Do Developer Sentiment and Emotions Affect Software Quality? An Exploratory Study

Md. Arid Hasan[1], Hung Cao[2], Francis Palma[1]

[1] SE+AI Research Lab, Faculty of Computer Science, University of New Brunswick, Fredericton, Canada

[2] AELab, Faculty of Computer Science, University of New Brunswick, Fredericton, Canada,

{arid.hasan, hcao3, francis.palma}@unb.ca

Short Abstract

We aim to study both sentiment analysis and emotion classification and the relationship between developer sentiment and software quality (i.e., the presence of bugs). We developed manually annotated datasets containing 12,005 commit messages for emotion to address the resource limitation. Moreover, we resample the classes to develop the dataset for sentiment analysis. We applied several machine learning (ML) algorithms including support vector machine, random forest, bidirectional long short-term memory (BiLSTM), and pretrained language models (PLMs) to extract emotions and sentiments automatically from the commit messages. We found that the performances of ML models for emotion classification are between 20% – 33% due to several challenges such as difficulties in capturing underlying emotions. As an ongoing effort, we are investigating the relationship between developer sentiment and software quality to understand whether developers' sentiments or emotions affect software quality.

Data Details

- Our dataset has seven classes that are *anger, fear, joy, neutral, sadness, surprise, or trust.*
- Each data was annotated by three annotators and the Inter-annotator agreement is 0.14 (Fleiss Kappa score).
- Approximately 58% data belongs to *Neutral and Trust* classes that represent the skewness of the data.

