National College of Ireland

Text Analysis of Russia and Ukraine War

MSc Research Project
MSc in Data Analytics

Alex Greenmount James
Student ID: x21110654

School of Computing
National College of Ireland

Supervisor: Mr Michael Bradford

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Alex Greenmount James
Student ID:	x21110654
Programme:	MSc in Data Analytics
Year:	2022
Module:	MSc Research Project
Supervisor:	Mr Michael Bradford
Submission Due Date:	01/02/2023
Project Title:	Text Analysis of Russia and Ukraine War
Word Count:	6,596
Page Count:	19

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	1st February 2023

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Text Analysis of Russia and Ukraine War

Alex Greenmount James
x21110654

Abstract

Sentiment analysis has a wide scope on various industrial applications. It has been used on applications including the user feedback support and product surveying. This project focuses on the various machine learning techniques used for sentiment analysis like Artificial Neural Network (ANN) and Random Forest (RF). The project starts with a collection of tweets based on the Ukraine and Russia war pattern using various hashtags and keywords. The collected data is being preprocessed to remove the unwanted hashtags and characters. The processed data is being tokenized and fed to the sentiment analysis tool under the Natural Language Toolkit (NLTK). The library will extract the features and achieve the polarity score and subjectivity of the tweets. The tweets are then arranged with the scores and polarity and fed for the neural networking and machine learning models created. For the machine learning libraries sklearn, keras, pandas, numpy, nltk etc are utilized. The machine learning algorithms used are Random Forest and ANN with Term Frequency-Inverse Document Frequency (TF-IDF). The data results are being compared with the NLTK reference. The data fed through the TF-IDF is given to both Random Forest and ANN systems. The results show that the ANN can have a better correlation with respect to the NLTK analysis. The ANN had a 10-percentage increase in the prediction capability compared to the Random Forest machine learning.

1 Introduction

"There were 5 exabytes of information created between the dawn of civilization through 2003, but that much information is now created every two days." Eric Schmidt, Executive Chairman at Google. Technology is being drastically increasing with tremendous speed. The driving force for the advancement is taken care of by data. Huge amount of data is being gathered and analyzed for various applications. It has become the root for most of the processes. Considering a typical example of the grocery, store position of a particular product is changed according to the customer feedback. It will increase their profit margin. The usual practice of getting the feedback from the customers is through the feedback form , which may be a big burden for most people thus reducing its credibility(Raschka; 2016). Here comes the advantage of sentiment analysis and natural language processing tools. This will help to make the predictions according to the people's sentiments, which is much more accurate.

This thesis paper focuses on the sentiment analysis capability of the Random Forest machine learning and ANN with comparison to the NLTK. The main method recognized to evaluate the sentiment analysis of any text is based on the Natural Language Toolkit

(NLTK)(Thakkar et al.; 2022). It has various libraries that contribute to defining the sentiment and various other analysis methods. The scope of the project lies on the evaluation ability of the RF and the ANN acting as the proxy machines in place of sentiment analysis engine. NLTK libraries always provided the sentiment analysis capability with inbuilt algorithms; it can be correlated and studied using various other machine learning algorithms.

Here, the thesis focuses on the sentiment analysis part of the Ukraine-Russia war. The fundamental way to find the sentiment of the tweet was to formulate a natural language processing algorithm to the model. Here the NLTK libraries were used to define the sentiment of the various tweets which are collected. The NLTK uses various techniques to calculate the sentiment of the tweets(PERKINS; 2017). The detailed discussion on the NLTK is given in this paper. The next step is to evaluate the same with the proxy scheme. Here Random Forest and ANN models were created, and then the tweets were divided into test and train data sets, then the training data sets was used to teach the system and tested the model with the testing data.

The platform used for the collection of data was the Twitter platform. The tweets collected, were gathered using various hashtags and keywords. The collected data sets were cleaned using various pre-processing tools. The cleaned data was processed using various methods like lemmatization and stemming. Then the data was tokenized to form the background of analysis. The tokenized data was used to create the sentiment data from the tweets. These datasets were fed to the TF-IDF for vectorization and to machine learning models.

The two main models used for the analysis purpose are the ANN and the Random Forest machine learning algorithms. Various tuning parameters are being used on the evaluation of these two approaches. The vectorized data from the TF-IDF is given to both models where a portion of the set is being created as a training dataset and the other is made as a testing dataset. The results are compared with the NLTK and analysed. The execution time for each model creation is also evaluated for the comparison.

