At the same time, these conclusions are tempered by the study's modest scale, its reliance on a single model family, and the third-party benchmark used, as set out in the limitations in Chapter 7. This web application is a sincere and necessary contribution to accessible NLP tools for Serbian speakers.

Bibliography

[Ale06] Ronelle Alexander. *Bosnian, Croatian, Serbian, a Grammar: With Sociolinguistic Commentary*. University of Wisconsin Press, Madison, WI, 2006.

[AU25] Smiljana Antonijević Ubois. From data scarcity to data care: Reimagining language technologies for Serbian and other low-resource languages. arXiv preprint arXiv:2512.10630, 2025.

[B+20] Tom B. Brown et al. Language models are few-shot learners. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[BCN20] Vuk Batanović, Miloš Cvetanović, and Boško Nikolić. A versatile framework for resource-limited sentiment articulation, annotation, and analysis of short texts. *PLOS ONE*, 15(11):e0242050, 2020.

[BEACC22] Francesco Barbieri, Luis Espinosa-Anke, and Jose Camacho-Collados. XLM-T: Multilingual language models in Twitter for sentiment analysis and beyond. In *Proceedings of the Language Resources and Evaluation Conference (LREC)*, 2022.

[BNJ03] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[BNM16] Vuk Batanović, Boško Nikolić, and Milan Milosavljević. Reliable baselines for sentiment analysis in resource-limited languages: The Serbian movie review dataset. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC)*, pages 2688–2696, 2016.

[BPGO+24] Dan Biderman, Jacob Portes, Jose Javier Gonzalez Ortiz, Mansheej Paul, et al. LoRA learns less and forgets less. *Transactions on Machine Learning Research (TMLR)*, 2024.

[BSS+24] Stella Biderman, Hailey Schoelkopf, Lintang Sutawika, Leo Gao, et al. Lessons from the trenches on reproducible evaluation of language models. arXiv preprint arXiv:2405.14782, 2024.

[BYQ+23] Christopher Bryant, Zheng Yuan, Muhammad Reza Qorib, Hannan Cao, Hwee Tou Ng, and Ted Briscoe. Grammatical error correction: A survey of the state of the art. *Computational Linguistics*, 49(3):643–701, 2023.

[CKG+20] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. In *Proceedings of ACL*, 2020.

[Cve24] Ivan Cvetanović. Morphological, syntactic, and semantic analysis of the Serbian language with visualization. Bachelor's thesis, University of Vienna, 2024. Available at `https://github.com/IvanCvetanovic/SerbianLanguageAnalyzer/blob/main/docs/reports/Bachelor_Thesis_Cvetanovic.pdf`.

[Cve25] Ivan Cvetanović. Praktikum 1 report. Technical report, University of Vienna, 2025. Available at `https://github.com/IvanCvetanovic/SerbianLanguageAnalyzer/blob/main/docs/reports/Praktikum1_Report_Cvetanovic.pdf`.

[Cve26] Ivan Cvetanović. Praktikum 2 report. Technical report, University of Vienna, 2026. Available at `https://github.com/IvanCvetanovic/SerbianLanguageAnalyzer/blob/main/docs/reports/Praktikum2_Report_Cvetanovic.pdf`.

[dat24] datatab. Serbian LLM evaluation benchmark dataset. `https://huggingface.co/datasets/datatab/serbian-llm-benchmark`, 2024.

[DCLT19] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, 2019.

[dMMNZ21] Marie-Catherine de Marneffe, Christopher D. Manning, Joakim Nivre, and Daniel Zeman. Universal Dependencies. *Computational Linguistics*, 47(2):255–308, 2021.

[FKM+21] Alexander R. Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. SummEval: Re-evaluating summarization evaluation. *Transactions of the Association for Computational Linguistics (TACL)*, 9:391–409, 2021.

[G+24] Aaron Grattafiori et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[GJ02] Daniel Gildea and Daniel Jurafsky. Automatic labeling of semantic roles. *Computational Linguistics*, 28(3):245–288, 2002.

[goo20] googletrans developers. googletrans: Free Google Translate API for Python. `https://pypi.org/project/googletrans/`, 2020.