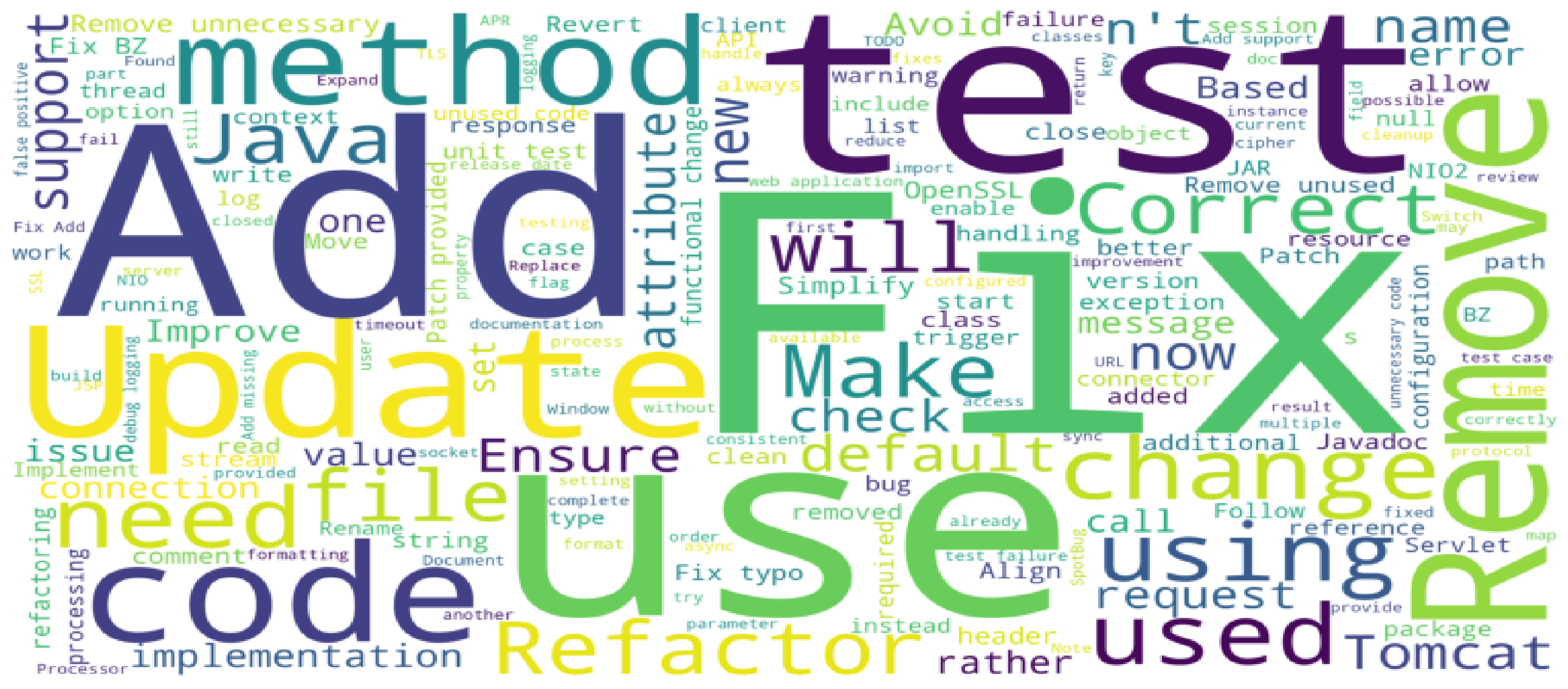


Figure 1. Word-cloud for the dataset

Proposed Methodology