2 Research Question

- Based on the polarity scores/classification labels generated by the NLTK sentiment analysis engine, operating on the tweet data of Russia and Ukraine war, to what extent can a set of proxy machine learning schemes replicate the scores of the original engine?

- Are there processing advantages in the execution time taken to produce the result in the proxy schemes rather than the original sentiment analysis engine?

3 Literature Review

Machine Learning models have dominant advantages over many fields including marketing and many others. The sentiment analysis part of the online social media data can affect many decisions making and understanding of human behaviors with patterns of prediction. This is considered as a significant contributor to natural language processing. This literature review mainly focuses on the various parts of the thesis that will be used along the path. The various subcategories in the system are the preprocessing tools. Here

various preprocessing tools, implementation of machine learning models and evaluation of those models are explained. The filtering methods are identified from the literature and are given to the system. Other major works include the emojis that are significant in determining the sentiment of the tweets. It also mentions about data preprocessing techniques like the lexicon method, and vectorization methods like the TF-IDF. Then comes the machine learning models' introduction. Here the focus is given to the Random Forest and ANN methods. Lastly, the challenges are also mentioned. The research uses different libraries such as pandas, numpy, keras, sklearn etc.

3.1 Filtering Methods

Some of the major techniques used for the definition of data extraction can be taken from the literature papers. For the emoticons, the positive, negative, and neutral parts of the general emoticons are defined in a dictionary. For example, pleasant expressions include 'ROFL,' 'LMAOL,' 'LMAO,' and 'LMAONF', which create a sentiment, can be classified, and stored in a dictionary. Slang corrector, here the slang of the tweets has been considered and stored like btn.tnx etc. Onomatopoeic expressions, here the Onomatopoeic expressions: like 'wow' (POSEXPRESSIONS) and 'bleh' (NEGEXPRESSIONS) are stored and classified in mapping dictionaries(Pozzi et al.; 2013).

Like the above parameters adjectives and stretched words were also considered. The data was preprocessed by removing the URLs, hashtags mentioning tags, and retweet symbols. A spell checker is also used to correct the words(Pozzi et al.; 2013).

The four main machine learning models that can be used on the dataset are Naive Bayes (NB), K-Nearest Neighbors (K-NN), Support Vector Machine (SVM) and Decision Trees (DT). The experiment was conducted in such a way to evaluate the effect of the polarity contribution of each parameter like emoticons to the overall statement polarity. The results collected from the literature paper showed that the model was able to predict the various characteristics with higher accuracy. The data was given without any preprocessing and experimentation. The assumptions were taken at emoticons and adjectives greatly affect the emotions of social media posts thus the results also show that the accuracy of the machine learning model was increased by almost 5 percent overall(Murthy et al.; 2020).

3.2 Emojis

Emojis play a major role in data transfer. We often use sarcastic words, and they can be only identified using the emojis expressed. Thus, analyzing and interpreting the emojis are very much important to analyze the text data for the sentiment analysis Emojis cannot be directly converted to the textual data. Thus, they are being converted to Unicode. Unicode is a method that is being universally accepted to encode data for the analysis purpose here the textual data is also converted for the same in analysis [1]. The Unicode generation is implemented using special libraries. Unicode encoding is applicable in most of the languages it is being widely used in the natural language processing system [2].

Python has an inbuilt library that helps to sort the emojis conversion using the emoji library the symbols can be easily converted to the Unicode's and later it can be repres-

[1] https://unicode-table.com/en/

[2] https://stackoverflow.com/questions/47716217/converting-emojis-to-unicode-and-vice-versa-in-pyth

ented via a textual format also. This will be very much helpful on the data processing and analysis part(Hönings et al.; 2022).

Table 1: Common Locale Data Repository (CLDR) shortname and unicode.

CLDR Short name	**Unicode**
Grinning face	U+1F600
Grinning face with big eyes	U+1F603
Grinning face with smiling eyes	U+1F604

The above Table 1 briefly provides a general idea of the sample of the Unicode and the emojis. It can be also used to create emojis either by the words or by the specific Unicode. The emokize function from the emoji library is used for the creation of emojis(Ayvaz and Shiha; 2017).

