Final Project:
LLM-Based Workflow for Sentiment Analysis, Toxicity Detection, and Toxic Style Transfer

COLX 565
Instructor: Dr. Muhammad Abdul-Mageed

March 10, 2025

1 General

- This assignment description is subject to slight changes, e.g., to provide clarifications and answer any questions, with notification.
- **Please take your time reading this assignment carefully, ensuring you understand it clearly. If there is any part that is not clear to you, please make sure you ask the instructor.**
- **Group Assignment** This is a group assignment. Students will form **groups** of ***2 students*** each. Smaller or larger numbers are **not allowed** without consulting the instructor. If you cannot find a group member, please consult with the instructor *timely*. You must work with the same team member for both milestone one and milestone two of this assignment. Changes are not allowed. Teamwork is an exercise of our ability to collaborate. It is also intended to provide a *layer of support*. You are encouraged to be supportive and kind to others.

2 Overview

In this assignment, you will apply NLP and ML methods to create an LLM-based framework for sentiment analysis and a number of related tasks. The framework can be viewed as an agentic workflow that handles the following tasks:

1. **Sentiment Analysis**
   - Labels: *positive, negative, neutral, mixed*
   - Must include an *explanation* for each predicted label
2. **Toxicity Detection**
   - Labels: *toxic, non-toxic*
   - Must include an *explanation* for each predicted label

3. **Toxic-to-Non-Toxic Style Transfer**
    - Rewrite toxic text into non-toxic equivalents

Furthermore, to support *non-English* input data, you must integrate a **translation step** (by using a translation model as part of the framework) that converts any given text into English before applying the above tasks.

3 Background and Resources

- **LangChain**: LangChain is a popular framework for developing LLM. You will be provided some code introducing you to LangChain and you are expected to build on this code to develop solutions for the current assignment. Clearly, one objective of this assignment is to provide you with a context to build LLM agentic workflows exploiting LangChain.
- **Other Resources**:

  You will use the following LLMs in this assignment. They are selected to be on the smaller side so that you do not have challenges with GPUs. They all can run on Google Colab and a personal machine with GPUs.

  - DetoxLLM Model: An end-to-end detoxification framework. It also introduces explanation to promote transparency and trustworthiness.
  - **Toucan Models:** toucan-base and toucan-1.2B. These are many-to-many translation models for 150 African language pairs covering 46 languages.
  - **NLLB Models:** nllb-200-distilled-600M and nllb-200-1.3B. The NLLB-200 models are machine translation models intended for research in machine translation, - especially for low-resource languages. They cover 200 languages.
  - **Granite Model:** granite-3.0-2b-instruct. This model is trained using a diverse set of techniques with a structured chat format, including supervised finetuning, model alignment using reinforcement learning, and model merging. It handles sentiment analysis. It supports 12 languages: English, German, Spanish, French, Japanese, Portuguese, Arabic, Czech, Italian, Korean, Dutch, and Chinese.
- **Labs**: You are encouraged to attend lab sessions for questions and support related to code shared with you that you may like to use for this assignment.

4 Objectives

By completing this assignment, you will:

- *Build agentic NLP workflows* using LLMs.
- *Practice* chaining multiple sub-tasks (translation, label prediction, explanation generation) into a cohesive system.
- *Develop* a solution for *style transfer* to rewrite toxic content into non-toxic text.

- *Practice* applying and evaluating these models on both English and non-English data.
- *Practice* writing up a report describing an engineering project involving LLM-based agentic workflows.

5 Milestone 1 (Week 3)

5.1 Tasks

1. **Framework Setup**
   - Develop a preliminary system to handle:
     - **Task 1**: Sentiment Analysis (with explanations)
     - **Task 2**: Toxicity Detection (with explanations)
   - Use *simple sequential chains* (or an equivalent straightforward approach) for these tasks.
2. **Datasets & Evaluation**
   - Two *English-language* **test only** datasets will be provided, one for sentiment and another for toxicity. (Links to datasets will be announced).
   - **You are not required to finetune the models and so there are no training data provided to you.** The models suggested to you are chat models that have already been finetuned on the different tasks. You are required to evaluate them in **zero-shot** setting, thus needing no training data points or **few-shot** setting (which needs only a handful of data points). Note that there are plenty of sentiment datasets on HuggingFace (see here), for example, and the training data for DetoxLLM is available here.
   - Demonstrate how your framework processes each input and outputs a label and a concise explanation. Report results in terms of all the following metrics: *accuracy, precision, recall*, and F_1 **for each task**. (Note that the main metric for how good your system is will be F_1 for the case of sentiment analysis but otherwise it is accuracy. Also, you are not required to evaluate the goodness of explanations automatically. Please look at 10 samples from the explanations manually and descibe how good you are satisfied with their quality.
3. **Deliverables (Milestone 1)**
   - **A short report (2–3 pages)** describing:
     - Your overall **approach** (e.g., any pipeline architecture, model selection).
     - How you **integrated the sentiment and toxicity detection tasks**.
     - **Implementation details** (such as libraries used, environment setup, or relevant code snippets).
     - **Evaluation** methods and any preliminary **results** or observations.
     - Any **challenges** encountered or **limitations** of your current approach.
   - Working code (scripts, notebooks, etc.) that runs *end-to-end* on the provided datasets.