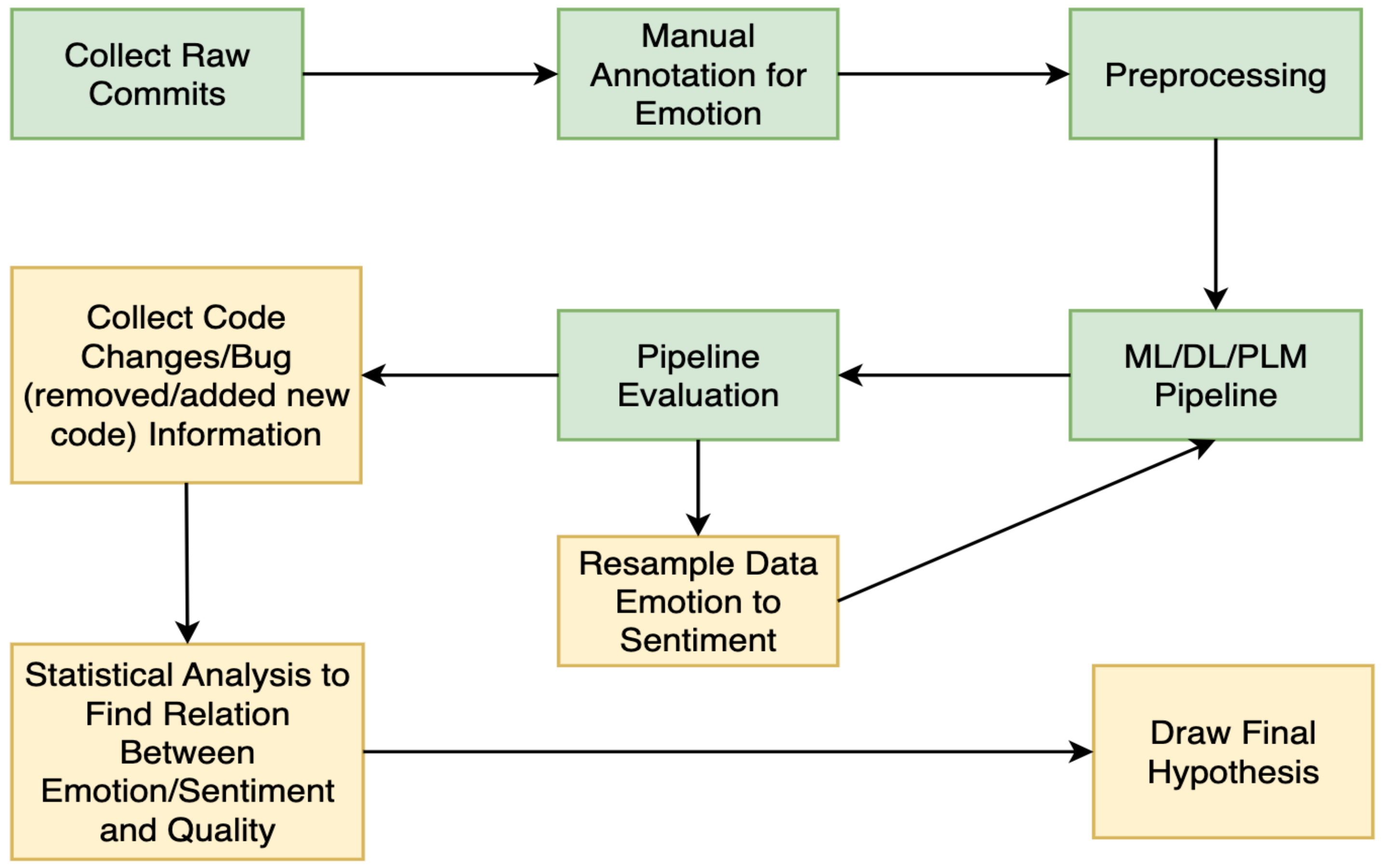
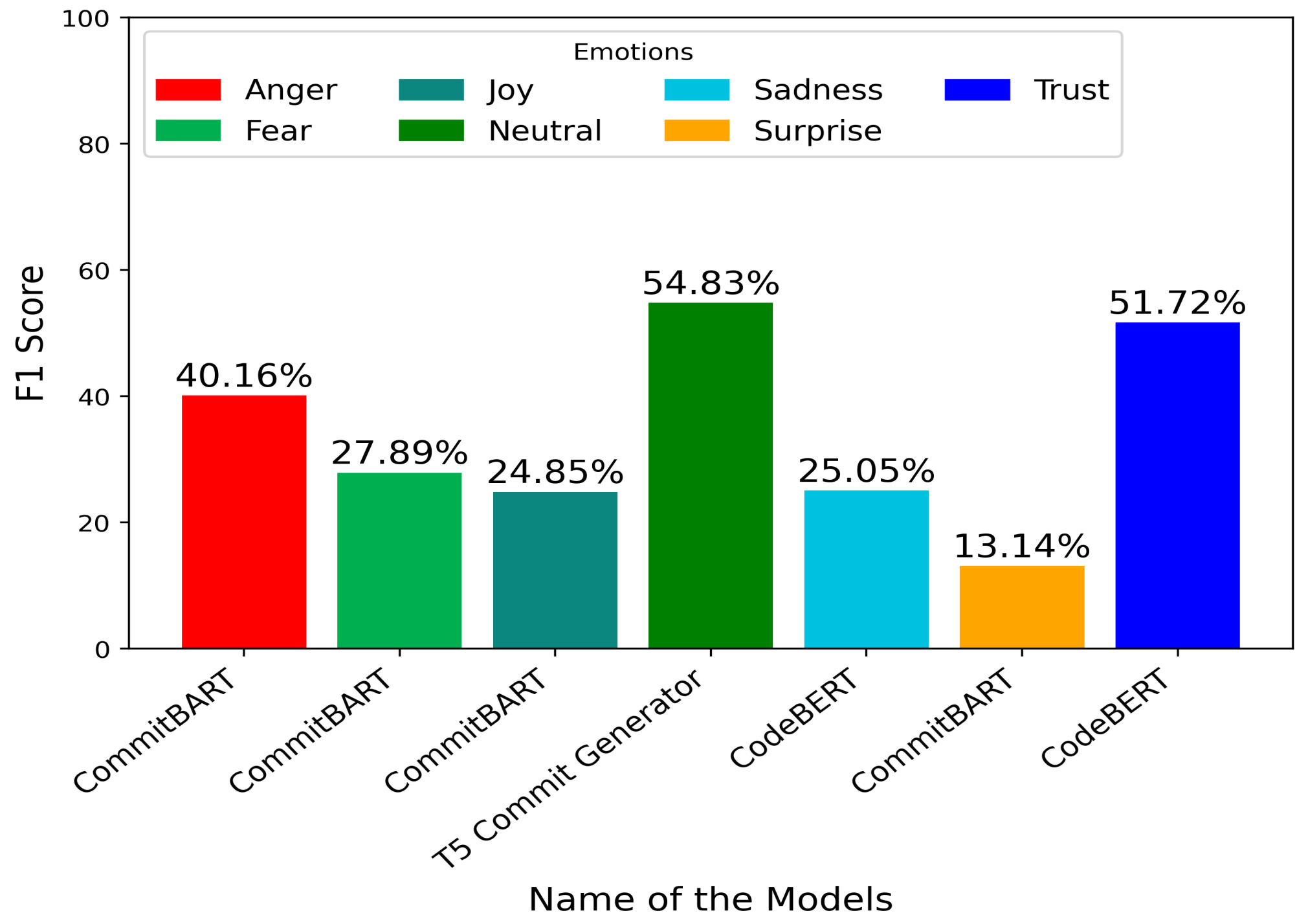

