Winfred Tibingana

2021/HD05/2313U | 2100702313

Data Visualisation and Analytics | MSCS

# **FIELD ATTACHMENT FEEDBACK ANALYSIS**

Makerere University interns Field Supervisors have been placed in organisations of attachment overtime. Valuable feedback to the University about its students has been provided by the Field Supervisors. Unfortunately, this feedback is in form of \*unstructured text and large in volume\* making it difficult to process and gain useful insights.

This report describes text analysis processing techniques that were applied to get more insights from the dataset at https://www.fams-cit.com/fscomments.

### Dataset Description

The Data Viz4.ipynb notebook was used. After loading the data into a pandas dataframe, the dataset contained 5010 comments with two columns: comment\_id and Comment as of September 18, 2022.

### Dataset Cleaning

* 1 row was found to contain a NA comment. This was removed.
* The Comment column contained 601 duplicate comments. These too were excluded from the data frame that would be used for analysis. The data frame thus contained 4408 comments with the two columns.
* A clean\_sentence function was used to process the text for better analysis. In this function: the text is transformed to lowercase, brackets removed from the text, punctuation removed and lastly numbers removed from the text.

A column of the cleaned text is added to the data frame.

## Create a corpus from dataset

Corpus: a collection of authentic text or audio (text written or audio spoken by a native of the language or dialect) organised into datasets. It is a collection of text on which processing can be performed. A corpus can be made up of newspapers, novels, recipes, radio broadcasts to television shows, movies, and tweets.

The cleaned text column can be considered as a corpus. However, further processing can be done on it. Such processing includes:

* Stemming: This is removing suffixes from words so as to end up with a word stem. Example: “likes”, “likely” and “liked” result in their common word stem “like” which can be used as a synonym for all three words.
* Lemmatizing: this is the process of reducing a word to its canonical or dictionary form. The process is similar to stemming but the root words have meaning. Much as this process takes long when applied, the context of the work is kept.
* Excluding Stopwords: Stopwords are words in any language which do not add much meaning to a sentence. They can safely be ignored without sacrificing the meaning of the sentence. Examples of stopwords in the English language: this, the, to, is

A comment stem and comment\_lema columns are added to the dataframe.

An additional tokenized\_corpus column is also added to the dataframe.

## Cluster the comments of Field Supervisors into categories: Excellent, Good, Neutral, Poor, Very Poor

To cluster the comments into the categories as: Excellent, Good, Neutral, Poor and Very Poor, two approaches were used in the notebook. To categorise the comments as such is a sentiment analysis of how positive or negative a comment is.

**Approach 1: K-Means Clustering with TF-IDF**

Term Frequency-Inverse Document Frequency (TF-IDF) is a numerical statistic that demonstrates how important a word is to a corpus. The TF-IDF extracts particular phrases from a corpus and translates it into a numerical representation, weighted very uncommon or extremely frequent terms differently to give them a low score.



*Figure 1: Convert to matrix of TF-IDF features*

Vectorization technique used by TF-IDF, enables one to classify sentences based on the key phrases that make them up. The *TfidfVectorizer* has parameters which are adjusted to improve the performance in the Kmeans. The following are some of the parameters that were adjusted.

* Max\_df was used to remove terms that appear too frequently which are also referred to as corpus-specific stop words. Terms that appeared in more than 50% of comments were ignored.
* Min\_df was used to remove terms that appeared too infrequently.
* Max\_features was set to None so that the entire corpus was considered.
* Ngram\_range is used to set the lower and upper bounds for different n-grams. The lower bound was set to 1 for unigrams and upper bound set to 2 for bigrams.
* Analyzer is used to extract the sequence of features out of the raw, unprocessed input. It’s used together with the stop\_words parameter. Analyzer is word and stop\_words is english> They are responsible for removing stop words and creating a word vocabulary

*Transformed\_tfidf\_array* as seen in figure 1 is an array of vectors that will be used to train the KMeans model.

K-Means clustering is a method of vector quantization, originally from signal processing, that aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean

The first record constrained a non-english word: djfjkdfjkjkffdk. This comment was dropped from the dataframe so as to avoid unwanted dimensions.