[Gor23] Aleksa Gordić. YugoGPT: An open-source large language model for Serbian and other BCS languages. `https://huggingface.co/gordicaleksa/YugoGPT`, 2023.

[HBB+21] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations (ICLR)*, 2021.

[HBD+20] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. In *International Conference on Learning Representations (ICLR)*, 2020.

[HBI+21] Tahmid Hasan, Abhik Bhattacharjee, Md. Saiful Islam, Kazi Mubasshir, Yuan-Fang Li, Yong-Bin Kang, M. Sohel Rahman, and Rifat Shahriyar. XL-Sum: Large-scale multilingual abstractive summarization for 44 languages. In *Findings of ACL-IJCNLP*, 2021.

[HFK+23] Nathan Habib, Clémentine Fourrier, Hynek Kydlíček, Thomas Wolf, and Lewis Tunstall. LightEval: A lightweight framework for LLM evaluation. `https://github.com/huggingface/lighteval`, 2023.

[HLLE14] Florian Heimerl, Steffen Lohmann, Simon Lange, and Thomas Ertl. Word cloud explorer: Text analytics based on word clouds. In *47th Hawaii International Conference on System Sciences (HICSS)*, pages 1833–1842, 2014.

[HSW+22] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022.

[HVD15] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531, 2015.

[HWS+21] Ari Holtzman, Peter West, Vered Shwartz, Yejin Choi, and Luke Zettlemoyer. Surface form competition: Why the highest probability answer isn't always right. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7038–7051, 2021.

[JLF+23] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Yejin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12), 2023.

[JSB+20] Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. The state and fate of linguistic diversity and inclusion in the

NLP world. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2020.

[KB15] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

[KLZ+23] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles (SOSP)*, 2023.

[KMS+25] Smita Khapre, Melkamu Abay Mersha, Hassan Shakil, Jonali Baruah, and Jugal Kalita. Toxicity in online platforms and AI systems: A survey of needs, challenges, mitigations, and future directions. arXiv preprint arXiv:2509.25539, 2025.

[KPRSL26] Taja Kuzman Pungeršek, Peter Rupnik, Vít Suchomel, and Nikola Ljubešić. The growing gains and pains of iterative web corpora crawling: Insights from South Slavic CLASSLA-web 2.0 corpora. arXiv preprint arXiv:2601.11170, 2026.

[LH19] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019.

[Lin04] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, 2004.

[Liu12] Bing Liu. Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language Technologies*, 5(1), 2012.

[LK77] J. Richard Landis and Gary G. Koch. The measurement of observer agreement for categorical data. *Biometrics*, 33(1):159–174, 1977.

[LL21] Nikola Ljubešić and Davor Lauc. BERTić – the transformer language model for Bosnian, Croatian, Montenegrin and Serbian. In *Proceedings of the 8th Workshop on Balto-Slavic Natural Language Processing (BSNLP)*, pages 37–42, 2021.

[LNN+23] Viet Dac Lai, Chien Van Nguyen, Nghia Trung Ngo, Thuat Nguyen, Franck Dernoncourt, Ryan A. Rossi, and Thien Huu Nguyen. Okapi: Instruction-tuned large language models in multiple languages with reinforcement learning from human feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP): System Demonstrations*, pages 318–327, 2023.

[MAM23] Ulfeta Marovac, Aldina Avdić, and Nikola Milošević. A survey of resources and methods for natural language processing of Serbian language. arXiv preprint arXiv:2304.05468, 2023.

[MNBM20] Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1906–1919, 2020.

[MRL24] Michal Mochtak, Peter Rupnik, and Nikola Ljubešić. The ParlaSent multilingual training dataset for sentiment identification in parliamentary proceedings. In *Proceedings of LREC-COLING 2024*, pages 16024–16036, 2024.

[MRS08] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[MT04] Rada Mihalcea and Paul Tarau. TextRank: Bringing order into texts. In *Proceedings of EMNLP*, 2004.

[Oll23] Ollama. Ollama: Run large language models locally. `https://ollama.com`, 2023.

[OWJ+22] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, et al. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[PBMW99] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The PageRank citation ranking: Bringing order to the web. Technical Report SIDL-WP-1999-0120, Stanford InfoLab, 1999.

