Sentiment Analysis Project Report

Detecting and Correcting Noisy Labels in Slovenian Sentiment Datasets

Authors: Pia Polutnik and Lea Vodopivec

Mentor: Gaurish Pandurang Thakkar, PhD

Faculty of Humanities and Social Sciences, University of Zagreb

Natural Language Processing, Academic Year 2025/26

1 INTRODUCTION

1.1 BACKGROUND AND MOTIVATION

Sentiment analysis is a fundamental task in natural language processing (NLP) that aims to identify and extract subjective information from text. While significant progress has been made in sentiment analysis for major languages such as English and Chinese, low-resource languages like Slovenian face unique challenges. These challenges include limited annotated datasets, scarce language-specific models, and the inherent complexity of Slavic morphology.

A key challenge in sentiment analysis research is the presence of noisy labels in training datasets. These are cases where the labeled sentiment does not match the actual sentiment of the text, which can hurt model performance and make evaluations unreliable. This issue is especially common in manually annotated datasets, where annotator disagreement or ambiguous cases can introduce systematic errors.

This project addresses these challenges by systematically investigating label noise in Slovenian sentiment datasets. This project develops a comprehensive pipeline that combines automated noise detection methods with human verification, ultimately producing cleaner training data and improved sentiment classifiers.

1.2 RESEARCH QUESTIONS

This research is guided by three core questions:

RQ1: Are there noisy (mislabeled) instances in pre-published sentiment datasets for Slovene?

RQ2: Can automated methods effectively detect annotation errors in a low-resource language corpus?

RQ3: Does correcting noisy labels improve sentiment classification performance?

1.3 REPORT STRUCTURE

This report is organized as follows. Section 2 describes the corpus creation process, including data collection from Reddit. Section 3 details the annotation methodology and quality assurance measures. Section 4 presents exploratory data analysis of the annotated corpus. Section 5 covers the main modelling pipeline for Slovenian sentiment analysis, including noisy label detection and model training. Section 6 reports quantitative and qualitative results. Finally, Section 7 concludes with answers to the research questions and directions for future work.

2 SENTIMENT ANALYSIS DATASET CREATION (CORPUS CREATION)

2.1 DATA COLLECTION METHODOLOGY

The corpus creation process followed a systematic pipeline designed to collect authentic, user-generated Slovenian text from online sources. The approach prioritized the capture of informal, everyday language that reflects genuine public opinion and sentiment expression. The implementation is contained in the Corpus-Creation repository.

2.2 SOURCE SELECTION AND JUSTIFICATION

The r/Ljubljana subreddit was chosen as our main data source for several reasons. Reddit provides a rich selection of user-generated content where sentiment is expressed naturally. The r/Ljubljana community is active and predominantly Slovenian-speaking, ensuring linguistic authenticity. The topics discussed range from local events and recommendations to complaints and praise, providing a diverse range of sentiment. Unlike formal news articles or product reviews, Reddit posts capture informal discourse patterns, colloquialisms, and dialectal variations that are underrepresented in existing Slovenian NLP resources.

Data collection was performed automatically using the *reddit_scraper.py* script, which leverages the Python Reddit API Wrapper (PRAW) to access Reddit's official API. For each post, the following fields were extracted: the unique identifier, title, body text (selftext), score, URL, number of comments, and creation timestamp.

2.3 DATA CLEANING AND QUALITY CONTROL

The raw data underwent several cleaning steps to ensure quality. Language detection was implemented using the langdetect library to filter out non-Slovenian content, as the subreddit occasionally contains English or multilingual posts. Posts consisting solely of images, videos, or external links were excluded, as they lack the textual content necessary for sentiment analysis. Duplicate posts were removed based on unique Reddit post IDs. Basic text normalization removed HTML artifacts, excessive whitespace, and other noise while preserving the original linguistic content, including emojis and informal punctuation, which may carry sentiment information.

The final output was a UTF-8 encoded CSV file with structured fields, compatible with standard data analysis tools and corpus processing pipelines. The collected dataset consists of 137 posts from the r/Ljubljana subreddit, spanning the period from October 12, 2025, to November 1, 2025. For detailed code documentation, refer to the document *Documentation for Reddit Scraper Script.pdf* in the project's GitHub repository.