3.3 Preprocessing

The raw data collected from Twitter should be cleaned and processed to get the proper data extraction. Various stages that are used for the cleaning process include the

- Removal of unwanted hashtags and other notions like special characters and blank spaces.
- The letters are converted to lowercase.
- The emojis are converted to words with the emoji library.
- The stop word library is used to remove the unmeaningful words from the corpus.
- The empty spaces are removed.
- Non-English words are filtered and removed.
- Using lemmatizer and PorterStemmer the words are converted to the base form[3].

3.4 Lexicon Methods

The data that are present in the online social medias like Twitter are highly unstructured. This is very much important to analyze. The main approach is sentiment analysis which can be correlated to the emotional portion of the texts.

Two major approaches found from the paper were machine learning algorithms and lexicons(Kharde and Sonawane; 2016). A hybrid method analysis is also mentioned. The methodology of the paper is sutured with the preprocessing of the data where the unwanted parts of the tweets are removed and processed. The later part will be the feature extraction like the features related to the sentiment are analyzed and discussed. Then comes the architecture of the machine learning model. The tweets are divided into training and test subjects then the analysis part down on the system. The first machine learning model introduced is Naive Bayes which is a probabilistic classifier. Python

[3] `https://www.repustate.com/blog/data-cleaning-in-sentiment-analysis/`

NLTK libraries are used for this. The next model discussed is the maximum entropy model where no presumptions are taken. The system tries to create a random regression model in which it creates the relationship with the various features extracted. A logistic regression method is adopted here. The next approach in the machine learning model is the support vector machine in which a selection decision matrix is created, and kernels are used to create the input space. This method typically uses two data sets, and it can be further applied to regression and classification models also. The model is trained with the above approach and the analysis is done either with supervised learning or unsupervised learning. the approach helps to evaluate the tweets' relationships. The next main topic apart from machine learning is the lexicon method where the approach is like creating a relationship with the already existing words and their meaning and using it to create a sentiment analysis on the tweets. It works on the inbuilt dictionaries in the python libraries(Kharde and Sonawane; 2016).

On the analysis part it is seen like the lexicon has only 76 percent of the accuracy whereas the machine learning model has accuracy of more than 80 percent in which the support vector machine has the highest rationing 86 percent. Various levels of the sentiment analysis done are discussed later that includes the word level, document level, sentence level and feature level. Feature level is the highest one. The next concept is the unigram and bigram where the bigram represents the relationship between two words thus the accuracy of the different model on uniform and bigrama are evaluated next. In the analysis the results were like Baseline :73.65 percent, Naïve Bayes 74.56 percent, SVM 76.68 percent, Maximum Entropy 74.93 percent(Medhat et al.; 2014).

On further literature review two major techniques that are much useful for sentiment analysis were found. The first one is based on obtaining the sentiment of each word and taking the combined text sentiment which is known as word-based sentiment analysis and later the sentence is used for the evaluation. Here the whole text is taken, and this is much more effective since individual words may not be applicable in certain cases. This approach is very effective for the analysis of data with very small data sets. The major problem this paper addresses is the polarity shift from the word level approach to the sentence level. The methodology of the proposal is done in such a way to find the polarity shift from the word wise approach then the sentence polarity and then using the algorithms a feature-based approach is taken to have a common polarity considering these both. On the word wise model, they are capturing 2 different parameters: the first one is the sentiment of each word and the polarity shift considering other words. The output will be the sentiment of the words and the polarity shift rate. The next will be the sentiment of the sentence, which is classified into positive, negative and neutral accordingly. In order to combine and evaluate models a hybrid approach is taken into account. Bag-of-words, n-grams or dependency trees etc. are some of the algorithms that can be used in the hybrid model[4]. For the experimental analysis 2 sets of data from the customer review and movie review was considered. Various models were taken into consideration. The hybrid model showed a higher accuracy compared to the sentence wise and word wise approach. The concussion also says that as the data set number increases it is seen like the increase in the accuracy of the model(Ikeda et al.; 2008).

VADER refers to Valence Aware Dictionary and Entiment Reasoner. This is a very powerful tool in natural language processing, and it can be used to evaluate the tweets sentiment using the inbuilt dictionary functions[5]. VADER is a lexicon-based sentiment

[4]https://www.sciencegate.app/keyword/9211

[5]https://www.researchgate.net/publication/275828927

analysis tool that helps to evaluate the sentiment of the tweets. It analyzes the words orientation and feature extraction methods to evaluate the effect of these. But the comparison of this technology with the machine learning technology shows a reduction in the accuracy of the system. It is more preferred to model the system using machine learning algorithms[6].