[PGK05] Martha Palmer, Daniel Gildea, and Paul Kingsbury. The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106, 2005.

[PGP+16] Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, et al. SemEval-2016 task 5: Aspect based sentiment analysis. In *Proceedings of SemEval*, 2016.

[Pop15] Maja Popović. chrF: Character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, 2015.

[PRWZ02] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2002.

[PUL20] Giancarlo Perrone, Jose Unpingco, and Haw-minn Lu. pyvis: Interactive network visualizations. `https://pyvis.readthedocs.io`, 2020.

[PVG+11] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[RKX+23] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, 2023.

[RL22] Peter Rupnik and Nikola Ljubešić. ASR training dataset for Serbian JuzneVesti-SR v1.0. Slovenian language resource repository CLARIN.SI, `http://hdl.handle.net/11356/1679`, 2022.

[ŠP26] Mihailo Škorić and Cosimo Palma. Wiki dumps to training corpora: South Slavic case. arXiv preprint arXiv:2604.25384, 2026.

[SS18] Noam Shazeer and Mitchell Stern. Adafactor: Adaptive learning rates with sublinear memory cost. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, 2018.

[SW17] Anna Schmidt and Michael Wiegand. A survey on hate speech detection using natural language processing. In *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media*, pages 1–10, 2017.

[TL23] Luka Terčon and Nikola Ljubešić. CLASSLA-Stanza: The next step for linguistic processing of South Slavic languages. arXiv preprint arXiv:2308.04255, 2023.

[ÜAY+24] Ahmet Üstün, Viraat Aryabumi, Zheng-Xin Yong, Wei-Yin Ko, Daniel D'souza, Gbemileke Onilude, Neel Bhandari, Shivalika Singh, Hui-Lee Ooi, Amr Kayid, et al. Aya model: An instruction finetuned open-access multilingual language model. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2024.

[VSP+17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[WBZ+22] Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations (ICLR)*, 2022.

[WDS+20] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Transformers: State-of-the-art natural language processing. In *Proceedings of EMNLP 2020: System Demonstrations*, pages 38–45, 2020.

[WKM+23] Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*, 2023.

[XCR+21] Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of NAACL-HLT*, 2021.

[Y+25] An Yang et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

[ZWTS20] Biao Zhang, Philip Williams, Ivan Titov, and Rico Sennrich. Improving massively multilingual neural machine translation and zero-shot translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1628–1639, 2020.

A. Appendix

A.1. Prompt Templates

Every generative task was queried with a Serbian system instruction, with the item placed in the user turn. Decoding was deterministic (temperature 0, greedy), and any `<think>` reasoning span was removed before scoring. The instructions used at inference, and (except for sentiment) also to generate the user-study outputs, were the following, with English glosses in parentheses:

Grammar correction *"Ispravi gramatičke greške u sledećoj rečenici."* ("Correct the grammatical errors in the following sentence.") Training items additionally used paraphrased instructions in both Cyrillic and Latin script.

Translation *"Prevedi sledeći tekst sa srpskog na engleski."* ("Translate the following text from Serbian into English.")

Summarization *"Napiši kratak rezime sledećeg teksta."* ("Write a short summary of the following text.")

Sentiment *"Ti si asistent za analizu sentimenta srpskog teksta. Odgovori SAMO jednom rečju: 'positive', 'negative' ili 'neutral'. Bez ikakvog objašnjenja."* ("You are an assistant for sentiment analysis of Serbian text. Answer with ONLY one word: 'positive', 'negative', or 'neutral'. Without any explanation.")

Aspect-based sentiment analysis required a strict JSON schema:

ABSA system instruction

```
Analiziraj aspekte i njihov sentiment u sledećem tekstu. Odgovor daj kao
JSON niz u formatu:
[{"sentence": "...", "aspects": [{"aspect": "...", "sentiment":
"positive|negative|neutral", "confidence": 0.0-1.0, "evidence":
"..."}]}]
Odgovori SAMO JSON-om, bez ikakvog dodatnog teksta.
```

("Analyse the aspects and their sentiment in the following text. Provide the answer as a JSON array in the given format. Reply with ONLY JSON, without any additional text.")