5.2 Example Inputs/Outputs for Milestone 1

Task 1: Sentiment Analysis (with Explanation) **Input**: "I love the new features of this product, but sometimes it crashes."
Output (example):

```
{
  "sentiment_label": "mixed",
  "explanation": "Positive about new features, negative about crashes."
}
```

Task 2: Toxicity Detection (with Explanation) **Input**: "You are completely clueless!"
Output (example):

```
{
  "toxicity_label": "toxic",
  "explanation": "Insulting language directed at the recipient."
}
```

5.3 Grading Rubric for Milestone 1 (24 points)

Component	Criteria	Points
Framework Design	Clarity, structure, and correct integration of sentiment and toxicity tasks in a simple chain	5
Model Performance	Accuracy of sentiment/toxicity classification on the provided dataset	4
Explanations	Depth and clarity of explanations for each classification	4
Written Report	Overall thoroughness of approach description, analysis, and discussion of results/limitations	6
Code Quality	Readability, adherence to best practices, reproducibility	5

6 Milestone 2 (Week 4)

6.1 Overview

Note: You cannot use API calls or LLMs larger than 7-8B parameters for this assignment. Make sure you employ an agentic workflow.

Milestone 2 is an extension of Milestone 1. This means that your code must still **Task 1** (sentiment) and **Task 2** (toxicity) in addition to the new task, **Task 3** (detoxification). In addition to the work you carried out in Milestone 1, there are two main aspects. These are explained below:

1. **Expanded Workflow**
   (a) Extend the existing framework to include:

- **Task 3**: Toxic-to-Non-Toxic Style Transfer

(b) Incorporate *a more advanced agentic workflow*. Here you need to choose an agentic wokflow that allows you to carry our all the tasks in tandem. Your agentic workflow can exploit multiple LLMs and employ different types of chains. A high-level visualization of what the agentic workflow could look like is in Figure 1. Note that Figure 1 is offered only to facilitate your understanding of the requirement and should not limit the way you choose to execute your overall system.

