Mid Term Report

By Sushant Goyal, Enrollment No - 24115152

1. Task Description

My goal behind this project is to build a chatbot that can figure out how someone is feeling based on what they type and also check if their messages hint at something serious and dangerous like depression or suicidal thoughts. The chatbot isn't just for chatting, it's meant to be helpful and supportive. For now, I'm focusing on two main parts: an emotion classifier and a suicide/depression detection model.

The emotion classifier will use the **GoEmotions dataset** (a dataset of 58,000 Reddit comments with 27 labeled emotions + neutral { much more detailed than other emotion datasets}), and the suicide detection part will be trained using the **Suicide- Watch dataset** from Kaggle. Once both models are ready, I'll combine them with a simple self made chatbot that feels human and actually cares about the user.

ML algorithms used and My Approach towards the Project

- **Text Preprocessing :**
    - Performed preprocessing of text using spacy (earlier I had thought of doing it only with NLTK but then I discovered that spacy has a more friendly approach towards text preprocessing, like it splits the corpus in a default manner).
    - At first, I concatenated the 3 csv files present in my **GoEmotions** dataset to 1, and then I converted them into dataframes using pandas, I dropped columns with trivial info like "id", "author", "subreddit", etc.
    - Then I created a function to preprocess the text of both my datasets, used spacy to lemmatize the tokens and used NLTK for stemming (spacy doesn't contain a stemmer). Ensured that all the stopwords and punctuation words are removed from the text.
    - My **GoEmotions dataset** had 28 columns for the emotions and the values under those columns had label names (0/1), this troubled me as I won't be able to label my data properly with them, so I removed the 28 cols and created 1 col named emotions with the name of emotions as values in them.
    - I studied different types of word embeddings, and I decided to use the **Word2Vec model** to turn the preprocessed tokens into vectors using the **gensim library** from Google, I kept the vector size to 300 to ensure the text representation as possible.

- **Emotion Classification:**
    - I'll start with simpler models like Logistic Regression for my model training, They are known to be easier to interpret, but I am also learning about transformer models like BERT, so I may pivot to make my model more efficient.
- **Suicide Detection:**
    - Same process as above but trained on the Suicide-Watch dataset to detect warning signs in text.
- **The Chatbot:**
    - I asked my mentors about it and they suggested me to code a simple chatbot myself instead of using any API. The bot will take the output from the models and respond based on the emotion or risk level detected. It will have features like suggesting mental wellbeing tips and inspirational quotes depending on the user's mood. (These tips can be scraped from many mental wellbeing subreddits on Reddit and other online websites too)

Suggestions by Mentors

- They helped me remove the binary labels from my GoEmotions Dataset by suggesting a way to solve it.
- They also suggested to me which library to use for vectorization.
- The idea of creating a simple chatbot to support the models was also given by my mentors.

---

2. Dataset

The emotion classifier will use two datasets given as follows:

1. GoEmotions Dataset

Source and Accessibility:

The GoEmotions dataset is available on Kaggle, and it was easier for me to access it, since I was making my project on Kaggle itself. It's a comprehensive dataset developed by Google, designed for emotion classification tasks. The dataset contains text data sourced from Reddit comments, which makes it quite diverse and reflective of real-world communication styles.

Size and Characteristics:

- The dataset has about 58,000 Reddit comments.
- Each comment is tagged with one or more emotions from 27 possible labels (like joy, sadness, anger, etc.), plus a neutral category.
- The comments are relatively short, which is perfect for this project since the chatbot needs to process quick inputs. However, some comments have overlapping labels, which could make classification a bit tricky.

Modifications and Preprocessing:

1. At first, I concatenated the 3 csv files present in my **GoEmotions** dataset to 1, and then I converted them into dataframes using pandas, I dropped columns with trivial info like “id”, “author”, “subreddit”, etc.
2. Then I created a function to preprocess the text of both my datasets, used spacy to lemmatize the tokens and used NLTK for stemming (spacy doesn’t contain a stemmer). Also, Converted all text to lowercase to ensure uniformity during vectorization.
3. Removed stopwords like "is," "and," or "the" that don’t add value for emotion detection.
4. My **GoEmotions dataset** had 28 columns for the emotions and the values under those columns had label names (0/1), this troubled me as I won’t be able to label my data properly with them, so I removed the 28 cols and created 1 col named emotions with the name of emotions as values in them.
5. Used **Word2Vec** to convert words into numeric vectors that capture relationships between them.

Challenges I’m Tackling

- The GoEmotions dataset has 27 emotions (it is pretty complex compared to other emotion datasets), some of which overlap. For instance, “grief” and “sadness” are pretty close, so the model might get confused. I will need to fine-tune it carefully.