3.5 TF-IDF (Term Frequency-Inverse Document Frequency)

TF-IDF (Term Frequency-Inverse Document Frequency) is one of the most used methods in natural language processing and machine learning. Here, this method is used to define the importance of a particular word in a corpus or series. This will help in evaluating the importance of the words in the text. It is generally useful in the data mining and textual identification methods like sentiment analysis. The various terminologies associated with the TF-IDF techniques are term frequency, document frequency and inverse document frequency. Normalization is applied to these modifications done in the document(Prathyusha and Reddy; 2021). This is a very useful methodology.

3.6 Machine Learning

Classification method in machine learning, especially the Random Forest, is a well-known method to determine the relationships between data and as a tool on various evaluation processes like sentiment analysis. The classification machine learning algorithms are used to evaluate the performance of the system under classification method. Here it's a discrete method where the output will be discrete classifications. One of the most used methods under classification are decision trees and Random Forest. Random Forest can be defined as a combination of the decision trees[7].

3.7 Random Forest

The understanding of the Random Forest method will be clearer with the decision tree method. In the decision tree method, we use 3 major elements the root, leaf and the branches each item is subdivided until it reached the leaves thus the branches with the nodes will be used to create the logic to make the reduction one of the biggest problem associated with this approach is the inaccuracy in the prediction due to the requirement of hyper parameter tuning. Also, the curve fitting is not very effective in some cases in decision tree methods. In contrast, the Random Forest method is a combination of many decision trees that helps to eliminate the error and make more accurate predictions(Medhat et al.; 2014).

Some of the major advantages of using Random Forest machine learning algorithms are the good prediction capability and higher accuracy with even missing data. Random Forest is being applied in many major fields like the stock market, real estate online shopping sector etc. One of the major limitations in the Random Forest machine learning system is the inaccuracy of the dataset produced by constant online data feeding. On some systems the model is being updated continuously and the system should be able to make modifications to the change. The major drawback is the time taken for the changing the system. Since the time consumption is higher it is not majorly used for

[6] https://www.jetir.org/view?paper=JETIR2005456

[7] https://www.researchgate.net/publication/236952762_Random_Forests

dynamic models. Also, the algorithm is more complex compared to the other machine learning algorithms(Karthika et al.; 2019)

3.8 Artificial Neural Network (ANN)

Artificial neural network is a machine learning algorithm that helps to predict and analyze the systems. It was made with the inspiration from the biological systems. It works similarly to the neurons in the human body. On the neural network its world with the help of various interconnections. It will have mainly three-layer systems: the input layer output layer and the hidden layers. The number of nodes on the input and output can be adjusted according to the requirement.

In ANN, a hidden layer is a layer between the input and output layers wherein artificial neurons take in a collection of input nodes and generate a result via an activation function. The various points are connected by nodes. The nodes will be defined by a weighting value also. The way in which the system works is like an input will be given to the system taking the hidden layers weights and some initial values, then the output expected is compared with the real output, and the weights are adjusted to get the optimum results. Here the function used to define the weighting adjustment is loss function where it tries to reduce the error to a minimum value. The process is known as backpropagation(Wang; 2003).

3.9 Challenges

The major challenges that were faced are:

- Sarcasm detection is limited by the models.
- Most of the tweets are depended on the domain in which the statement is done.
- Some of the text has a small part of the text that contributes to the polarity of the text.
- The text may also depend on other topics which will only determine the sentiments thus making it more difficult to determine the sentiment.
- Some tweets are comparison based, thus the lack of reference make it difficult to analyze.
- Some tweets depend on the subject level, others on the objective level. Thus, the model should address these also

4 Implementation and Design Methodology

The implementation steps carried out in this research is shown below:

- Collecting the data from Twitter developer API.
- Preprocessing the data collected.
- Sentiment analysis is done with the NLTK.
- TF-IDF vectorization is done with Random Forest model and ANN to compare with the sentiment analysis engine.

The implementation approach undertaken for this research is shown in Figure 1 below.

