Shipeng Sun
DSCI 510
January 26, 2025

GoEmoX: Multi-Label Emotion Classification with DistilBERT

Project scope update

The overall scope of the project is unchanged compared to the proposal.

Data sources

The main training data come from the GoEmotions dataset released by Google Research. I have implemented a loader in src/load.py that queries the HuggingFace Datasets Server API endpoint https://datasets-server.huggingface.co/rows. The function get_goemotions_from_api downloads a subset of the train split and returns it as a pandas DataFrame, and save_goemotions_to_csv writes the result into CSV files under the project data/ directory. The current sample already includes textual Reddit comments (text), their emotion label indices, and unique IDs.

To prepare out-of-distribution evaluation data, I have also implemented two additional loaders. The first one collects Hacker News content using the official Firebase-based Hacker News API. The function get_hackernews_from_api first retrieves top story IDs and then downloads each story's JSON record. It builds a text field by combining the title and the optional HTML body, and saves a CSV file such as hackernews_sample_test.csv via save_hackernews_to_csv. The second loader targets GitHub issues using the GitHub REST API. The function get_github_issues_from_api fetches a batch of issues from a public repository (currently pallets/flask), filters out pull requests, and creates a text field that concatenates the issue title and body. The helper save_github_issues_to_csv writes these records to data/github_issues_sample_test.csv. All three loaders are covered by simple tests in src/test.py, and running python src/test.py successfully downloads small samples from each API and confirms that the expected CSV files are created in data/.

Issues / difficulties

So far I have focused on setting up the project structure to match the provided sample project, configuring the data/, src/, and doc/ directories, and implementing reliable data loading functions for all planned APIs. I have not yet started model training or detailed experimentation. The main anticipated challenges are handling the multi-label and highly imbalanced emotion distribution in GoEmotions when fine-tuning DistilBERT, managing runtime and memory usage when scaling from small samples to the full training set, and dealing with API rate limits and possible authentication requirements for larger-scale collection from Hacker News and GitHub.

In the next phase, I plan to first run a small proof-of-concept fine-tuning experiment on a subset of GoEmotions to verify that the training pipeline works end-to-end. After that, I will expand to a larger portion of the dataset, add appropriate evaluation metrics, and then construct OOD test sets from the Hacker News and GitHub CSV files. Finally, I will compare in-distribution and out-of-distribution performance to analyze how robust the fine-tuned DistilBERT model is to domain shift and different writing styles.