Output example (each row in the CSV file corresponds to one Reddit post):

```
| id | title | selftext | score | url | num_comments | created_utc |
| t3_xabcd | Ljubljanski promet | Danes spet zastoji na Celovški. | 52 | https://redd.it/xabcd | 13 | 1730462100 |
```

2.4 Example use cases

Use Case 1: Building a corpus of slovene online discourse

The dataset serves as a foundational resource for constructing a corpus of authentic, user-generated Slovene text. By focusing on informal online discussions, it captures natural language phenomena, such as slang, idiomatic expressions, and colloquial syntax, that are often missing in formal written sources.

Use Case 2: Training or testing NLP models on informal slovene text

The collected dataset is suitable for training and evaluating NLP models on informal, user-generated Slovene text. Its inclusion of colloquialisms, emojis, and non-standard punctuation provides realistic input for tasks such as text classification, tokenization, or language modeling. Models trained on this data are likely to perform better on real-world, conversational Slovenian, improving their accuracy on tasks involving online communication.

Use Case 3: Performing sentiment or topic analysis on local Reddit discussions

This dataset allows targeted sentiment and topic analysis within a local online community. Each post contains explicit or implicit expressions of opinion, praise, or complaint, making it ideal for sentiment classification, opinion mining, or identifying emerging discussion themes.

3 SENTIMENT ANALYSIS DATASET ANNOTATION

3.1 ANNOTATION GUIDELINES AND LABEL SCHEME

The sentiment annotation pipeline is implemented in the Sentiment-Annotation-Corpus repository and follows a principled approach based on established sentiment annotation practices. Our annotation scheme employs five sentiment categories to capture the nuanced nature of sentiment expression in informal online discourse. The complete annotation guidelines are documented in *Sentiment Annotation Guideline.pdf* published in the GitHub repository.

The five sentiment categories are defined as follows: (1) Negative – expresses a negative attitude, criticism, complaint, or disapproval; (2) Neutral – reports objective facts or information with no evaluative stance; (3) Positive – expresses approval, satisfaction, or positive emotion; (4) Mixed – contains multiple or conflicting sentiments, combining both positive and negative elements; and (5) Sarcastic – uses positive or neutral wording with an opposite (typically negative) implied meaning, where sentiment reversal is evident from context or tone.

3.2 ANNOTATION METHODOLOGY

The sentiment annotation process followed a two-step cognitive model. In the first step, called comprehension, the annotator reads and understands the content and communicative intent of the sentence, considering context, tone, and any pragmatic factors. In the second step, called sentiment judgment, the annotator assigns one of the five sentiment labels based on the perceived sentiment orientation. This two-step approach encourages careful reading and reduces hasty judgments based on surface-level features alone.

Annotators were guided by the question: *What kind of language is the speaker using?* This question helps focus attention on the communicative function of the text rather than just its lexical content, making it easier to spot examples such as irony or sarcasm.

3.3 INTER-ANNOTATOR AGREEMENT

To measure the consistency of sentiment annotations between annotators, we calculated Cohen's kappa coefficient, a standard statistical metric for inter-annotator agreement for categorical data. Unlike simple percent agreement, Cohen's kappa accounts for the possibility that agreement may occur by chance, providing a more reliable measure of annotation consistency.

In our dataset, **Cohen's kappa** was computed at **0.8663**. This high score confirms that annotators applied the sentiment labels consistently, supporting the quality and reliability of the annotated corpus. The Python implementation for computing this metric is available in *cohen_kappa_score.py*.

3.4 CLASH RESOLUTION PROTOCOL

Even with high inter-annotator agreement, disagreements are inevitable, especially for ambiguous or borderline cases. To handle such cases, a systematic clash resolution protocol was implemented.

When annotators assigned different labels to the same instance, the disagreement was first addressed through discussion, allowing each annotator to explain their reasoning. If the disagreement arose from unclear guidelines, the guidelines were clarified, and the instance was re-annotated. For cases of genuine ambiguity where multiple interpretations are valid, the label “Mixed” was applied to reflect the uncertainty.

3.5 SENTENCE SEGMENTATION OF THE CORPUS

The script *divide_into_sentences.py* implements a preprocessing step that converted raw textual input into sentence-level units. This step was necessary because the sentiment annotation process in this corpus operates at the sentence-level rather than on full documents or paragraphs.

The script takes raw text files as input and applies rule-based sentence boundary detection, primarily relying on punctuation markers such as periods, question marks, exclamation points, and line breaks. In the output file each row corresponds to a single sentence extracted from the original source texts.

3.6 INTENDED USE

The corpus is intended to support a range of research and practical applications in both computational and linguistic domains. It can be used to train or evaluate sentiment analysis models, providing sentence-level annotations that capture subtle variations in sentiment.