A.2. Grammar Data: Error-Injection Rules

The grammar-correction training set (≈6,000 pairs) was built by rule-based corruption of clean Serbian sentences (length 40–200 characters, low digit ratio) drawn from the summarization corpus. Each clean sentence was corrupted by a single randomly selected rule (sampled with the relative weights in Table A.1) and stored as a (corrupted, corrected) pair. Both Cyrillic and Latin scripts were handled.

Table A.1.: Error-injection rules used to synthesize the grammar-correction data, with illustrative Latin-script examples and the relative sampling weight of each rule.

Rule	Example (corrupted ← correct)	Weight
Diacritic stripping (š→s, ž→z, č/ć→c, đ→dj)	*covek* ← *čovek*	3
Negation spacing	*ne ću* ← *neću*	2
č/ć confusion	*kuča* ← *kuća*	2
Case-ending swap	*gradu* ← *gradom*	2
Character typo (swap/double/drop)	*rekka* ← *reka*	2
ije/je swap	*djete* ← *dijete*	1
Capitalization	*beograd* ← *Beograd*	1
Punctuation (drop/duplicate)	*Zdravo* ← *Zdravo.*	1
Doubled word	*u u gradu* ← *u gradu*	1

A.3. Fine-Tuning Hyperparameters

Table A.2 reports the configuration of the deployed full fine-tune (Qwen3-14B-Instruct) and the exploratory parameter-efficient run (Qwen3-4B with LoRA) discussed in Chapter 7. Both were trained on an NVIDIA DGX Spark (GB10, 128 GB unified memory). The 14B run additionally used gradient checkpointing and masked the loss to the assistant response.

A.4. Compute and Reproducibility Environment

All fine-tuning and inference were run on an NVIDIA DGX Spark (GB10, 128 GB unified memory). Models were served through a vLLM OpenAI-compatible endpoint in `bfloat16` with a maximum context of 4096 tokens (`-enforce-eager`). Generation used greedy decoding (temperature 0). Any `<think>` reasoning span emitted by the model was removed before scoring. The base model was `OpenPipe/Qwen3-14B-Instruct`, and training relied on the Hugging Face Transformers, TRL, and PEFT libraries. All generative and sentiment evaluation sets were checked for overlap with the training data and decontaminated where necessary, so that no evaluation item appeared in training. The decontaminated sentiment test sets contain 600 ParlaSent and 168 SerbMR items.

Table A.2.: Fine-tuning configuration for the deployed full fine-tune and the exploratory LoRA run.

Setting	Full fine-tune (deployed)	Parameter-efficient (exploratory)
Base model	OpenPipe/Qwen3-14B-Instruct	Qwen/Qwen3-4B
Adaptation	Full fine-tuning	LoRA (r=32, α=64, dropout 0.05)
LoRA target modules	—	q,k,v,o,gate,up,down proj.
Epochs	1	2
Per-device batch	1	2
Grad. accumulation	16	8
Effective batch	16	16
Learning rate	1×10^{-5}	1×10^{-4}
LR schedule	cosine, 3 % warmup	cosine, 5 % warmup
Precision	bf16	fp32
Max. sequence length	2,048	TRL default
Optimizer	Adafactor	AdamW (default)
Framework	HF `Trainer`	TRL `SFTTrainer`

A.5. Data, Model and Code Availability

The artefacts produced in this thesis are publicly available in order to support reproducibility. The fine-tuned 14B model, named *Slava*, is released on the Hugging Face Hub at `https://huggingface.co/IvanCvetanovic/Slava-Qwen3-14B-Serbian`, together with the training script and a per-source provenance and licence summary. The instruction-tuning dataset is not redistributed as a single corpus, since its sources carry incompatible licences. Instead, the data-construction scripts and pointers to the original public datasets (Serbian Wikipedia, OPUS-100 [ZWTS20], SentiComments.SR [BCN20] and ParlaSent [MRL24]) are provided, so that the dataset can be regenerated from the originals. The source code of the web application is available at `https://github.com/IvanCvetanovic/SerbianLanguageAnalyzer`. The released model weights are licensed under CC BY-NC-SA 4.0 for non-commercial research use, which reflects the most restrictive licence among the training-data sources, and the model is built with Qwen. Anyone reusing these artefacts is asked to cite this work and the underlying datasets.