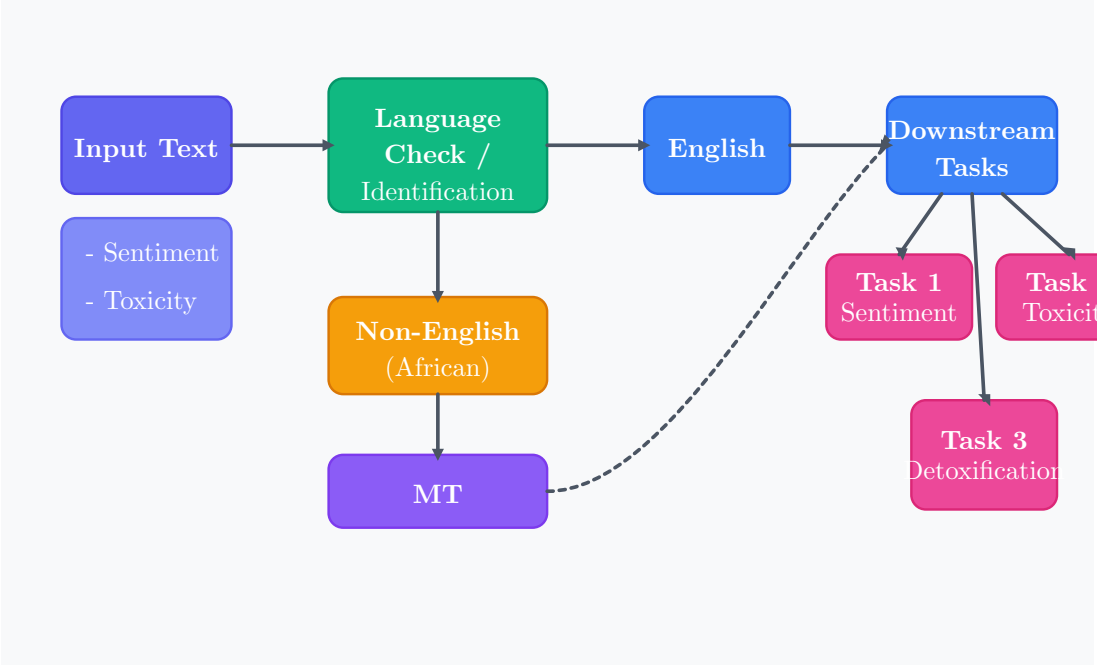

Figure 1: One possible agentic workflow.

2. **Multilingual Sentiment Analysis Dataset**

   - A *multilingual* dataset will be provided. The dataset involves a number of African languages in addition to English.
   - Integrate a translation model (such as those listed earlier in this assignment) to convert non-English text into English before applying Tasks 1, 2, and 3.[1]

3. **Deliverables (Milestone 2)**

   - Updated code that automatically detects or assumes non-English input and performs translation prior to classification and style transfer. Here, you can possibly employ any

[1] We will assume we have no models for direct application on sentiment analysis (or toxicity, or detoxification) and so we translate the text into English and use the English models.

language identification tool that is good with detecting English then assume any non-English is just African that should be sent to the machine translation model. You should find a suitable language identification tool on your own (none is suggested here).

- A brief report (3–4 pages) describing:
  - The enhanced agent-based workflow.
  - How you identify the language of the input text.
  - How you perform style transfer for toxic content, and results of your evaluation of this task (see below).
  - Any improvements or challenges compared to Milestone 1 for Task 1 and Task 2.
  - Other details you deem relevant based on the description of the tasks in Milestone 2. **Make sure you evaluate all three tasks. For Task 3, you will carry out human evaluation.**

6.2 Example Input/Output for Style Transfer (Task 3)

Example inputs for Task 1 and Task 2 were provided in Milstone 1 and you should keep them into account.

Task 3: Toxic-to-Non-Toxic Style Transfer **Input**: "You are completely clueless!"
Output (example):

```
{
  "original_text": "You are completely clueless!",
  "rewritten_text": "I think you're mistaken about that."
}
```

6.3 Evaluation of Detoxification Quality

Carry out a manual analysis of 15 detoxified samples on a scale of 1 to 10 with 1 meaning low quality and 10 meaning high quality (see below). This should be done by two annotators where each annotator labels the same 15 samples, then compare the labels. Describe the evaluation process and results in your report. Details are provided below:

6.3.1 Rating Scale (1 to 10)

- **1**: *Extremely poor detoxification*
  - Almost no reduction in toxicity; original offensive content remains or new offensive content introduced.
- **2–4**: *Below average detoxification*
  - Some elements of toxicity remain.
  - Text may still contain obvious insults, aggressive wording, or majorly problematic phrasing.
- **5–6**: *Adequate detoxification*

  - Major insults or strong profanity may have been removed, but the text may still be harsh or borderline rude.
  - Some negativity or aggression might be lingering, but it's less severe.
- **7–8**: *Good detoxification*
  - The text has little to no offensive language; most or all toxic elements are neutralized.
  - Mild negativity may remain in tone, but it no longer reads as toxic or hostile.
- **9**: *Very good detoxification*
  - The text is polite or neutral in tone, showing minimal or no aggressive elements.
  - Any negative aspects are presented without hostility or insult.
- **10**: *Excellent detoxification*
  - The text is polite, respectful, and completely free of toxicity.
  - The original meaning (including constructive feedback, if present) remains intact without any unpleasant or insulting language.

6.3.2 Annotator Instructions

1. **Read the Original vs. Detoxified Text**
   Familiarize yourself with the original (toxic) version, then carefully read the model's detoxified version.
2. **Judge the Degree of Improvement**
   Compare the two texts and note if the offensive elements have been removed, softened, or replaced with more polite language.
3. **Assign a Score from 1 to 10**
   Refer to the rating scale above. Enter your chosen score in a shared spreadsheet or annotation tool.

6.3.3 Comparing Annotator Scores

Annotators Provide Independent Ratings Each annotator reviews the text privately and records a single integer (1–10) that best reflects the detoxification quality.
Measuring Agreement After both annotators finish, compare their scores for each text.

