Lab Design 2 – Sentiment Analysis using Pre-Trained Models

Name – **Ravi Tarun Prasad Nimmalapudi**

Course - **CPSC 5960 03 PyTorch Lab Design**

❖ Problem Statement:

The objective of this lab is to perform sentiment analysis on a given data set containing tweets (textual data). This data also includes sentiment labels like 'Positive', 'Negative' and 'Neutral'. The task is to evaluate and compare the performance of two of the sentiment analysis models: 'VADER' and 'Hugging Face'.

The main task is choosing the appropriate model for the job. With numerous models available for sentiment analysis, it becomes crucial to identify which model is best suited for the specific problem. Pre-trained models play a key role in solving such problems, making our lives easier. These models save time and effort compared to training a model from scratch because they have been trained on huge datasets and can handle sentiment classification effectively.

So, this lab mainly focuses on classifying the given text's sentiment using pre-trained models and comparing the performance of both.

❖ Approach:

The approach for this sentiment analysis task is straightforward and follows the standard steps for any other sentiment analysis problem. The goal and the workflow go like understanding the data, cleaning it, preparing it and then applying the models for sentiment classification and comparing their results.

Before going into the workflow let us make sure what are pre-trained models exactly, how do they make our lives easier and which models we are going to use below.

What are pre-trained models?

As the name suggests pre-trained models are machine learning models that have already been trained on large datasets and are ready to be used for specific tasks. Some of the common tasks are image recognition, text classification and sentiment analysis etc.

Why pre-trained models?

In simple terms, pre-trained models are like ready-to-use tools that have already been learned from experience and can quickly act upon given data without the need for starting from scratch. Like instead of training a model from scratch, which requires a lot of time, data and computational power, pre-trained models have already learned useful patterns from a large amount of data.

For example, a pre-trained model for sentiment analysis has already been trained on thousands or millions of text examples to understand how words, phrases, and contexts convey sentiment (like positive or negative feelings).

Now let us understand the two models we are going to compare:

VADER:

This basically stands for 'Valence Aware Dictionary and sEntiment Reasoner'.

This is a simple and efficient tool for sentiment analysis that is particularly designed for social media texts (like tweets). This basically uses a predefined lexicon of words and rules to assess the sentiment of a text. Every word in the lexicon has a sentiment score, and VADER uses the scores of the words in the text to determine the overall sentiment. Based on the strength of the word scores, it classifies sentiment as neutral, negative, or positive. VADER works well for short texts and accounts for things like punctuation, capitalization, and slang commonly used in social media.

Hugging Face Pipeline:

Hugging Face is an open-source community that focuses on making state-of-the-art natural language processing (NLP) models and tools more accessible to everyone. They are best known for their **Transformers** library, which provides easy access to pre-trained models for tasks like sentiment analysis, text generation, translation, summarization, and more.

The pipeline API simplifies the integration of various components like tokenization, named entity recognition, and sentiment analysis into a single, cohesive workflow. This enables quick experimentation, efficient fine-tuning, and seamless deployment of models into production. Built on top of PyTorch and TensorFlow, the library supports python 3, making it versatile and accessible for a wide range of applications.

❖ **Workflow:**

- **Understanding the data and the problem:**

Putting in simple terms, the task is to perform sentiment analysis on a given set of text data and compare the performance of two popular approaches: 'VADER' and The pre-trained models available through 'Hugging Face's Transformers Library'.

This project's main goal is to evaluate and compare how well these two methods work on the provided dataset in terms of accuracy, ease of use, and performance.

Data:

To have a brief understanding of the data we will be working on, this dataset was sourced from a 'Kaggle' competition event called 'Tweet Sentiment Extraction' and contains the following columns:

- textID – Unique ID for each piece of text.
- text – The text of the tweet.
- sentiment – The general sentiment of the tweet.
- selected_text - The text that supports the tweet's sentiment.

The reason for selecting this dataset is it already has a refined column called 'selected_text' that indicates the sentiment-determining component of the tweet. This makes it particularly suitable for our basic sentiment analysis task, as it provides a clear and focused reference for the tone of the tweet.

This dataset is perfect for evaluating the performance of various sentiment analysis models, such as VADER and Hugging Face's pre-trained models, because it makes it simple to train and assess models without the need to preprocess or extract pertinent sentiment-bearing text.

