Universiti Teknikal Malaysia Melaka (UTeM)

BITI 3413 NATURAL LANGUAGE PROCESSING

2022/2023

PROJECT

Title: Sentiment Analysis

Prepared for: Dr. Halizah Binti Basiron

Prepared by:

NAME	MATRIC NO
MUHAMMAD NUR HASIF BIN ABU BAKAR	B032010021
MUHAMAD FARIS BIN AWAL	B032010053
MOHD BRUCE LEE BIN MUSTAPA	B032010120
RUSYDI NASUTION BINTI RIDUAN	B032010215

Table of Contents

1.0 Introduction

The process of determining the emotional tone underlying a text, whether a phrase, paragraph or page, is commonly referred to as sentiment analysis. It is commonly used to fundamentally determine the writer's or speaker's attitude and can be used to assess public opinion on a certain subject, or so they believed. Sentiment analysis frequently yields a label, such as positive, negative, or neutral, that describes the sentiment reflected in the text, or so they believed.

Sentiment analysis is often employed in natural language processing and computational linguistics, and it has a wide range of applications, including customer service, market research, and subtle social media monitoring. The process may generally be carried out using either a rule-based strategy or machine learning techniques.

Sentiment analysis can provide notably positive, negative, or neutral findings, revealing that sentiment analysis frequently produces a label, such as positive, negative, or neutral, that reflects the sentiment expressed in the text, which is rather significant.

Sentiment analysis is used for several purposes, including:

1. Businesses may use Sentiment analysis to track public opinion about their brand or products on social media platforms.
2. Customer service: Sentiment analysis may be used to automatically categorize client comments and complaints, allowing businesses to respond to customer demands more effectively.
3. Market research: Companies may utilize sentiment analysis to learn about the public's perception of their products, services, or rivals.

2.0 Analysis of the developed system

The system that develop by team is done on *Streamlit* web-based app. It can create a user-friendly and interactive sentiment analysis app. In the web-based app, sentiment analysis is performed based on text input or text retrieve from the Yelp and Foursquare websites.

The system uses the *requests* library to scrape reviews from Yelp and Foursquare websites. While *BeatifulSoup* library parses the HTML and extracts the reviews.

The model for the system is a pre-trained model using the BERT model 'nlptown/ bert-base-multilingual-uncased-sentiment'' from the transformers library. It uses the tokenizer from the model to encode the text before passing it through the model to get a sentiment score. The sentiment scores are then displayed in *pandas* dataframe for the user to see.

However, this system had a few limitations as this is just a prototype product. First, it can only scrape Yelp and Foursquare websites. As we know that Yelp and Foursquare are not as widely used and famous as other platforms such as TripAdvisor or Google Business. But using this prototype system, it can be improved to develop a better app that just not only performs sentiment analysis on Yelp and Foursquare website but also on other platforms too.

As this system was developed using the pre-trained model, it can be used for a variety of tasks, but may not be specialized for a specific task or dataset. BERT model in this system is trained on a large dataset of diverse texts that make it suitable for a wide range of NLP tasks, but they may not perform as well on tasks for sentiment analysis of reviews of a specific product. However, this model still outweighs the pros and cons.

The BERT model training data is based on reviews in six languages. So, it has limited to performing sentiment analysis in only six languages. For other languages than these six, it may not be able to be analyzing the reviews.

3.0 Design of the developed system

Our project is the project where we can have the sentiment analysis that provide the score of each rating of the review.

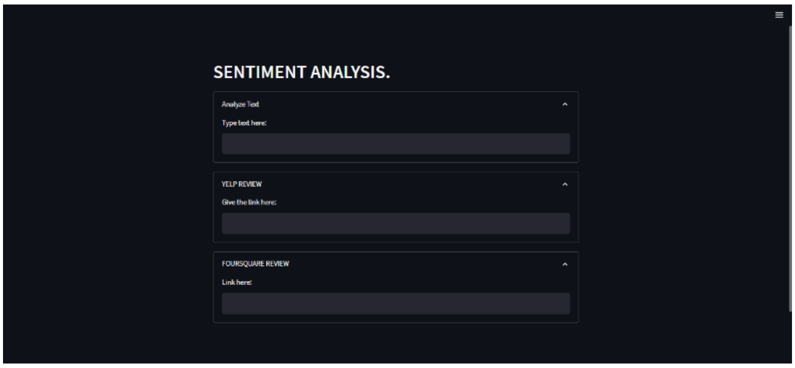

Figure 1

We design the interface of the system like that because people need to see what function that we have easily. When the user click the square box, drop down menu will appear and they need to enter a text. For the text analysis user just need to enter a sentence or review then press enter and the output or rating will come out.

Next, for another two functions, the user needs to enter the link of the specific product or location from each respective website to rate the review that people have given on the product.

In figure 2 we can see the flowchart of our project. We design it as where it can accept an input which is a string or text. With the text, we tokenize it to split the string or text into smaller units so that it can be more easily assigned meaning and do the sentiment analysis.

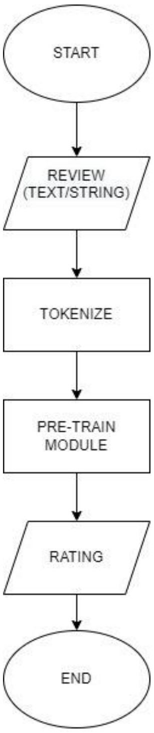

Figure 2

After that, we give the tokenize text to the model. The model is a pre-train model where it can accept a text and analyse it base on the data that the model has. With that it can give the rating to the text. The rating that being output contains numbers from 5 to 1, where 5 meaning that the text or the review is good and 1 means that the review or text is bad.