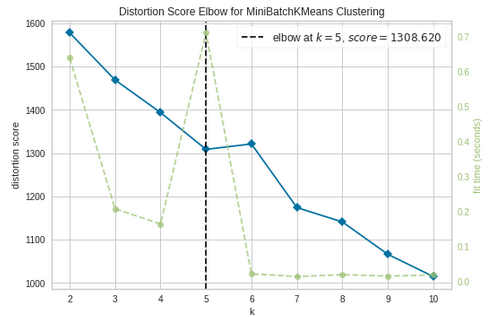
Given the fact that lemmatizing keeps the context, the comment\_lemma column was used.

The K-Means clustering is then applied to the transformed\_tfidf\_array that is returned.

The clusters returned do not make good categorizations.

Evaluate Model Performance Elbow Method and Silhouette Analysis

The Elbow Method and the Silhouette Analysis are techniques used to determine the number of Ks.



*Figure 2: Elbow curve*

The point after which there isn’t a sudden change is when k=5. The vertical line in black in figure 2 above marks the best number of ks as 5.

In the Silhouette Analysis, the best score is one closer to 1.

Also, the distribution of good clusters is one with similar sizes of clustered area. Figure 3 below shows that this isn’t the case. One of the clusters is way larger than the rest plus the points are not well distributed.

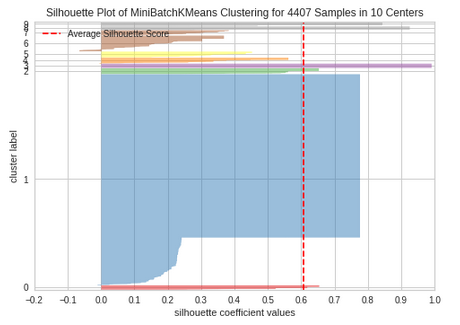


Figure 3: Silhouette Analysis chart

**Approach 2: VADER**

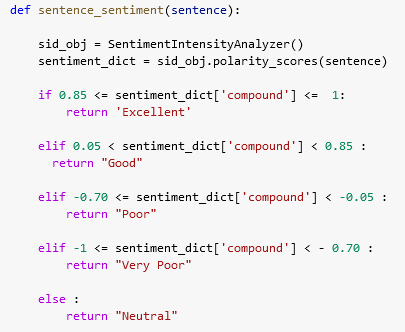
Since the clusters returned couldn’t correctly categorise the comments, I used VADER (Valence Aware Dictionary for Sentiment Reasoning), a model used for text sentiment analysis that is sensitive to both polarity (positive/negative) and intensity (strength) of emotion. It is available in the NLTK package and can be applied directly to unlabelled text data. VADER offers the benefit of analysing the sentiment of any text without the requirement for prior training, which is something that machine learning models could require[1].

VADER is popularly used for analysing unlabelled social media data such as tweets and performs very well.

The result generated by VADER is a dictionary of 4 keys neg (negative), neu (neutral), pos (positive) and compound. Its values are between -1 (most extreme negative sentiment) and +1 (most extreme positive sentiment). The compound score is used to determine the overall sentiment of text. The compound score for a:

* Positive sentiment is ≥ 0.05
* Negative sentiment is ≤ 0.05
* Neutral sentiment is between -0.05 and 0.05

Given the score above, I was able to create a function that could categorise text into any of the 5 categories: Excellent, Good, Poor, Very Poor and Neutral as shown in figure 4 below:



*Figure 4: Function to categorise text.*

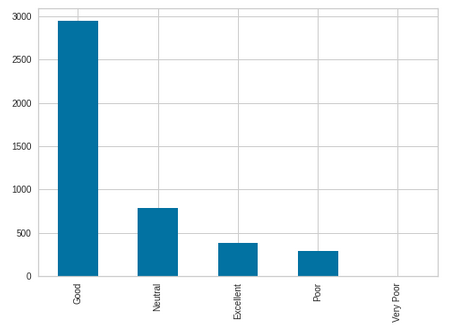
* Excellent between 0.85 and 1
* Good between 0.05 and 0.85
* Poor between -0.70 and -0.05
* Very Poor between -1 and -0.70
* Neutral between -0.05 and 0.05

**Results:**

|  |  |
| --- | --- |
| **Category** | **Count** |
| Excellent | 385 |
| Good | 2945 |
| Neutral | 785 |
| Poor | 290 |
| Very Poor | 3 |
| **Total** | **4408** |

Table 1: count of comments based on categories.