Beyond computational modeling, the corpus can be used for corpus-based linguistic and discourse studies, offering systematically annotated examples of evaluative language in informal online communication. It is also particularly valuable for research on sarcasm detection, subjectivity, and nuanced sentiment phenomena, where sentiment is implicit, mixed, or context dependent. Additionally, the corpus can serve as a benchmark dataset for assessing NLP tools in fine-grained sentiment classification, supporting consistent evaluation and comparison across methods.

4 EXPLORATORY DATA ANALYSIS (EDA) FOR ANNOTATED CORPUS

4.1 CORPUS STATISTICS

The annotated corpus comprises a substantial collection of Slovenian text suitable for sentiment analysis research. The corpus contains 137 Reddit posts, which were segmented into 612 individual sentences containing a total of 8,121 words. The data was collected over a three-week period from October 12, 2025 to November 1, 2025, capturing contemporary discourse patterns from the r/Ljubljana community.

4.2 LABEL DISTRIBUTION

The corpus shows a discrete distribution across the five sentiment categories. An analysis of the label distribution (implemented in *distribution_of_labels.py*) reveals the relative frequency of each sentiment class. The distribution reflects the natural occurrence of sentiment in informal online discourse, with neutral and negative sentiments occurring more frequently than positive or sarcastic ones.

Sentiment label	Count	Percentage
Positive	214	19.1%
Negative	360	32.2%
Neutral	448	40.1%
Mixed	71	6.4%
Sarcastic	25	2.2%

Table 1: Label distribution

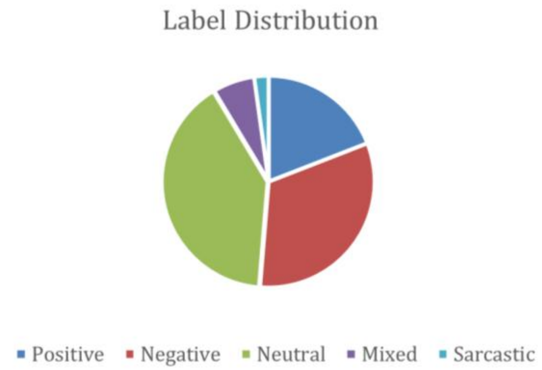

Figure 1: Label distribution

4.3 Lexical analysis

A detailed lexical analysis (implemented in *dataset_overview.py*) gives insights into the linguistic characteristics of the corpus. Title and body text were merged, lowercased, and stripped of non-alphabetic material before tokenization. Key metrics include:

- **Average sentence length:** 13.2 words per sentence
- **Average document length:** 59.3 words per document
- **Vocabulary size:** 3,143 unique tokens
- **Unique token coverage:** 0.555 (type-token ratio)
- **Number of documents:** 137 posts (all from the same subreddit r/Ljubljana)

The **most frequent content words** include: "*zanima*" (English: interested), "*hvala*" (English: thanks), "*lahko*" (English: can), "*ima*" (English: has), and "*res*" (English: really), which is consistent with the type of content found on the subreddit. An analysis of sentence length by sentiment category shows little variation:

Sentiment category	Average sentence length (in words)
Negative	13.9
Neutral	12.9
Positive	11.8
Mixed	13.8

Sarcastic	13.8

Table 2: Sentence length by sentiment category

5 MODELLING THE SLOVENIAN SENTIMENT ANALYSIS

5.1 METHODOLOGY OVERVIEW

The modelling pipeline for Slovenian sentiment analysis followed a comprehensive approach designed to identify and correct mislabeled instances in an existing sentiment dataset, ultimately improving classifier performance. The work was conducted on the **KKS Opinion Corpus**, a publicly available sentiment-annotated collection of Slovene web commentaries compiled from Slovene news portals. The implementation is contained in the Modelling the Slovenian sentiment analysis repository.