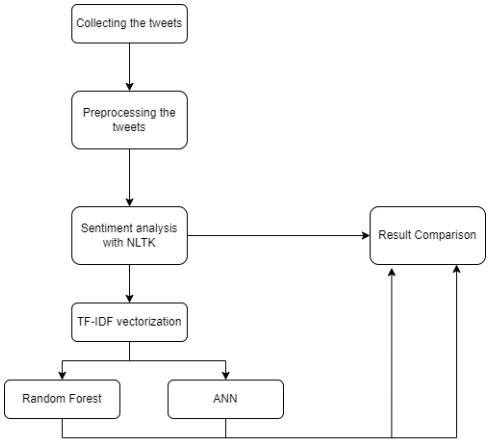

Figure 1: Implementation approach

4.1 Tweet Collection

The objective of the project is to evaluate the performance of the machine learning models to evaluate the sentiment analysis of the tweets; thus, the primary objective was to collect tweets. Twitter provided APIs to get various tweets using different hashtags. A developer API account was created, and using the elevated access, tweets were collected from the Twitter account. 5,39,868 tweets were collected and processed by using Twitter developer API and also Python as a toolkit. After the processing the count of positive tweets, count of negative tweets, count of neutral tweets, subjectivity and polarity were all included in the dataset. The polarity of the tweets were determined by a built-in python library called TextBlob. The tweets were collected based on the Ukraine and Russia war. The various hashtags/keywords used for the tweet capturing are:

- Zelensky Russia.
- Ukraine crisis
- UkraineNato
- Ukraine Russia conflict
- Ukraine antiwar
- Ukraine war
- Russian war
- Russian invasion
- StandWithUkraine
- Russia invade

4.2 Twitter Data Collection

For the sentiment analysis the data collection(tweets) was the most initial step taken. Here the data collection was done mainly through the API given by tweeter. The initial step is to create a Twitter developer account which will help to access the API once the account is created then it can be elevated to get the API download access [8]. Once the account is activated the following credentials can be generated;

- Secret key
- API key
- Bearer token
- Access Token

These are used for accessing the data from the Twitter account. In python a special library called tweepy was used to do the data extraction. Using the credentials tweepy library can be used to access the Twitter account. Once the account is activated using the keywords and hashtags specific tweets can be sorted. Sorting parameters are mainly the number of tweets, date, author etc.. By using these the data filtration can be done.

4.3 Snscrape

Due to the limited number of data extraction permission from the Twitter credentials an alternative method was also implemented for collecting large amounts of tweets. Using the regular credentials only 2 hundred thousand (2,00,000) tweets were only able to be extracted. Using the snscrape library it was possible to implement the tweet pulling code to extract a much larger number of tweets.

Another important library used is the pandas library. Here the data was taken into different tweets and the data frame was created and saved in the csv format. It helped a lot in the data processing time. The various parameters stored were the tweets and key words. 50,000 tweets were taken at a time due to huge time constraints. In total, 5 hundred thousand(5,00,000) tweets were collected with 10 different keywords with respect to the Ukraine and Russia war. The duplication in the tweets were verified and the NaN files and was deleted. Then the tweets were combined into a single file and exported as the main file.

The developer account was created from the Twitter platform given access to collect datasets from various accounts [9].. The above tweet keywords were used to gather data from various Twitter accounts. The data collected were cross verified and stored.

The below Figure 2 shows distribution of words in the tweets the majority is on the Ukraine and Russia war. 5,39,868 tweets were collected for the analysis. The repeated tweets were sorted and eliminated. The most used unique words were found from the tweets after the tokenization process.

[8] `https://developer.Twitter.com/en/docs/Twitter-api`
[9] `https://developer.Twitter.com/en/docs/Twitter-api`

Figure 2: Word cloud showing the distribution of words in the tweets

4.4 Preprocessing the Data

Preprocessing is the step that is followed prior to the data processing. This mainly involves the data cleaning processes. Various filters are used for the implementation. Some of the typical cleaning filters used are, removal of the unwanted hashtags and other taglines, the unwanted spaces are removed, the unwanted numbers are also removed. The emojis are converted to the equivalent binary form. The stop word library is used to eliminate meaningless words. Porter stemmer library is being used to eliminate the randomness in the words. This helps to bring the words to the base form thus reducing the calculation complexity. The next is to use the lemmatization process. The main difference between the lemmatization and stemming is that even if both converts the word to base form the lemmatization also considers the context of the sentences. The nil columns are also eliminated from the file. The later cleaned dataset is given for the tokenization process the various tools used for preprocessing are listed below.