Collect Raw Commits
Manual Annotation for Emotion
Preprocessing
ML/DL/PLM Pipeline
Pipeline Evaluation
Collect Code Changes/Bug (removed/added new code) Information
Resample Data Emotion to Sentiment
Statistical Analysis to Find Relation Between Emotion/Sentiment and Quality
Draw Final Hypothesis

Figure 2. Proposed Methodology. Green rectangular represents that the steps are completed.

Results and Discussion

- While most of the pre-trained language models outperform classical and deep learning models, XLM-RoBERTa-large failed to demonstrate superior performance on commit messages.
- CommitBART is the best-performing model, while CodeBERT shows prominent performance.
- Although the performance is not significant, the PLMs are performing approximately **1.5 times better than the baseline result**.
- The models are delivering better performances on the Anger, Neutral, and Trust classes.
- We identified that capturing the human sentiment/emotion is challenging where the text contains insufficient information.



Model	Accuracy	Precision	Recall	F1-macro
Baseline				
Multinomial Naive Bayes	39.66	35.74	39.66	21.43
Classical Models				
Support Vector Machine	38.25	35.16	38.25	**26.51**
Random Forest	42.27	40.00	42.27	**28.65**
Deep Learning Model				
Bidirectional LSTM	38.50	35.75	38.50	**27.42**
Pre-trained Language Models				
XLM-RoBERTa-large	37.05	21.34	37.05	13.56
CodeBERT	43.72	44.06	43.72	**30.71**
CodeReviewer	42.40	41.48	42.40	**29.97**
CommitBART	43.72	42.96	43.72	<u>**32.70**</u>
T5 Commit Generator	42.77	39.16	42.77	**28.11**

Figure 3. The left figure represents class-wise best-performing models. The right figure represents the performance of different models. **Bold** indicates the models outperformed baseline and <u>Underline</u> indicates the best performing model

Introduction

- The content of the artifact sometimes belongs to multiple classes, making it difficult to annotate in one class and making it challenging for the model to differentiate between classes.
- Identifying different emotions from similar content is challenging to maintain the common standard within the project.
- Some common words are used for most of the commit messages, such as *fix, use, update, add, remove, etc.*

Contributions & Findings

- We built the largest manually annotated datasets for emotion analysis on commit messages.
- We investigated pre-trained language models and fine-tune the models with the data. We are the first to provide a comprehensive analysis of emotion classification on commit messages.
- We also provided a comprehensive performance analysis among the classical, deep learning, and pre-trained models.
- All the models except CommitBART and random forest struggled to predict the surprise class, while XLM-RoBERTa only predicted neutral and trust classes.