- **Quick Data Exploratory Analysis:**

Whenever dealing with dealing with any real-world (scrapped) data, the first step is to prepare the data for our task. In our case, we are working on a sentiment analysis task and the dataset contains various columns and we would be starting by selecting required columns.

This stage of data preparation is essential for ensuring the quality of the data before applying sentiment analysis models, helping to achieve more accurate and reliable results.

- Select Required Columns:

  We mainly focus on the 'selected_text' column, which contains the text that supports the sentiment of the tweet, and the column 'sentiment' contains the actual labels that would be later used for evaluating our models.
  The 'textID' column, which might be used to identify individual tweets, does not contribute meaningful information for our task, so dropped.

- Handle 'sentiment' Column:

  We eliminate any rows labelled 'Neutral' since Hugging Face's pre-trained model does not classify neutral sentiment, we only keep rows labeled as either positive or negative for a fair comparison between the models.

- Check for null values:

  Since we selected the features, we want now one important thing while dealing with real-time data would be dealing with null values because missing values can lead to issues during model training and evaluation. Early detection and management of these missing values guarantees that the dataset is complete and prepared for additional processing.

- Basic Clean:

  Now we proceed for some basic cleaning to prepare the textual data for tone analysis. As we already have the 'selected_text' column which contains the refined text that directly supports the tone of the text, we would be performing some basic preprocessing on the 'selected_text' column which includes converting to lowercase for consistency, removing special characters and stripping whitespaces.

- **Building the Sentiment Analysis Pipeline:**

As the data is prepared, we now build the sentiment analysis pipeline. Building the analysis pipeline involves initializing models that would basically allow us to predict the tone of the statement.

This pipeline setup acts as the basis for the project's evaluation phase and allows us to evaluate and compare the sentiment analysis capabilities of VADER and Hugging Face.

Starting by initializing the VADER sentiment analyzer, which determines the sentiment of a given text by using a vocabulary of terms with corresponding sentiment scores. VADER is especially effective for social media text, as it can account for various forms of sentiment expression, including slang, emojis, and punctuation. With VADER, we quickly obtain a sentiment score for each text row, which will be classified as "Positive," "Negative," or "Neutral" based on the score.

Next, setting up the Hugging Face Pipeline. Hugging Face offers pre-trained models for sentiment analysis and other NLP tasks that have been fine-tuned on huge datasets. In this case, we use the default sentiment analysis pipeline, which is based on a model fine-tuned for text classification tasks. Each text row is given a sentiment label of either "Positive" or "Negative" via this pipeline.

In the end, we ignore the 'Neutral' sentiment rows predicted by the VADER model after getting predictions from both models. Because Hugging Face does not assign a "Neutral" label to its predictions, this guarantees an even comparison between the two models.

- **Evaluation:**

Now we simply evaluate the predictions obtained from both the models with the actual ground labels present in the 'sentiment' column. Also, for a fair comparison we just ignored the VADER predicted 'Neutral' rows.

The accuracy of both models is then calculated and compared revealing that **Hugging Face achieves 94.15%** accuracy while **VADER achieves 94.97%**. This suggests that both models function equally, with VADER having a minor edge in this particular task.

Analysis:

VADER has a slight edge in this specific case because of the nature of the data we are working with simple tweets. A lexicon-based model such as VADER can more easily predict the sentiment because the dataset already offers a refined sentiment in the form of the "selected_text" column. VADER is highly effective for short and informal text, like tweets, as it is specifically designed to handle social media language, including slang, emoticons, and capitalization used for emphasis.

Hugging Face's pre-trained model works well, particularly for more complicated tasks, but it works best in situations when the data contains more noise or ambiguity. Hugging Face models can be tuned for domain-specific activities while understanding increasingly complex language. This makes it a great choice for handling more complex, nuanced texts that may require deep contextual understanding. However, in our case, where the dataset is relatively straightforward and already has a structured sentiment classification, VADER is able to produce accurate results with less computational overhead.

- **Future Scope:**

This task can be significantly extended in various ways:

- Multiclass Sentiment Analysis:

  Instead of binary classification (‘Positive’/ ’Negative’), the scope can be expanded to multiclass sentiment analysis revealing deeper insights.

- Fine Tuning:

  Fine tuning the hugging face model on larger corpus of tweets or social medial specific topics to a certain extent could yield better results.

- Testing on Diverse Datasets:

  Assessing the robustness of generated model by testing it on a broader range of datasets.