A Project Report on Sentiment Analysis of Internship Comments

Grace Kwagalakwe

Department of Computer Science

Makerere University

kwagalakwegrace10@gmail.com

2021/HDO5/2302U

Abstract

Field attachment is defined, in Makerere University, as the field based practical work carried out by staff and students for the purpose of teaching and/or research in places outside of the University control but where the University is responsible for the necessary safety of its staff, students, and others exposed to their activities. Students are always attached to organization and field supervisors. These Field Supervisors have provided valuable feedback to the University about its students. Unfortunately, this feedback is in form of unstructured text and large in volume which makes it difficult to process and gain useful insights. This project is undertaken to do sentiment analysis on the comments provided by the field supervisors so that meaningful insight can be obtained from them. A Textblob sentiment analysis algorithm was used to cluster the comments and the results show that majority of students are good and work well in their internship placements. The clustering performed can be improved by using another model like K means clustering, which model can be evaluated and assess how well the model performs. A pre-trained NER model from the spacy library was used to identify entities in comments belonging to the categories; Person, Geopolitical Entity, time, organization etc. The model performance is fair as it couldn't identify some entities and so there is need to improve it.

Introduction

Field attachment is defined, in Makerere University, as the field based practical work carried out by staff and students for the purpose of teaching and/or research in places outside of the University control but where the University is responsible for the necessary safety of its staff, students, and others exposed to their activities. During Field Attachment, students get attached to organizations to experience the real life of work. Overtime, Makerere University has placed its interns Field Supervisors in organizations of attachment. These Field Supervisors have provided valuable feedback to the University about its students. Unfortunately, this feedback is in form of unstructured text and large in volume which makes it difficult to process and gain useful insights. Therefore, there is a need for tools and techniques that can be used to structure comments and make the processing easier and also make it easy to gain useful insights out of them.

This project presents the methodology and the discussion of results for the text analysis tasks done to gain useful insights from the internship comments provided by the student's supervisors. The comments were downloaded from (https://www.fams-cit.com/fscomments). The tasks include (i) creating a corpus from the downloaded comments, (ii)Clustering the comments about Interns into categories: Excellent, Good, Neutral, Poor, Very Poor, (iii) Assessing the performance of the clustering algorithm, (iv) Creating a Named Entity Recognition model that takes in a comment as an input and outputs the Entities, if any, belonging to the categories: Person, Organization, Place/Location, Time, and lastly Creating a visualization to show insights about the dataset. The rest of the paper is organized as follows, in the next section the materials and Methodology are presented and then results and discussion are then presented.

2. Materials and Methodology

2.1 Dataset

In this task, a dataset of internship comments from field supervisors of Makerere students was utilized. This dataset was obtained from the following link; https://www.fams-cit.com/fscomments. The dataset included 4947 comments.

Pre-processing, clustering and creating models was done in google colab notebook and the necessary libraries imported in the notebook. Visualization of data was done using visual studio code.

2.2 Creating a Corpus

A corpus is a collection of text document over which we would apply text mining or natural language processing routines to derive inferences.

The corpus was created using the NLTK tokenize package for sentences. This tokenizer required the Punkt sentence tokenization models to be installed. These were downloaded via the command: nltk. download('punkt') and were successfully installed. The length of the corpus was then printed and a method to calculate the word frequency defined. This was then converted to a dataframe and then printed out.

2.2 Clustering

A distribution model-based clustering method that involves dividing the data based on the probability of how a dataset belongs to a particular distribution was used. Specifically the text blob analysis.

Libraries Used

Openpxl

Openpxl is used to reading the Comments.xlsx workbook into a data frame and we iterate the dataframe into a python list.

Text Blob

Textblob is a Python NLP library that uses a natural language toolkit (NLTK). Textblob can be used for complex analysis and working with textual data. When a sentence is passed into Textblob it gives two outputs, which are polarity and subjectivity. Polarity is the output that lies between [-1,1], where -1 refers to negative sentiment and +1 refers to positive sentiment. Subjectivity is the output that lies within [0,1] and refers to personal opinions and judgments. Textblob is mostly used to carry out the task of sentiment analysis using its pre-trained inbuilt classifier and can carry out several sentiment analyses.