A.6. Training-Set Composition

The instruction-tuning mix comprised 24,601 examples across the five tasks, formatted as system/user/assistant chat turns with paraphrased instructions in both Cyrillic and Latin script. Table A.3 gives the per-task composition and the source of each subset.

Table A.3.: Composition of the multi-task instruction-tuning set (24,601 examples).

Task	Examples	Source
Summarization	6,000	Serbian Wikipedia
Grammar correction	6,000	Synthetic error injection (App. A.2)
Translation	5,000	OPUS-100 [ZWTS20] + teacher-distilled Wikipedia
ABSA	4,000	SentiComments.SR [BCN20] (2,500) + synthetic reviews (1,500)
Sentiment	3,601	SentiComments.SR + ParlaSent (BCS) [MRL24]
Total	**24,601**	

A.7. User-Study Materials

Each item presented the base and fine-tuned outputs as "Variant A" and "Variant B" in a randomized order, with a key kept to decode A/B back to the systems only after analysis. An attention-check item (in which one option was a random string of letters) was included. All 16 responses passed it and came from self-reported native speakers. For every item, raters gave a forced-choice preference and rated each variant on the task-specific Likert dimensions in Table A.4 (each scored from 1 to 5, higher is better). The rater demographics are summarized in Table A.5, and the per-task mean Likert ratings in Table A.6.

Table A.4.: Likert rating dimensions per task (each scored 1–5).

Task	Dimensions
Grammar correction	errors corrected, meaning preserved, naturalness
Translation	faithfulness, fluency
ABSA	aspects correct, sentiment correct, no spurious aspects
Summarization	accuracy, coverage, readability

Table A.5.: Rater demographics ($n = 16$, all self-reported native speakers).

Attribute	Count
Predominantly Ekavian dialect	12
Predominantly Ijekavian dialect	3
Mixed dialect	1
Native speakers	16

A.8. Example Model Outputs

This section shows real, unedited outputs of the base and the fine-tuned model on the held-out evaluation items from Chapter 5, placed side by side. They make the patterns from the error analysis (Sect. 5.5) concrete. Every output is shown exactly as the model

Table A.6.: Mean Likert ratings (1–5) per task, with the fine-tuned minus base difference (Δ, computed from unrounded means), the paired Wilcoxon signed-rank p-value, and the number of paired ratings. Complements Fig. 6.2.

Task	Fine-tuned	Base	Δ	Wilcoxon p	Pairs
Grammar	3.80	4.12	−0.32	1.365×10^{-4}	480
Translation	4.28	4.35	−0.07	0.2586	320
ABSA	4.63	4.59	+0.05	0.3040	384
Summarization	3.47	4.56	−1.09	1.614×10^{-36}	384

produced it. Long outputs are shortened with [...], and nothing was edited to favour any of the models. ABSA is not included here, because its per-item outputs were not saved.

Grammar correction

Input: On je uvek bio najbrzi trkac u nasoj generaciji, niko nije mogao da ga stigne.

Reference: On je uvek bio najbrži trkač u našoj generaciji, niko nije mogao da ga stigne.

Base: Ispravljena rečenica glasi: **"On je uvek bio najbrži trkač u našoj generaciji, niko nije mogao da ga stigne."** **Objašnjenje:** - **"brzi trkac"** → **"brži trkač"** – ispravno je koristiti pridjev u pravom obliku (uporedbeni oblik "brži" i pravilno imenice "trkač"). - **"nasoj"** → **"našoj"** – [...]

Fine-tuned: On je uvek bio najbrži trkač u našoj generaciji, niko nije mogao da ga stigne.

The fine-tuned output is an exact match. The base model gives a correct fix, but it wraps it in commentary, so the exact-match score rejects it.

Input: Voznja bicikla pored reke je moj omiljeni nacin da se opustim posle napornog radnog dana.

Reference: Vožnja bicikla pored reke je moj omiljeni način da se opustim posle napornog radnog dana.

Base: Ispravljena rečenica glasi: **"Voznja bicikla pored reke je moj omiljeni način da se opustim posle napornog radnog dana."** Izmenjeno je: - **"nacin" → "način"** (pravopisna greška, pravilno je "način" sa kritikom).