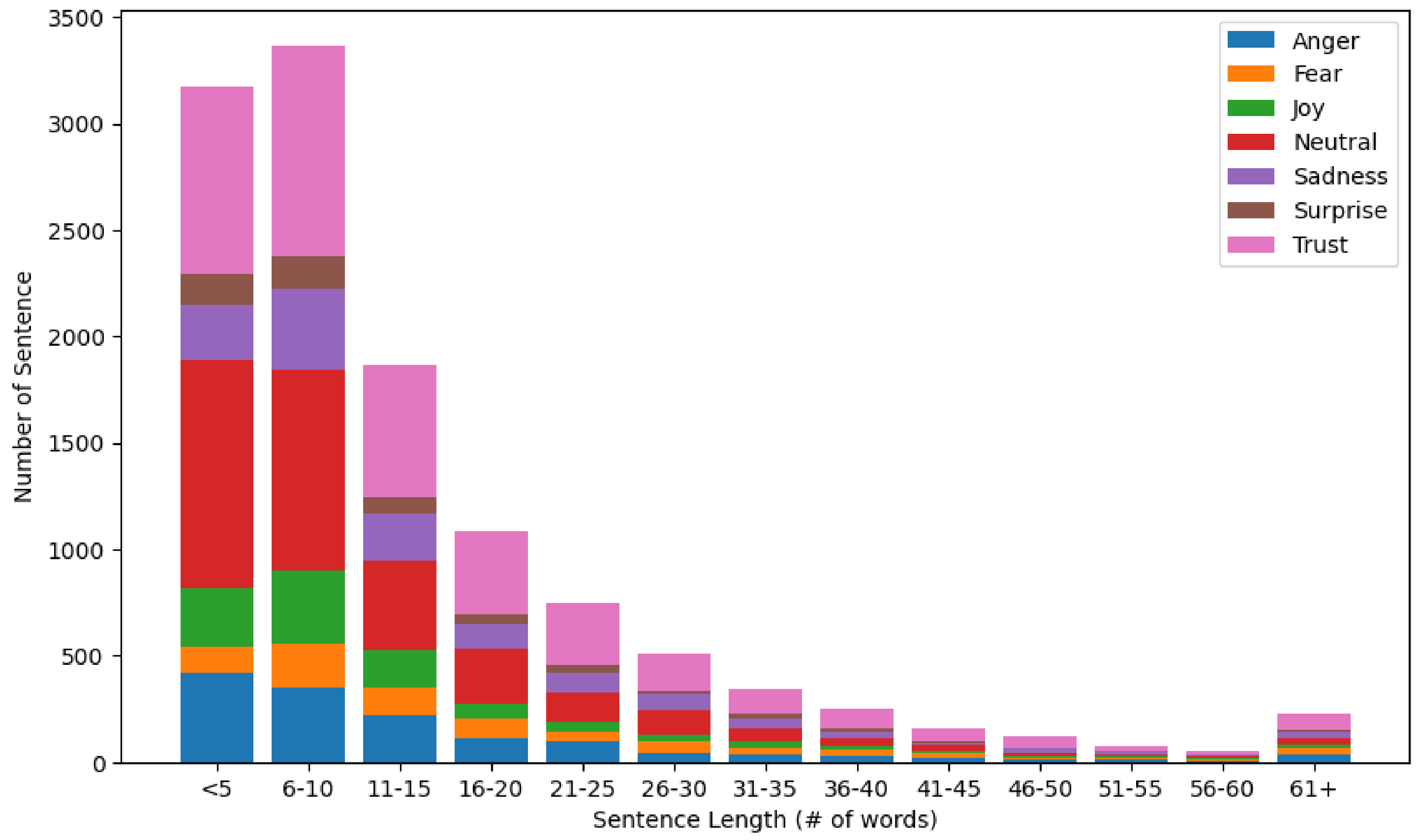


Figure 4. The distribution of sentence length (number of words) associated with each emotion label.

Conclusion

- We present the evaluation performances of pre-trained language models along with deep learning and classical models.
- We also provide a detailed performance comparison among the models.
- We identify the challenges of getting the underlying meaning of the text that causes low performances across the models.
- We will evaluate the ML/DL/PLM pipeline for sentiment analysis using the same dataset.
- We will also investigate how developer emotions and sentiments affect software bugs or code changes.