The methodology follows this sequential pipeline:

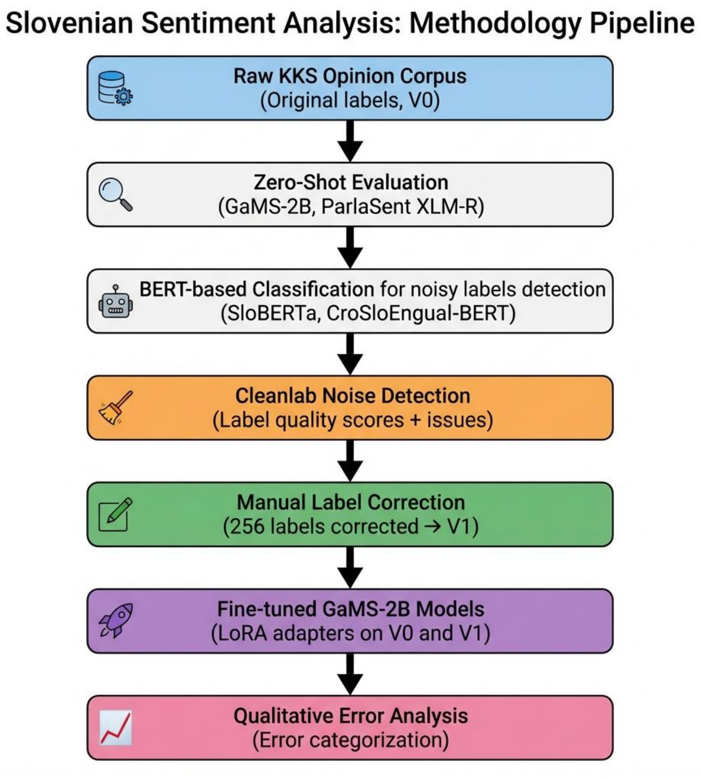
Slovenian Sentiment Analysis: Methodology Pipeline
Raw KKS Opinion Corpus
(Original labels, V0)
Zero-Shot Evaluation
(GaMS-2B, ParlaSent XLM-R)
BERT-based Classification for noisy labels detection
(SloBERTa, CroSloEngual-BERT)
Cleanlab Noise Detection
(Label quality scores + issues)
Manual Label Correction
(256 labels corrected → V1)
Fine-tuned GaMS-2B Models
(LoRA adapters on V0 and V1)
Qualitative Error Analysis
(Error categorization)

Figure 2: Methodology flowchart

5.2 ZERO-SHOT EVALUATION

Zero-shot evaluation was used to assess model generalization without task-specific training, providing baseline performance measures. Two approaches were evaluated:

- **GaMS-2B-Instruct** (*zero_shot_performance_gams.py*): A decoder-only causal language model with approximately 2-3 billion parameters, pretrained on Slovene-centric corpora and instruction-tuned to follow natural language prompts. GaMS-2B is based on Google's Gemma 2-2B architecture and was continually pretrained on approximately 13.6 billion tokens covering Slovenian and related languages.
- **ParlaSent XLM-RoBERTa** (*zero_shot_performance_parlasent.py*): An encoder-only transformer fine-tuned for sentiment classification on parliamentary and political text. This model provides a complementary perspective as it was trained on formal discourse rather than informal social media content.

5.3 BERT-BASED CLASSIFICATION

Two BERT-based classifiers were used to identify potentially noisy labels by comparing high-confidence model predictions with the original annotations:

- **SloBERTa** (*classifier_sloberta.py*): A monolingual Slovenian RoBERTa model pretrained on large Slovenian corpora. It provides contextual token representations specifically optimized for the Slovenian language, capturing morphological and syntactic patterns unique to Slovene.
- **CroSloEngual-BERT** (*classifier_crosloengual-bert.py*): A multilingual BERT-base architecture pretrained jointly on Croatian, Slovenian, and English. This regionally-focused multilingual model leverages cross-linguistic transfer while maintaining strong performance on South Slavic languages.

Result files (*sloberta_results.ods, crosloengual-bert_results.ods*) capture all the instances flagged as potentially mislabeled by each classifier. The corresponding files with human corrected labels (*sloberta_results_corrected_labels.ods* and *crosloengual-bert_results_corrected_labels.ods)* contain instances where the model prediction did not match the original annotations.

The *compare_bert_results.py* script was used to analyse both models to identify the intersection of detected label errors. Results show that CroSloEngual-BERT detected 679 potentially noisy instances, SloBERTa detected 657, with an intersection of 349 instances flagged by both models. This intersection represents high-confidence error candidates for manual review.

5.4 CLEANLAB-BASED NOISY LABELS DETECTION

To further support systematic detection of label noise, we integrated the cleanlab library into the pipeline. Cleanlab uses confident learning algorithms to identify instances where the assigned label is likely incorrect based on model predictions. Both BERT-based classifiers (SloBERTa and CroSloEngual-BERT) were used as input models for this process.