- Look at how often the scores are identical or within a close range (e.g., ± 1).
- If the scores differ significantly (e.g., 3 vs. 9), discuss to see if there was a misunderstanding of the scale.

Averaging Scores For the final evaluation, you can **average** the two annotators' scores to get a single figure representing the detoxification quality for that text. Alternatively, you can keep both scores and **report their range** or compute *inter-annotator agreement* (e.g., Cohen's Kappa) for a more formal approach.

6.3.4 Suggested Steps for You

1. **Read the Instructions Thoroughly**
   Ensure you understand the 1–10 scale and what constitutes each rating level.

2. **Perform Independent Annotations**
   Each annotator labels all the detoxified sentences without discussing them first.

3. **Record Scores**
   Maintain a structured format (e.g., a spreadsheet) with columns such as *Text ID*, *Annotator A Score*, *Annotator B Score*.

4. **Compare and Discuss**
   Check if there are large discrepancies. Annotators can hold a brief discussion to clarify rating differences.

5. **Aggregate Results**

   - Calculate the average score per text.
   - Optionally, compute measures of consistency (e.g., differences in scores or Cohen's Kappa).

6.3.5 Tips for High-Quality Annotation

- **Focus on Politeness**
  A high score means the text is free from direct insults, profanity, and reads as non-threatening.

- **Watch for Subtle Toxicity**
  Even if major profanity is removed, the text may still be sarcastic or condescending. Factor this into your rating.

- **Maintain Consistency**
  Use the same criteria for each text so that, for example, a "5" always indicates the same detoxification quality level.

Again, note that your code must handle *Task 1, Task 2*, and *Task 3*. In Milestone 2, you are adding Task 3. Also note that you can now update (or make changes to) the way you handled task 1 and task 2. If you are not very clear on what you are required to do, please ask the instructor.

6.4 Grading Rubric for Milestone 2 (30 points)

Component	Criteria	Points
Advanced Workflow	Proper integration of advanced agentic techniques	7
Translation and Lang id Integration	Handling detection of non-English text, correct insertion of translation step	5
Style Transfer Quality	Effectiveness and fluency of rewriting toxic text into non-toxic based on manual analysis of 15 samples on a scale of 1 to 10 with 1 meaning low quality and 10 meaning high quality	3
Model Performance	Accuracy of sentiment/toxicity on the new dataset (2 grades for each)	4
Code Quality	Readability, best practices, reproducibility	4
Written Report	Completeness, clarity, and logical flow in the milestone report	7

7 Due Dates

- **Milestone 1 Due**: Sunday Mar 9, 11:59 pm.
- **Milestone 2 Due**: Sunday Mar 16, 11:59 pm.

8 Formatting Requirement

You are required to use the ACL 2025 Overleaf template for your write-up.

9 How to Submit

- **Identification:** Please make sure your name, name of the assignment, and the course, are clearly marked in your PDF as well as code scripts.
- **Report**: Upload a PDF for each milestone to Canvas.
- **Code**: Upload a zip file containing all relevant scripts/notebooks to Canvas.

10 Academic Integrity and Collaboration Policy

- You may discuss the assignment at a conceptual level with classmates other than your team member, but all submitted work must be your team's.
- Do not share code or written materials directly with other students.
- Cite any external sources or libraries used. Please make sure you cite all your references clearly. This includes any papers you review, any tutorials you benefit from, any code you re-use or modify, etc. It is required to categorically and explicitly cite any material created by

others that you consult. **Failing to abide by this crucial requirement will be treated as plagiarism**. Recommended range of references is $3 - 7$ references, but you can use more. **Please note that reports without any references, or with irrelevant references will be penalized.**

11 FAQ / Additional Notes

- **Q**: Can we use other LLMs not listed for any of the tasks?
  A: Yes, so long as these are models you are able to run for inference and are open models (not closed models such as Claude or ChatGPT). Make sure you cite the library/model name and version in your report.
- **Q**: Do we need to provide confidence scores for each task?
  A: Confidence scores are *optional* but may strengthen your explanation components.
- **Q**: Can we fine-tune models locally or must we rely on zero-shot/in-context methods?
  A: You are not required to fine-tune models, but either is acceptable. Clearly document and justify your approach.
- **Q**: How do we acquire good performance from the models?
  A: There are different ways. These include employing clear prompts of different types (e.g., Chain-of-Thought) as well as in-context learning (i.e., showing the models some samples with labels). You can also explore different chaining methods. Overall, you are free to explore different approaches, including ones not listed here.

12 Late Submission Policy

Assignments received after this deadline will NOT be accepted.