Fine-tuned: Voznja bicikla pored reke je moj omiljeni način da se opustim posle napornog radnog dana.

The fine-tuned model fixes nacin→način*, but it leaves* Voznja *uncorrected, which is an under-correction failure.*

Translation (sr→en)

Input: Zbog nezapamćene suše koja je ovog leta pogodila veći deo zemlje, poljoprivrednici očekuju drastičan pad prinosa kukuruza i pšenice. Udruženja poljoprivrednika

su zatražila hitnu pomoć od države kako

Reference: Because of the unprecedented drought that hit most of the country this summer, farmers expect a drastic drop in corn and wheat yields. Farmers' associations have requested urgent aid from the state to compensate for the damage and avoid a potential crisis in the supply of basic food staples this autumn.

Base: Due to an unprecedented drought that has affected much of the country this summer, farmers expect a drastic decline in corn and wheat yields. Farmers' associations have requested urgent government assistance to compensate for the damage and prevent a potential crisis in the supply of basic [...]

Fine-tuned: Due to the unprecedented drought that has affected much of the country this summer, farmers expect a drastic decline in corn and wheat yields. Farmers' associations have requested urgent assistance from the state to compensate for the damage and prevent a potential crisis in the supply of basic [...]

Both translations are faithful and fluent. The differences are only small word choices, which matches the almost flat BLEU/chrF (ceiling effect).

Summarization

Input: Худ је почео да вежба издржај 2011. Он је у овом положају остао осам сати, 15 минута и 15 секунди. Џорџ Худ поставио је рекорд у категорији Најдужи абдоменални мушки планк у Чикагу, саопштили су из Ги

Reference: Бивши амерички маринац у 62. години је оборио светски рекорд у издржају, који се популарно назива планк.

Base: Џорџ Худ је поставио нови светски рекорд у категорији Најдужи абдоменални мушки планк, одржавши се у том положају осам сати, 15 минута и 15 секунди у Чикагу. Овај рекорд надмашио је претходни рекорд осам сати и један минут, који је поставио Мао Веидонг из Кине 2016. године. Худ је за припрему [...]

Fine-tuned: Џорџ Худ, 34-огодишњи амерички тренер из Чикага, поставио је нови светски рекорд у категорији Најдужи абдоменални мушки планк.

The fine-tuned summary hallucinates a 34-year-old trainer*, while the source and the reference describe a 62-year-old former marine. The base summary is longer, but it is faithful.*

Input: Poznat je po krugovima oko očiju Popa langur, nazvan po staništu - planini Popa, kritično je ugrožen pošto je populacija pala na oko 200 jedinki. Languri su grupa majmuna biljojeda koji naseljavaju po

Reference: Do sada nepoznata vrsta majmuna otkrivena je u udaljenim šumama Mjanmara.

Base: **Rezime:** Popa langur, nova vrsta majmuna sa prepoznatljivim flekama oko očiju, otkrivena je u Mjanmaru. Ova kritično ugrožena vrsta, koja živi isključivo u zaštićenim šumama planine Popa, ima populaciju od oko 200 jedinki. Naučnici su koristili DNK uzorke iz muzeja i fizičke karakteristike da [...]

Fine-tuned: U Mjanmaru je otkrivena nova vrsta majmuna, popa langur, koji je ugrožen izumiranjem. Naučnici su otkrili da se radi o novoj vrsti, koja se razlikuje od drugih vrsta majmuna u grupi popa langura.

The base summary is longer and more complete, while the fine-tuned one is much shorter (the over-compression pattern from Chapter 6).

Sentiment (SerbMR)

Input: Betmen počinje gigantski je pomak u odnosu na Schumacherovu lakrdiju od pre mnogo godina, ali ipak leži u senci autentičnog gotskog ugođaja Tima Burtona. Novi Betmen je utemeljeniji u realnosti, ili j

Reference: neutral

Base: negative

Fine-tuned: neutral

Reference: neutral. *The base model predicts* negative *too often on this mixed review, while the fine-tuned model gives the neutral label.*

Input: Kada sam video kakvi sve glumci učestvuju u ovom projektu, naprosto nisam mogao da dočekam priliku da pogledam "The Incredible Burt Wonderstone". Iako sam u startu znao da ne treba nešto previše da oč

Reference: neutral

Base: negative

Fine-tuned: neutral

Reference: neutral. *The same pattern: the base model answers* negative*, and the fine-tuned model* neutral.