Label noise was identified using the cleanlab library by combining cross-validated model predictions with label quality estimates. Stratified 5-fold cross-validation was used to obtain out-of-sample predicted probabilities for all instances, ensuring that predictions were not influenced by exposure to the same data during training. Based on these predictions, a label quality score was assigned to each instance, ranging from 0 (likely incorrect) to 1 (likely correct). Instances with low label quality and high model confidence were then flagged as potential labeling errors.

For automatic label correction, we applied the following rule: if the model's predicted probability for an alternative class exceeds 0.8 (high confidence) and the label quality score is below 0.4 (low quality), the label is flagged for automatic correction. This conservative threshold ensures that only high-confidence corrections are applied automatically, while borderline cases are reserved for manual human review.

5.4.1 Human review process

It is important to distinguish between detected noisy instances (candidates flagged by cleanlab) and confirmed real errors (verified by human review). Cleanlab flagged 679 instances from CroSloEngual-BERT and 657 from SloBERTa as potentially mislabeled, with 349 instances flagged by both models. However, not all flagged instances are actual errors – many are borderline cases or incorrect model predictions.

All flagged instances were exported to spreadsheet files (*sloberta_results.ods* and *crosloengual-bert_results.ods*) for manual review. Human annotators reviewed each flagged instance, comparing the original label with the model's prediction and the text content. Analysis of the overlap between models revealed that 349 instances were flagged as potentially noisy by both classifiers. However, within this intersection of 349 flagged cases, only 28 were confirmed as real errors by human reviewers (**8.0% precision for the intersection**). The remaining 321 cases in the intersection were determined to be correct original labels or ambiguous instances where the models' disagreement was not indicative of an error.

After human review, the precision of noise detection was measured (what percentage of flagged instances were actually mislabeled):

Model	Flagged by Cleanlab	Confirmed errors	Precision
CroSloEngual-BERT	679	237	34.9%
SloBERTa	657	57	8.7%

Table 3: Precision of noise detection

5.4.2 Creating the corrected dataset

The *prepare_datasets.py* script merged the human corrections from both models to create the final corrected dataset. The merging logic prioritized human-corrected labels: if a correction existed from either SloBERTa or CroSloEngual-BERT review, it was used; otherwise, the original label was kept. Combining corrections from both models (237 from CroSloEngual-BERT + 57 from SloBERTa - 28 overlap) yielded approximately 266 unique corrections. The final V1 dataset contains 256 label changes, reflecting some merging and deduplication in the process.

The corrected dataset (V1) contains 256 label corrections out of 4,777 total instances (5.4%). The corrections are distributed as follows:

Original label → Corrected	**Count**
positive → negative	79
positive → neutral	23
negative → positive	12
negative → neutral	22
neutral → positive	22
neutral → negative	98
Total	**256**

Table 4: Label corrections

The most common correction was neutral → negative (98 instances), suggesting that annotators may have been overly conservative in assigning negative labels, particularly for texts with subtle criticism or implicit negative sentiment. The second most common was positive → negative (79 instances), often involving sarcastic or ironic content where surface-level positive words masked underlying negative intent.

5.4.3 Label distribution

The KKS Opinion Corpus underwent a systematic correction process resulting in two versions: V0 (original labels) and V1 (corrected labels). The following table shows the distribution of sentiment labels across both versions, illustrating the net impact of the correction pipeline.

Sentiment	**V0 (Original)**	**V1 (Corrected)**	**Net Change**
Negative	3,291	3,434	+143
Positive	898	830	-68
Neutral	588	513	-75

Table 5: Distribution of sentiment labels in original and corrected datasets

The Grouped Bar Chart shows a direct side-by-side comparison of the total counts for each sentiment class (Negative, Positive, Neutral) in V0 vs V1. The chart shows a clear increase in the Negative class (+143 instances), while Positive and Neutral both decreased.

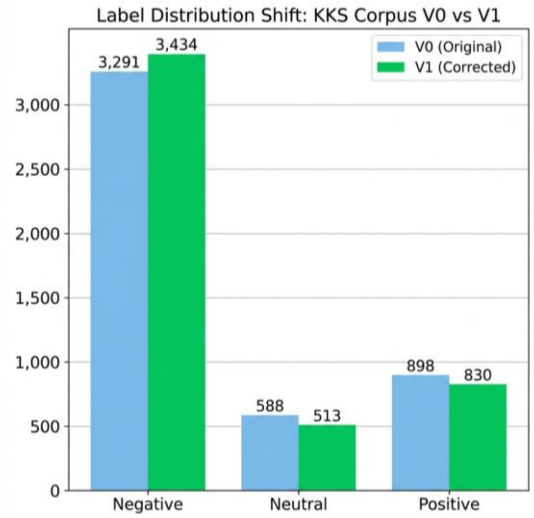

Figure 3: Comparison of sentiment labels in original and corrected datasets

Sankey Diagram shows the "migration" or flow of specific labels. It connects the original labels V0 (left) to their corrected versions V1 (right) using proportional bands. The diagram reveals exactly where the original annotators were most inconsistent. The largest shifts occurred from neutral → negative (98 cases) and positive → negative (79 cases, mostly missed sarcasm).