- Converting all letters to lowercase.
- Removing words other than English.
- Removing all numerical components.
- Converting emojis to sentiment readable data.
- Eliminating all NaN statements from tweets.
- Eliminating repeating tweets.

Various techniques like TF-IDF weighting scheme, stemming, stop-words removal, tokenization have been used for the preprocessing of the data(Krouska et al.; 2016).

4.5 Sentiment Analysis

It is very much important to create the dataset that can be used to make the machine learning model. The main agenda of the project is to evaluate the sentiment analysis models using machine learning platforms. Thus, the base model is being created using the natural language processing libraries.

TextBlob library was used for text processing in a straight forward manner to evaluate the polarity the of the tweets. TextBlob is a built-in library in the NLTK. It provides a user-friendly interface to the NLTK library. The library returns two values: polarity score and subjectivity. The polarity defines the sentiment of the tweet to be a value ranging from -1 to 1 and subjectivity refers to the subject's content dependency of the tweet. The tweets were classified into positive, negative and neutral according to the polarity value.

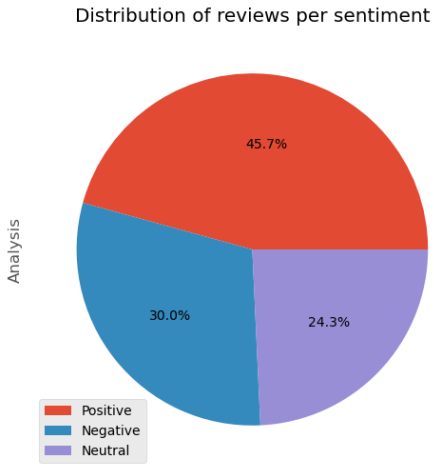

Figure 3: Distribution of reviews per sentiment

The above Figure 3 shows the distribution of the data in positive, neutral, and negative aspects. This sentiment analysis data will be used to train the models [10]. Various libraries used for the sentiment analysis are TextBlob, pandas, numpy, matplot etc.

Number of tweets taken for the analysis: Due to the memory error in the system used for the computation, the tweets given to the Random Forest and ANN were limited to 2 hundred thousand rows in count.

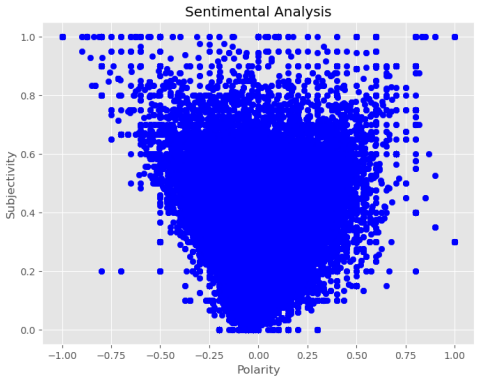

Figure 4: Sentiment analysis plot of the polarity and the subjectivity

[10] https://techvidvan.com/tutorials/python-sentiment-analysis/

4.6 TF-IDF Vectorization

The next main step followed in the process was to implement the vectorization on the dataset. The vectorization was done using the TF-IDF library.TF-IDF is a very powerful tool while analyzing the textual data. Here the repetition of the words is analyzed and evaluated to create a probability approach. It uses two methods: term frequency and inverse documentation frequency. In term frequency the number of times the word is being used is identified and classified. In the next step the relationship if the word with respect to the document is analyzed and quantified. This will help to evaluate and characterize each word in context of the content(Alfarizi et al.; 2022).

On the sentiment analysis performed the data is being vectorized using the TF-IDF method and given to various machine learning approaches like Random Forest ANN etc. The accuracy can be increased by increasing the amount of data in the sequence. One of the alternatives that can be used instead of the TF-IDF is the bag of words approach. But usually even if it's very simple to interrupt it's generally not used for machine learning approaches(Hung et al.; 2022).

