Sentiment Analysis and Detoxification using Agentic Workflow Milestone 2

Kartik Sirwani
University of British Columbia
kartiksirwani@gmail.com

Muhammad Mujtaba Khan
University of British Columbia
mujtaba.41.mm@gmail.com

Abstract

This document extends milestone 1 tasks of sentiment and toxicity detection by adding the task of detoxification of toxic texts. We also add a module that translates African languages to english, before doing the three tasks mentioned above. Finally, we combine all the components efficiently in an agentic workflow by making use of the LangGraph library. We also present the evaluation results of our pipeline along with other findings.

1 Models Used

1.1 Overview

As our dataset consisted of multilingual texts (African and English languages), we developed three models:

- Translation Model:- Converts African text to english.
- Sentiment Detection:- Detects the sentiment of the text and provides an explanation of the label.
- Toxicity Detection:- Detects if a text is toxic or not and provides an explanation of the label. Along with this in this milestone, we use this model to detoxify the text.

1.2 Language Detection and Translation

- **Language Detection:** Our dataset consisted of only African and English texts. We used langid library which is available as a python package to detect if the given text is in English or in an African language.
- **Language Translation:** We tried two models **"UBC-NLP/cheetah-1.2B"** and **"UBC-NLP/toucan-base"**. We found that the **"UBC-NLP/toucan-base"** was more stable in giving the translation as we tried different prompts in comparison to the **"1.2B"** model. So we used the **"UBC-NLP/toucan-base"** in our final pipeline. The final model translates African sentences into English. This is a sample prompt to the model: **"eng: Nashukuru huduma ya haraka"**. Output is the translated sentence in English.

1.3 Sentiment Analysis

We used the same model (**"ibm-granite/granite-3.0-2b-instruct"**) to do sentiment detection and provide an explanation as it gave satisfactory results (f1-score of 0.84). Along with the label of sentiment, it also provides the explanation on why the model thinks sentiment is positive, negative or mixed. Prompt used is shown below:

```
template=(
    "You are an expert sentiment analysis assistant. "
    "Classify the given text as 'positive', 'negative', or 'mixed'. "
    "Provide your response in the following JSON format:\n\n"
    "{{\n"
    '  "sentiment": "<positive/negative/mixed>",\n'
    '  "explanation": "<Brief explanation of sentiment>"\n'
    "}}\n\n"
    "Analyze the sentiment of the following sentence: {text}"
)
```

Figure 1: Sentiment Detection Prompt

1.4 Toxicity Detection and Detoxification

For detecting if a text is toxic, we tried several quantized models because our milestone1 model(**"UBC-NLP/DetoxLLM-7B"**) was taking longer than 1 minute to classify if a text is toxic or not. However even after switching to these quantized models, we did not see any improvement in the inference time. Hence we switched back to **"UBC-NLP/DetoxLLM-7B"**. This model detects if the text is toxic or not and also provides explanation of why it thinks the text is toxic. Prompt used is shown below:

```
template=(
    "You are an expert Toxicity Classifier. Classify the text as toxic or not-toxic and also provide explanation.\n"
    "Here are a few examples of your task:\n\n"

    "Sentence: \"You are completely clueless.\"\n"
    "Toxicity Label: toxic\n"
    "Explanation: Insulting language directed at the recipient.\n\n"

    "Sentence: \"go blow your bitch GabrielF =\"\n"
    "Toxicity Label: toxic\n"
    "Explanation: Offensive language like bitch used\n\n"

    "Sentence: \"Kill yourself! Fuckin' bastard!\"\n"
    "Toxicity Label: toxic\n"
    "Explanation: Offensive language and personal attack aimed at recipient's life.\n\n"

    "Sentence: \"No, you're being unwise.\"\n"
    "Toxicity Label: non-toxic\n"
    "Explanation: This input text is not toxic.\n\n"

    "Sentence: \"{text}\""
)
```

Figure 2: Toxicity Detection Prompt

Similarly we used the same model to detoxify the toxic text using the prompt:

```
detoxify_prompt = PromptTemplate(
    input_variables=["text"],
    template="Detoxify the following toxic sentence:\n{text}\nNon-toxic version:"
)
```