The design of the system is to make sure that the rating that being output by the system has good accuracy and good sentiment analysis of the rating text from a website.

4.0 Implementation

In development of the sentiment analysis, there is various library that been use in the systems which is textblob, pandas, streamlit, transformers, torch, requests, bs4, re and numpy. The library has their specific functionalities that commonly used in natural language processing (NLP) and data manipulation tasks. It needs to be installed in the virtual environment before can perform the code execution.

```
#create virtual environment
#py -3.10 -m venv .venv

#library installation
#pip install torch
#pip install transformers
#pip install streamlit
#pip install bs4
#pip install pandas
#pip install textblob

from textblob import TextBlob
import pandas as pd
import streamlit as st

from transformers import AutoTokenizer, AutoModelForSequenceClassification
import torch
import requests
from bs4 import BeautifulSoup
import re
import numpy as np
```

There is pre-trained model that use in our system. The pre-trained model that use is ‘nlptown/bert-base-multilingual-uncased-sentiment’, which is based on BERT (Bidirectional Encoder Representation from Transformers) architecture to perform the sentiment analysis. BERT is a pre-trained transformer model that is trained on a large corpus of text data and can be fine-tuned for various natural language processing tasks.

```
#instantaite model
tokenizer = AutoTokenizer.from_pretrained('nlptown/bert-base-multilingual-uncased-sentiment')
model = AutoModelForSequenceClassification.from_pretrained('nlptown/bert-base-multilingual-uncased-sentiment')
```

The idea of using pre-trained model is to give benefits to our system as it can reduce the computational resources required to train a model from scratch as it been trained on large amounts of data. So, the system can achieve better performance than models trained from the scratch.

The next step, the system will implement streamlit to building interactive web-based data applications. It try to retrieve text input or the collection of the review from Yelp and Foursquare websites. From the input, either text input, Yelp URL or Foursquare URL, the collection of the text will pass through to model and tokenizer to get the sentiment score from 1 to 5.

```
#for text from user
with st.expander('Analyze Text'):
    text = st.text_input('Type text here: ')
    if text:
        tokens = tokenizer.encode(text, return_tensors='pt')
        result = model(tokens)
        st.write('Rating (*/5) : ',int(torch.argmax(result.logits))+1)
```

```
#retrieve the url of Yelp website
with st.expander('YELP REVIEW'):
    text = st.text_input('Give the link here: ')
    if text:
        #collect review from the Yelp website
        r = requests.get(text)
        soup = BeautifulSoup(r.text, 'html.parser')
        regex = re.compile('.*comment.*')
        results = soup.find_all('p', {'class':regex})
        reviews = [result.text for result in results]

        df = pd.DataFrame(np.array(reviews), columns=['review'])

        #encode and calculate sentiment
        def sentiment_score(review):
            tokens = tokenizer.encode(review, return_tensors='pt')
            result = model(tokens)
            return int(torch.argmax(result.logits))+1

        #adding rating column to dataframe
        df['Rating (*/5)'] = df['review'].apply(lambda x:
sentiment_score(x[:512]))

        #displaying the dataframe to user
        df
```

```
#retrieve the Foursquare website
with st.expander('FOURSQUARE REVIEW'):
    ##collect review from the Foursquare website
    text = st.text_input('Link here: ')
    if text:
        r = requests.get(text)
        soup = BeautifulSoup(r.text, 'html.parser')
        regex = re.compile('.*tipText.*')
        results = soup.find_all('div', {'class':regex})
        reviews = [result.text for result in results]

        df = pd.DataFrame(np.array(reviews), columns=['review'])

        #encode and calculate sentiment
        def sentiment_score(review):
            tokens = tokenizer.encode(review, return_tensors='pt')
            result = model(tokens)
            return int(torch.argmax(result.logits))+1

        #adding rating column to dataframe
        df['Rating (*/5)'] = df['review'].apply(lambda x:
sentiment_score(x[:512]))

        #displaying the dataframe to user
        df
```

Lastly the system will display the score of the sentiment in pandas dataframe for the user to see in the web.

5.0 Conclusions

A summary of the overall attitude indicated in the reviews might be included in the conclusion. It displays, for example, if the majority of evaluations were positive, negative, or neutral. Specific insights, such as common themes or phrases connected with favorable or negative reviews, as well as any noticeable patterns or trends, may also be included. However, in our study, all of the reviews will produce information from the rating bar, such as the number of stars the reviews have.

A breakdown of emotion by specific features of the product or service being assessed, such as design, performance, or customer service, may also be included in the conclusion. This might assist firms in identifying particular areas where they shine or fall short and making adjustments appropriately.

Furthermore, the conclusion might address any limits or potential sources of inaccuracy in the study, such as the likelihood of biased or fraudulent reviews or the difficulty in effectively distinguishing sarcasm or irony. Any recommendations for more research or changes that may be addressed in the future analysis may also be included in the conclusion. Overall, the findings of sentiment analysis of web reviews may give useful insights for firms seeking to understand and enhance consumer happiness.