Clustering the comments

Firstly, before clustering the comments, data pre-processing was done. This involved cleaning the test using a defined clean text function. The function performed several transformations among

them include; tokenizing the text and removing punctuations, removing words that contain numbers, removing stop words, part of speech tagging that assigns a tag to every word to define if it corresponds to a noun, a verb, adjective etc, lemmatizing the text which transforms words into their root form.

A python list of comments is created from the comments extracted to the Comments sheet. Iterations are done on the list and for each item a sentiment analysis is done on them to produce a polarity score of the sentences using Textblob library (the polarity score ranges from 1 to -1. A module called comments which has a function called comments is used to cluster these comments based on their polarity score. Fig 1 shows the assignment of the polarity score to the clusters, where polarity between 0.80 and 1 – Excellent is awarded, polarity between 0.8 and 0 – Good, polarity that is 0 – Neutral, polarity between -0.5 and 0 – Poor and polarity between -0.50 and -1 – Very poor is awarded by the function.

```
score = ''
if sentiment >= 0.80 and sentiment <= 1:
    score = "Excellent"
if sentiment > 0 and sentiment < 0.80:
    score = "Good"
if sentiment == 0:
    score = "Neutral"
if sentiment < 0 and sentiment >= -0.50:
    score = "Poor"
if sentiment < -0.50 and sentiment >= -1:
    score = "Very_Poor"
```

Fig1: Assignment of Polarity scores to clusters.

The score was appended to the corpus and for every sentence in the corpus, a class label was assigned depending on the value of polarity scored. The cluster names and cluster labels were displayed to indicate the label corresponding to each cluster.

The corpus was then transformed into a two column vector using the TfidfVectorizer with one column showing the comment index and another showing the assigned cluster. A two column vector was then printed out with the first column showing the comment index and the second column showing the cluster assigned to that particular column.

The number of comments belonging to each cluster were obtained using the count function and lastly a csv file was generated from the cluster count of the corpus. The csv had three columns, the column for the cluster, the column for the count and the column for the percentage of each cluster out of the overall comments.

2.3 **Evaluation**

The only means for evaluating the performance of the Textblob analysis is using the test accuracy and coming up with a confusion matrix to identify the true positives, true negatives, false positives and false negatives. This was not possible with the given dataset as the comments had no labels or pre assigned clusters to test with them.

2.4 Named Entity Recognition model

A Named Entity Recognition (NER) is a natural language processing (NLP) technique that automatically identifies named entities in a text and classifies them into predefined categories.

The NER model was created using a Spacy pre-trained model to identify the entities in a comment.

The spacy en_core_web_lg and en_core_web_sm libraries were loaded. An input field that prompts the user to enter the comment was created. When a user enters the comment, the comment is looked through the libraries to identify entities in the comment belonging to categories: Person, Organization, Place/Location, Geopolitical entity, Time. If there are any entities belonging to the categories, the text and its corresponding entity label are printed out as an output. Fig 2 shows an illustration testing the model with the comment and out putting the named entity.