Figure 3: Detoxify Prompt

2 Agentic Workflow:

We used langgraph library to build our pipeline and integrate the models defined in the previous section. The visual representation of our generated workflow is shown on the right. The workflow is as follows:

- We start at the start node.
- From the start node we go to the language detection node. Which detects if the given text is in English or African language.
- If the text is not english we will go to the translate node which will translate the text to english before proceeding to next node. If the text is english, we directly go to the next node.
- The next step or node is sentiment detection and providing explanation.
- Next step is to detect if the text is toxic along with explanation.
- Finally if the text is not toxic we proceed to the end step or node and our workflow completes. If the text is toxic we first go to detoxification node which detoxifies the text and proceeds to the end node.
- The workflow outputs **sentiment label, explanation of sentiment label, toxicity label, explanation of toxicity label, detoxified text** if the text was toxic.
- We pass both the datasets: **"multi lingual sentiment dataset"** and **"toxicity analysis dataset"** through this workflow.

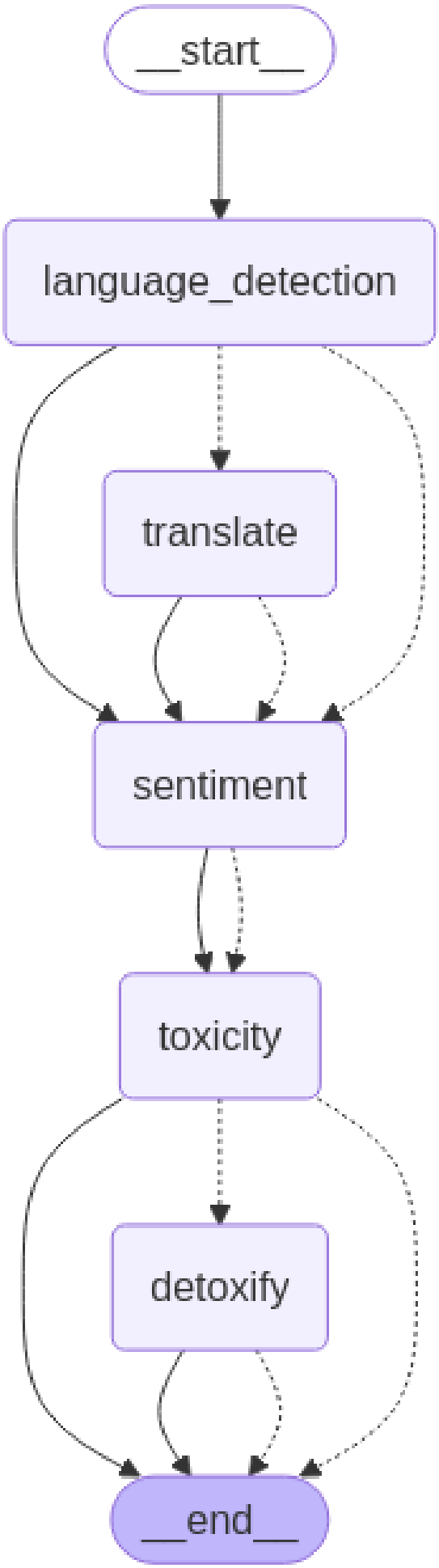

Figure 4: Workflow graph

3 Evaluation

In this section, we present the evaluation results for the three tasks in Milestone 2: **Task 1: Sentiment Classification**, **Task 2: Toxicity Detection**, and **Task 3: Detoxification**. We compare the results of Task 1 and Task 2 with Milestone 1 and provide a detailed evaluation of Task 3, which was introduced in Milestone 2.

3.1 Task 1: Sentiment Classification

For sentiment classification, we used the same model as in Milestone 1 (**"ibm-granite/granite-3.0-2b-instruct"**). However, the addition of the translation module for multilingual texts impacted the performance. The evaluation metrics for sentiment classification are as follows:

Metric	Milestone 1	Milestone 2
Accuracy	0.84	0.69
Precision	0.87	0.73
Recall	0.84	0.69
F1 Score	0.84	0.69

Table 1: Comparison of Sentiment Classification Metrics between Milestone 1 and Milestone 2

The decrease in performance can be attributed to the translation step, which may not have preserved the sentiment nuances when converting African languages to English.