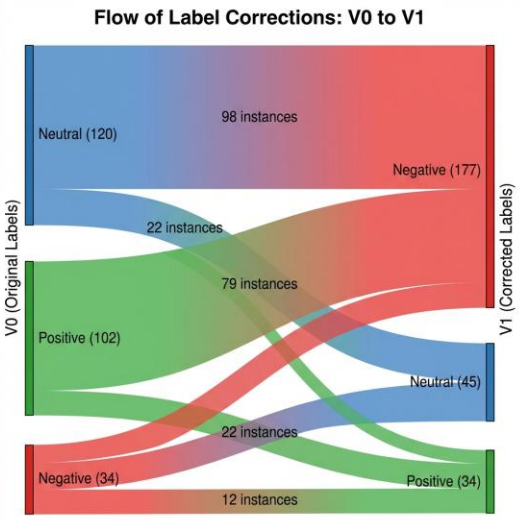

Figure 4: Migration of specific labels in original and corrected datasets

5.5 GaMS fine-tuning

GaMS-2B model: GaMS-2B (available at cjvt/GaMS-2B on Hugging Face) is a continually pretrained variant of Google's Gemma 2-2B model with adaptations for Slovene and related languages.

The publicly listed specifications include the following concrete model details from its Hugging Face page:

- Parameters and architecture:
  - Parameter count: 3 billion (3 B) total parameters.
  - Base architecture: Gemma 2-2B (Google's decoder-only model).
  - Model type: Gemma2ForCausalLM / gemma2 family.
- Model files, precision, and size:
  - Tensor type (full weights): F32 (32-bit floats) in the statically hosted safetensors.
  - Reported model size in HF metadata: 3 B params (file summary).
  - Quantized variants (from community GGUF builds): ~2.61 B params in some quant formats, with sizes ranging ~1.2–5.2 GB depending on precision.
- Context & tokenization:
  - Context/window length (for related instruct variant): 8192 tokens supported.
  - Tokenizer class: GemmaTokenizer (inherits Gemma2 tokenization).
- Training data and pretraining: continually pretrained on a mix of parallel corpora and monolingual corpora covering the listed languages, totaling ~13.6 billion tokens in second-stage pretraining.

Base pretraining of GaMS-2B provides general linguistic competence, but downstream usability depends on alignment via **instruction fine-tuning (IFT)**, which adapts a pretrained model to follow human instructions using ⟨instruction, input, output⟩ triples. Building on the GaMS-2B base model, we performed instruction fine-tuning (IFT) using **LoRA (low-rank adaptation)** to create sentiment classifiers. Two versions were trained:

- **GaMS fine-tuned on original KKS dataset (V0):** Trained on the original labels as published, this model serves as the baseline for evaluating the impact of label correction. Script: *gams_finetune_original_KKS_dataset.py*
- **GaMS fine-tuned on corrected KKS dataset (V1):** Trained on the manually corrected labels after noise removal, this model demonstrates the performance gains from cleaner training data. Script: *gams_finetune_corrected_KKS_dataset.py*

5.6 QUALITATIVE ERROR ANALYSIS

The *qualitative_error_analysis.py* script performed in-depth analysis of misclassified instances to identify systematic patterns in model errors. Analysis of 148 misclassified instances from V0 and 129 from V1 revealed several distinct error categories:

- **Irony/Sarcasm:** Surface sentiment contradicts intended meaning. Example: *Wauu, za 50 litrov bom pridobil celih 30 centov. Hvala! :)*[1]
- **Mixed sentiment:** Multiple conflicting signals in one text. Example: *Borili so se. ... Jaz sem z veseljem gledal... Drugi polčas malo slabša igra.*[2]
- **Domain-specific:** Requires contextual knowledge. Example: *Če hočemo, da gre cena še bolj dol, bi moral sedaj tečaj €/$ narasti...*[3]
- **Colloquial/Slang:** Informal expressions. Example: *fak stari.... to pa je pravljica.....*[4]
- **Short/Ambiguous:** Insufficient context. Example: *Megla bo...*[5]
- **Complex syntax:** Negation and conditionals. Example: *Nepremičninski pok se nikoli ne bo zgodil*[6]