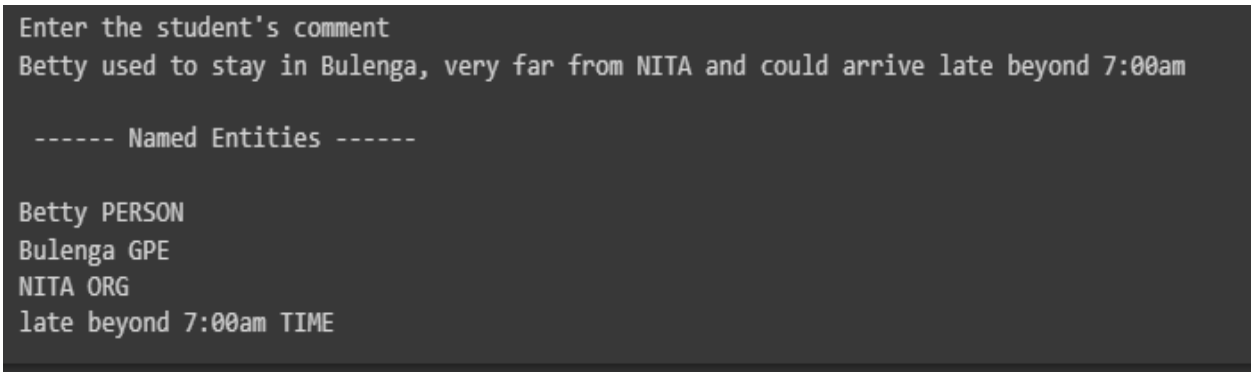

Fig 2: Illustration of inputs and out puts for the NER model

2.5 Visualization showing insights about the dataset

Fig 3 represents the pie chart showing the percentage distribution comments among clusters. To create a D3 visualization, the cluster count obtained from clustering the comments was utilized. and the csv file that was generated was exported to the visual studio code. A pie chart showing the percentage distribution of comments to the five clusters was visualized. This was created using d3, html and JavaScript. The pie chart is interactive with a hover on function whereby if any partition is hovered on, it displays the percentage of comments belonging to each cluster and the cluster name. The visualization was hosted on the previously used server. Attached is the link to the Visualization.

https://kwagalakwe-grace.github.io/KwagalakweDataVisualization/Text_analysis_Visualization.html .

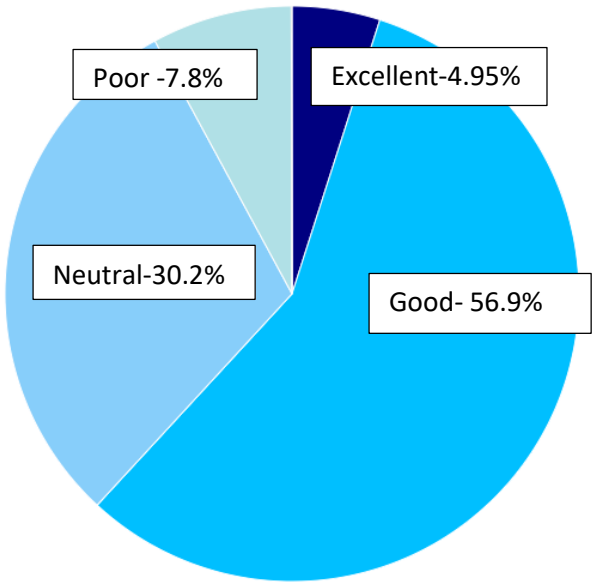

Fig 3: Percentage Distribution of comments to clusters.

3. Discussion of Results

The results from the clustering of comments indicated that majority of comments belonged to the category of good, followed by neutral, followed by poor, followed by excellent and lastly very poor. Table 1 represents the count and percentage of comments belonging to each cluster.

Cluster	Count	Percentage
Excellent	245	4.95
Good	2816	56.92
Neutral	1494	30.20
Poor	390	7.88
Very poor	2	0.04

Table 1: Count and Percentage of each Cluster

The Named Entity Recognition model performs fairly good though since its using a pre-trained model from spacy, some entities cannot be recognized more so the sur names of people and also it hard to recognize some organization names. This is due to the fact that the surnames and organization names are very new to the model and so the model can't identify a category to which they belong.

References

[1] https://www.fams-cit.com/fscomments.

[2] Named Entity Recognition (NER) with spaCy by Sanidhya : https://medium.com/analytics-vidhya/named-entity-recognition-with-spacy- Accessed on 2nd September,2022.

[3] NLTK Tokenize: Words and Sentences Tokenizer by Daniel Johnson : https://www.guru99.com/tokenize-words-sentences-nltk.html - Accessed on 2nd September, 2022.

[4] Sentiment Analysis using Textblob by Parthvi Shah : https://towardsdatascience.com/my-absolute-go-to-for-sentiment-analysis-textblob. Accessed on 9th September. 2022.