A.9. Training Data Examples

To show what the unified instruction format from Chapter 4 looks like in practice, this section gives one real training example per task, taken from the training mix and shown in the system/user/assistant chat format that the model was trained on. Long inputs and responses are shortened with [. . .]. These examples show how the five tasks are separated only by their instructions, and not by task-specific model heads.

Grammar correction

System: Napiši gramatički ispravnu verziju sledeće rečenice.

User: Kruna gornjeg centralnog sekutića je veća od krune ostalih sekutića u obe vilice, i pokazuje pravilan znak ugla i lka i mezijalni nagib korena.

Assistant: Kruna gornjeg centralnog sekutića je veća od krune ostalih sekutića u obe vilice, i pokazuje pravilan znak ugla i luka i mezijalni nagib korena.

Translation

System: Prevedi ovaj tekst sa srpskog na engleski jezik.

User: Partizani preusmerava ovde. Za ostala značenja vidi Partizani (višeznačna odrednica) Narodnooslobodilačka vojska Jugoslavije (NOVJ; ;) bila je oružana sila Narodnooslobodilačkog pokreta Jugoslavije (NOP), predvođenog Komunističkom partijom Jugoslavije (KPJ). Nastala je juna 1941. formiranjem prvih partizanskih odreda, a 1. marta 1945. prerasla je u [...]

Assistant: Partisans redirects here. For other meanings, see Partisans (disambiguation). The National Liberation Army of Yugoslavia (NOVJ) was the armed force of the National Liberation Movement of Yugoslavia (NOP), led by the Communist Party of Yugoslavia (KPJ). It was formed in June 1941 with the establishment of the first partisan detachments, and on March 1, 1945, [...]

Summarization

System: Napiši kratak rezime sledećeg teksta.

User: Istorija Meri Tiso (1761.—1850), rođena Mari Grosholc u Strazburu, u Francuskoj, radila je kao kućna pomoćnica kod doktora Filipa Kirtjisa, koji je bio vešt u modeliranju. Kirtjis je naučio Tisoovu umetnosti modeliranja. Godine 1765. godine Kirtjis je napravio figuru Mari-Žan di Bari , ljubavnice Luja XV i ovo je ujedno najstarija izložena figura. Prva [...]

Assistant: Madam Tiso () je muzej voštanih figura u Londonu. Osnovala ga je Meri Tiso (engleski izgovor: Meri Tuso), koja je bila vajarka voštanih figura. Muzej ima mnogobrojne ogranke u velikim svetskim gradovima.

Sentiment

System: Ti si asistent za analizu sentimenta srpskog teksta. Odgovori SAMO jednom rečju: 'positive', 'negative' ili 'neutral'. Bez ikakvog objašnjenja.

User: Notes on a Scandal je pokušaj da se snimi triler tipa Fatalne privlačnosti za art-house publiku. Sa glumačkom ekipom koju čine dve najbolje glumice sadašnjice - Judi Dench i Cate Blanchett - i scenariom neuobičajene pismenosti, film skoro da postiže svoj cilj. Međutim, da bi događaje priveo kraju, Notes on a Scandal poseže za ružnim i očiglednim lukavstvom [...]

Assistant: neutral

Aspect-based sentiment (ABSA)

System: Analiziraj sentiment po aspektima u sledećem tekstu. Odgovor daj kao JSON.

User: Odnos Tima i njegovog oca je apsolutno perfektno dočaran i predstavljen na filmu. Ovu kritiku je trebao da napiše Goran a ne Sandra. Očigledno nismo gledali isti film.

Assistant: [{ "sentence": "Odnos Tima i njegovog oca je apsolutno perfektno dočaran i predstavljen na filmu.", "aspects": [{ "aspect": "odnos Tima i njegovog oca", "sentiment": "positive", "confidence": 0.95, "evidence": "apsolutno perfektno dočaran i predstavljen" }] }, { "sentence": "Ovu kritiku je trebao da napiše Goran a ne Sandra.", "aspects": [] }, { [...]