[1] English: *Wow, I'll get a whole 30 cents for 50 liters. Thank you! :)*

[2] English: *They fought. ... I enjoyed watching... The second half was a little worse.*

[3] English: *If we want the price to go down even further, the €/$ exchange rate would have to rise now...*

[4] English: *damn dude.... this is a fairy tale.....*

[5] English: *It will be foggy...*

[6] English: *The real estate boom will never happen*

6 RESULTS AND EVALUATION

6.1 QUANTITATIVE RESULTS

The evaluation employed standard classification metrics: **accuracy** (proportion of correct predictions) and **macro-averaged F1-score** (harmonic mean of precision and recall, averaged across classes). Both metrics were computed using 5-fold stratified cross-validation to ensure robust estimates.

6.2 MODEL COMPARISON

This section compares the performance of different models across multiple experimental configurations. Evaluations are conducted on two versions of the KKS dataset: V0, which represents the original dataset, and V1, which contains corrected labels. The comparison focuses on accuracy and macro-averaged F1 scores to assess both overall performance and class-balanced effectiveness across models.

	Model	**Accuracy**	**Macro-F1**	**Dataset**
1	GaMS-2B (zero-shot)	24.37%	17.08%	V0
2	XLM-R-Parlasent (zero-shot)	68.83%	27.18%	V0
3	SloBERTa	77.75%	62.66%	V0
4	CroSloEngual-BERT	82.21%	70.97%	V0
5	GaMS-2B (fine-tuned V0)	84.52%	75.43%	V0
6	**GaMS-2B (fine-tuned V1)**	**86.51%**	**77.02%**	**V1**

Table 6: Model performance across different configurations

6.3 PERFORMANCE ANALYSIS OF INDIVIDUAL MODELS

6.3.1 GaMS-2B (zero-shot)

The zero-shot GaMS 2B model performs poorly on both metrics, with low macro-F1 (0.17) and accuracy (0.24). The low accuracy shows that the model struggles even to predict the most common class reliably, which points to weak zero-shot transfer and poor alignment with the GaMS label space. Overall, the results suggest the model has minimal task understanding in a zero-shot setting.

6.3.2 XLM-R-Parlasent (zero-shot)

The model shows an imbalance between metrics: while accuracy is relatively high (0.69), the macro-F1 is low (0.27). This indicates that the model largely relies on majority-class predictions and fails to capture minority classes, making the accuracy look misleading. The result suggests limited usefulness of zero-shot transfer for Parlasent without domain adaptation or supervised fine-tuning.

6.3.3 SloBERTa

SloBERTa shows a clear improvement over zero-shot multilingual models, achieving high accuracy (0.78) and a substantially higher macro-F1 score (0.63). The reduced gap between accuracy and macro-F1 indicates more balanced performance across classes, suggesting that the model captures minority categories more effectively. As a Slovene-specific pretrained model, SloBERTa benefits from strong language and domain alignment, which helps it distinguish classes well even without task-specific fine-tuning. The results demonstrate that language-specific pretraining provides a significant advantage over generic multilingual models.

6.3.4 CroSloEngual-BERT

CroSloEngual-BERT achieves the best performance among non–fine-tuned models, with accuracy of 0.82 and macro-F1 of 0.71. The relatively small difference between the two metrics indicates stable and balanced performance across all classes. Compared to SloBERTa, the higher macro-F1 suggests it handles minority classes even better, making CroSloEngual-BERT the strongest pretrained baseline prior to supervised fine-tuning.

6.3.5 GaMS-2B (fine-tuned V0)

After fine-tuning, GaMS-2B reaches 0.85 accuracy, while macro-F1 is slightly lower at 0.75. This shows that the model is very reliable on the dominant categories but less consistent across all classes, reflecting the class imbalance in the original IFT KKS dataset.

6.3.6 GaMS-2B (fine-tuned V1)

The fine-tuned V1 model achieves a macro-F1 of 0.77, showing more balanced performance across all classes, and accuracy of 0.87, indicating overall improvement. In comparison to original IFT results, both metrics increased, with macro-F1 seeing a notable rise. This confirms that correcting the IFT KKS dataset improved not only performance on the majority class but also across the full label set, giving the model a cleaner learning signal.