2. Suicide Watch Dataset

Source:

This dataset was also sourced from Kaggle. The dataset is a collection of posts from Suicide and depression subreddits of Reddit.

Size and Characteristics:

- Roughly 232700 entries of text data, labeled as either "suicidal" or "non-suicidal."
- The dataset is well authored as the dataframes didn’t require much editing, removing trivial columns and preprocessing and vectorising it worked quite fine.

  Its modification and processing is similar to the previous dataset.

---

3. Model Architecture

I have already discussed the Text Preprocessing part and vectorisation part in detail in section1. Here I discuss why I have decided to start with Logistic regression as a model for my emotion classifier model.

Logistic Regression

Logistic Regression is a simple yet effective model for binary classification tasks, which is why I’ve chosen it as the starting point. In our case, for the emotion classification task, I am technically dealing with multiple emotions, so each emotion is treated as a separate binary classification problem. Logistic Regression will determine whether a given input text expresses a particular emotion (1) or not (0).

For the **suicide detection** task, the model will similarly classify each input as either indicating a risk (1) or not (0). It is ideal for a starting point before exploring more complex models like transformers...

Pipeline in a Nutshell

1. **Preprocessing**: Clean and transform text data using spacy and NLTK.
2. **Vectorization**: Use Word2Vec to represent the text in a way the logistic regression model can understand.
3. **Model Training**: Train logistic regression model to predict emotions and detect signs of suicide.
4. **Evaluation**: Test and evaluate the performance of the model using metrics like accuracy and F1 score.
5. **Fine-Tuning**: After getting a solid baseline, we can improve performance by exploring more advanced models like BERT

---

4. Implementation

Here is the link of the work that I have done until now.
mENTAL wELLBEING cHATbOT

Implementation Steps performed till now

I initially planned to perform text preprocessing only using NLTK, but then I discovered that SpaCy offers a more streamlined and efficient approach, particularly in how it splits the corpus by default. To start, I merged the three CSV files from the GoEmotions dataset into one and converted them into a single dataframe using pandas. I removed unnecessary columns like "id," "author," and "subreddit" that didn't contribute to the analysis.

Next, I created a function to preprocess the text in both datasets. I used SpaCy to lemmatize the tokens and NLTK for stemming, as SpaCy lacks a built-in stemmer. I also ensured that stopwords and punctuation were removed.

After that, I explored different word embedding techniques and decided to use the Word2Vec model from the Gensim library to convert the preprocessed tokens into word vectors. I set the vector size to 300 to ensure the most comprehensive text representation.

---

5. End-Term Goals

1. Emotion Classification Model

- Current Status: Preprocessing and Word2Vec for vectorizing text are done. Now, it's time to train the emotion classification model.
- Next Steps:
  - Train Model: Start with logistic regression on the GoEmotions dataset and evaluate its performance using metrics like accuracy and F1 score.
  - Enhance: If the logistic regression model works well, I'll explore transformer models like BERT to improve accuracy.
- Target: Aim for an F1 score of at least 0.75 using logistic regression, with plans to improve it using advanced models.

2. Suicide Detection Model

- Current Status: Preprocessing and Word2Vec are done for the Suicide-Watch dataset. Next, I will focus on building the detection model.
- Next Steps:
  - Train Model: Train a logistic regression model on the Suicide-Watch dataset to detect potential signs of depression or suicidal thoughts.
  - Testing: Evaluate the model's performance and try other algorithms if needed to improve results.
- Target: Achieve at least 80% accuracy in detecting suicide-related content.

3. Chatbot

- Current Status: So far, I've set up the preprocessing, vectorization, and emotion detection models. The next step is integrating them into the chatbot.
- Next Steps:
  - Build Chatbot: Create a simple chatbot that takes user input and responds based on detected emotions or risk levels.
  - Integrate Models: Link the emotion and suicide detection models so the chatbot can provide relevant responses, like mental health tips and inspirational quotes.
- Target: Have a working chatbot that interacts empathetically based on emotion and risk detection.

---

6. References

Datasets

1. **GoEmotions Dataset**

   GoEmotions

2. **Suicide-Watch Dataset**

   Suicide and Depression Detection

Research Papers and Tutorials (used besides the one mentioned in proposal)

1. Frontiers | Detection of emotion by text analysis using machine learning

2. Gensim Word2Vec Tutorial

Tools

1. **SpaCy**
   SpaCy Official Website
2. **NLTK (Natural Language Toolkit)**
   NLTK Documentation
3. **Gensim**
   models.word2vec – Word2vec embeddings — gensim