Figure 5 below is a graph of comment distribution based on the categories determined by VADER.



*Figure 5: Vader categorization of comments*

The details results can be found in the sentence\_sentiment column of the vader\_comment\_sentiments.csv attachment.

Create a Named Entity Recognition (NER) model

NER Model that takes in a comment as an input and outputs the Entities, if any, belonging to the categories: **Person, Organisation, Place/Location, Time**

An NER is a sub-task of information extraction that seeks to locate and classify named entities mentioned in unstructured text into predefined categories such as person names, organisations, locations, medical codes, time expressions, quantities, monetary values, percentages, etc [2].

NER is used in Natural Language Processing (NLP) to answer a plethora of questions such as which companies are mentioned in an article, or even specific products for a company.

In this notebook, SpaCy was used for this task. SpaCy is an open-source library for advanced NLP. It may be used to build information extraction or natural language understanding systems, or to pre-process text for deep learning [3]. It can provides a model which can recognize a wide range of named or numerical entities,

A function (NER\_Model\_ftn) was created that takes text and returns all the entities in that text. It’s label\_ function was used. Figure 5 shows sample output.

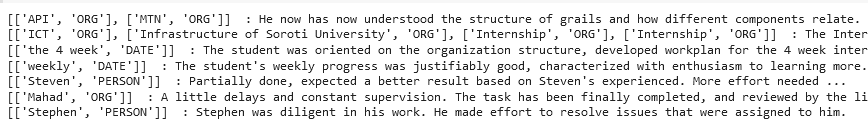


Figure 5: Sample SpaCy output.

This feature can be tried out here: <https://effortless-cocada-9481fe.netlify.app/>

Visualisation

A word cloud was created from the lemmatized comments.

A word cloud is where the often-used terms are displayed in large text, while the least frequently used words are displayed in tiny print.

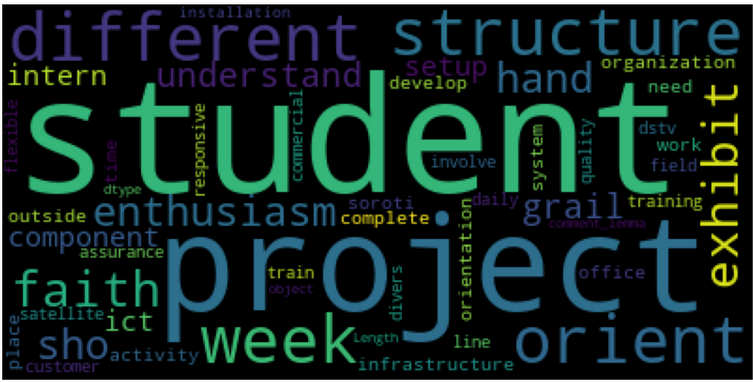


Figure 6: WordCloud

N-grams are other visualisations that were created to get more insights from the data. An N-gram means a sequence of N words; a continuous sequence of words or symbols or tokens in a document.

Example:

sentence = 'I came by car'

Tokens:['I', 'came', 'by', 'car'] # 4 tokens in sentence

unigrams = 'I', 'came', 'by', 'car'

bigrams = 'I came', 'came by', 'by car'

trigrams = 'I came by', 'came by car'

Details can be found here: <https://effortless-cocada-9481fe.netlify.app/>

REFERENCES:

[1] <https://towardsdatascience.com/social-media-sentiment-analysis-in-python-with-vader-no-training-required-4bc6a21e87b8>

[2] <https://en.wikipedia.org/wiki/Named-entity_recognition>

[3] <https://towardsdatascience.com/custom-named-entity-recognition-using-spacy-7140ebbb3718>