3.2 Task 2: Toxicity Detection

For toxicity detection, we continued using the **"UBC-NLP/DetoxLLM-7B"** model due to its ability to provide explanations for its predictions. However, the performance metrics dropped compared to Milestone 1. The evaluation metrics for toxicity detection are as follows:

Metric	Milestone 1	Milestone 2
Accuracy	1.00	0.61
Precision	1.00	0.60
Recall	1.00	0.61
F1 Score	1.00	0.60

Table 2: Comparison of Toxicity Detection Metrics between Milestone 1 and Milestone 2

The drop in performance may be due to the increased complexity of the task, as the model now handles multilingual inputs.

3.3 Task 3: Detoxification

Task 3, detoxification, was introduced in Milestone 2. We used the same model as in Task 2 (**"UBC-NLP/DetoxLLM-7B"**) to detoxify toxic texts. Since this task involves subjective evaluation, we conducted a human evaluation by annotating 15 samples. Two annotators evaluated the detoxified outputs, and we calculated the following metrics:

Metric	Value
Cohen's Kappa Score	0.25
Krippendorff's Alpha	0.72
Average Score of Detoxification	7.07

Table 3: Human Evaluation Results for Detoxification in Milestone 2

The results indicate that the detoxification process is moderately effective but leaves room for improvement as the average score is 7 out of 10. The low Cohen's Kappa Score (0.25) suggests that detoxification is a subjective task, and annotators had difficulty agreeing on the quality of the outputs. However, the higher Krippendorff's Alpha score (0.72) indicates better agreement when considering interval distances, suggesting that while absolute agreement is low, the relative differences in scores are more consistent.

4 Challenges

During Milestone 2, we encountered several challenges that impacted our workflow and results across all three tasks:

- **Inference Time**: The toxicity detection and detoxification model (**"UBC-NLP/DetoxLLM-7B"**) took more than 2 minutes per sample, which significantly slowed down the pipeline. We experimented with quantized models but did not observe any improvement in inference time.
- **GPU Limitations**: We were restricted by Google Colab's GPU limits, which further exacerbated the inference time issue and limited our ability to experiment with larger models and different prompts.
- **Translation Efficiency**: The translation step from African languages to English may not have been efficient, leading to a drop in sentiment classification performance. This

suggests that the translation model (**"UBC-NLP/toucan-base"**) may not fully preserve sentiment nuances.

- **Accuracy Drop**: Both sentiment classification and toxicity detection saw a drop in accuracy compared to Milestone 1. This could be due to the added complexity of handling multilingual inputs.
- **Annotator Agreement**: The low Cohen's Kappa Score (0.246) for detoxification evaluation indicates that human annotators had difficulty agreeing on the quality of detoxified outputs. This highlights the subjective nature of detoxification and potential personal biases. But the interval differences are lower suggesting there is some consistency between range of scores being assigned to each text.

References

[1] LLM Prompting Tutorial. Available at: GitHub UBC - LLM Prompting, Accessed: 09-Mar-2025.

[2] LangChain Tutorial - Part 4: Loading Tools and Chains. Available at: GitHub UBC - LangChain Tutorial, Accessed: 09-Mar-2025.

[3] LangChain Tutorial - Part 5: Agents. Available at: GitHub UBC - LangChain Agents, Accessed: 09-Mar-2025.

[4] A Coding Guide to Sentiment Analysis of Customer Reviews Using IBM's Open-Source AI Model Granite 3B and Hugging Face Transformers. Available at: MarkTechPost (2025), Accessed: 09-Mar-2025.

[5] Pandas Documentation. Available at: Pandas Official Site, Accessed: 09-Mar-2025.

[6] Scikit-Learn Documentation. Available at: Scikit-Learn Official Site, Accessed: 09-Mar-2025.

[7] DetoxLLM-7B-i1-GGUF Model. Available at: Hugging Face - DetoxLLM, Accessed: 09-Mar-2025.

[8] Toucan-Base Model. Available at: Hugging Face - Toucan-Base, Accessed: 09-Mar-2025.

[9] Loading Quantized DetoxLLM Tutorial. Available at: GitHub UBC - Load Quantized DetoxLLM, Accessed: 09-Mar-2025.

[10] Hugging Face Transformers Pipelines Documentation. Available at: Hugging Face - Pipelines, Accessed: 09-Mar-2025.