6.4 DATASET CORRECTION IMPACT

The results clearly show that correcting the labels leads to better model performance. Comparing GaMS fine-tuned on V0 (original labels) versus V1 (corrected labels), we see an **increase of +3 percentage points in accuracy** (0.84 → 0.87) and **+2 percentage points in macro-F1** (0.75 → 0.77). This improvement is particularly notable given that only 5.4% of labels were corrected, suggesting that even a small proportion of noisy labels can have a measurable impact on model performance.

The trained models and datasets are publicly available on HuggingFace:

- GaMS-2B finetuned on original KKS dataset: https://huggingface.co/lea-vodopivec7/gams-2b-finetuned-kks-V0
- GaMS-2B finetuned on corrected KKS dataset: https://huggingface.co/lea-vodopivec7/gams-2b-finetuned-kks-V1

7 CONCLUSION

7.1 RESEARCH QUESTION OUTCOMES

RQ1: Are there noisy (mislabeled) instances in pre-published sentiment datasets for Slovene?
Yes. The findings clearly confirm the presence of noisy labels in pre-published Slovenian sentiment datasets. Through a combination of automated detection and systematic human verification, we identified a substantial number of mislabeled instances. Specifically, after merging corrections from both SloBERTa and CroSloEngual-BERT review processes, the final corrected dataset (V1) contained 256 confirmed label corrections out of 4,777 total instances, corresponding to 5.4% of the dataset.

RQ2: Can automated methods effectively detect annotation errors in a low-resource language corpus?
The results show that automated methods can successfully support noise detection, but their effectiveness depends heavily on human verification. Cleanlab-based detection combined with BERT-based classifiers flagged a large number of potential errors: 679 instances by CroSloEngual-BERT and 657 by SloBERTa, with an overlap of 349 cases detected by both models.

However, human review revealed varying precision across models. CroSloEngual-BERT achieved a precision of 34.9%, while SloBERTa achieved 8.7% precision, indicating that although many instances were flagged, only a subset were genuine errors. Notably, within the 349-instance intersection, only 28 cases (8.0%) were confirmed as real errors.

RQ3: Does correcting noisy labels improve sentiment classification performance?
Yes. Models trained on the corrected dataset (V1) showed consistent performance improvements over those trained on original labels (V0). The GaMS-2B model improved by +3 percentage points in both accuracy and macro-F1 after training on

corrected labels. This demonstrates that label quality directly impacts model quality, and that investing in dataset cleaning yields measurable returns.

7.2 KEY FINDINGS

This project makes several contributions to Slovenian NLP and sentiment analysis research. The analysis shows empirical evidence that pre-published datasets contain meaningful amounts of label noise (approximately 5%) that affects downstream model performance. The results further demonstrate that a combination of automated detection methods can effectively identify noisy labels for human review. In addition, publicly available corrected datasets and fine-tuned models are provided to support future research. Finally, the qualitative error analysis highlights specific challenges for Slovenian sentiment analysis, including irony detection, colloquial language processing, and domain-specific sentiment expressions.

7.3 LIMITATIONS

The evaluation of noisy-label detection in this study is limited to a single target dataset, namely the KKS Opinion Corpus, which constrains the generalizability of the findings. The effectiveness of the proposed approach is closely tied to the corpus's original annotation scheme, class distribution, and domain characteristics, as the dataset consists of sentiment annotations for news commentaries; consequently, the observed results may not directly transfer to other Slovenian sentiment datasets, domains, or label taxonomies. Moreover, the automated noise detection methods employed in this pipeline exhibited low to moderate precision and required conservative confidence thresholds combined with extensive human verification, which limits the scalability of the approach in settings where expert annotators are not available. Persistent challenges further arise from instances of sarcasm and mixed sentiment, which remain difficult for both models and human annotators to interpret consistently, thereby introducing an element of residual subjectivity that cannot be fully resolved through technical means alone. Finally, although label correction led to consistent improvements in downstream classification performance, these gains are moderate in magnitude and closely linked to the specific models and training configurations used in this study; alternative model architectures or application domains may therefore exhibit different degrees of sensitivity to label noise.

7.4 FUTURE WORK

Several directions remain for future investigation. Expanding the dataset with additional sources such as news articles, social media posts, and reviews would expand the coverage of Slovenian sentiment analysis. Developing specialized models for irony and sarcasm detection could help address one of the most common error categories identified in the analysis. Active learning approaches also offer a promising direction, as they could make the noise detection process more interactive and efficient. Finally, cross-lingual transfer from related South Slavic languages may provide additional training signal for low-resource Slovenian NLP tasks.