4.6.1 TFDIF in Random Forest/ANN

TF-IDF as discussed is for the data extraction The TF-IDF makes the words content relationship to be converted into a vector format. This will help to evaluate the system more efficiently. At first the TF-IDF library is imported from the sklearn and the instance of TF-IDF is taken. The tweets that are preprocessed are fed to the TF-IDF with the required processing parameters and the output will be the machine readable/ processable form. Lastly it will be fed to a Random Forest model/ ANN model.The various parameters are:
-max_features: As already discussed the model will be created using the vectorization of the text in relationship with the context thus the feature extracted will be limited by the variable provided in max features. Taking the example of the sentiment analysis conducted, the value taken is 2000. The top 2000 features will be taken for the analysis based on the systems memory capability.

-min_df and max_df: These values act as a filter. The values with the range in between the max and min will be only taken. Frequency value will be used for this kind of analysis. This will help in making the analysis more efficient.

-stop_words: This is also used as a filter. Here the typical string passed is English the generally commonly used words will be also eliminated. Taking the example of verbs its frequency will be very high, but it does not contribute to analysis thus it can be either filters using the stop words or by using the max df function.

4.7 Random Forest

Random Forest is a supervised machine learning algorithm used in various applications in the current scenario the regression and classification models were used for the analysis of the sentiment of tweets.

On the classification approach the preprocessed model was fed to the classification Random Forest model. The expected outputs were neutral positive and negative; the explicit values of the 3 outputs were taken using the natural language library and the

data was fed as the training and testing datasets.
-n_estimators: It indicates the number of trees in the random forts or roughly the number of iterations like epochs in ANN. Here 100 is the selected value which is sufficient for the proper convergence of the model.
-random_state: It is used for defining the randomness of the model If it's true that on each iteration the training dataset will be taken, and the testing set will be utilized for accuracy validation.

4.7.1 Testing in Random Forest:

Using the sklearn library the random forest classification model is imported and the training dataset is given for training. Once the model is done a prediction model is created using the test data. Both the test data from the classification model and the NLTK library are compared for getting the accuracy. Here, the training and testing datasets are split into 80:20 respectively.

The f1-score obtained for the negative neutral and positive tweets after the testing are 91 percent, 86 percent and 90 percent respectively. Also, the f1-score of the macro avg and weighted avg is 89 percent. The precision of of the negative, neutral and positive tweets are 91 percent, 86 percent, 90 percent respectively.
The accuracy of the Random Forest machine learning model was found to be 88.63 percent.

The result from the Random Forest analysis is shown below in the Figure 5.

	precision	recall	f1-score	support
Negative	0.91	0.82	0.86	11193
Neutral	0.86	0.96	0.90	12776
Positive	0.90	0.88	0.89	16031
accuracy			0.89	40000
macro avg	0.89	0.89	0.89	40000
weighted avg	0.89	0.89	0.89	40000

0.888625

Figure 5: Results of Random Forest model

4.8 ANN

The final machine learning model used is the ANN or the artificial neural network model. The data collected has been subdivided into 2 major sets for training and testing. The proportion taken for the training and testing are 80:20. This will ensure to have a good amount of data to be used for validation. The selected ANN model was framed to run 100 epochs. In a sense, epoch are the number of times the training dataset will pass for optimizing the results(Sandagiri et al.; 2020). It is being widely used with the nanotechnology and biomedical research. It helps create an ability for the system to make it learn self. This will help to make automation more effectively(Abiodun et al.; 2018) .

4.8.1 Structure of an ANN

Here a simple neural network model was defined and used for the purpose. Keras model is used for the implementation of the code. The various parameters are listed below.
• The model used is : sequential
• Activation : Relu and Softmax
• Loss='categorical_crossentropy'
• Optimizer='adam'
• epochs = 100

-**Validation split:** Validation split is a parameter that is being used in the model fitting. It takes values from 0 to 1 and the entire training data set will be subdivided into the split ratio. For example, for 0.2 validation split, for a total of 2,00,000 training data 1,60,000 data will be used for the training; the balance will not be trained and after each epoch it will be used to evaluate the trained data. Here indirectly the data is splitted into testing and training thus the accuracy can be inferred.
-**Optimizer:** An optimizer is used in the optimization phase of the model. There are various optimizers used. Adam, Ftrl, Adagrad, Nadam are some of the generally used optimizers. Here the selected algorithm is based on Adams optimizer[11].
-**Epoch:** Epoch is the number of time step data will be iterated. As the number of epochs increases it gives more time for the model to converge. The taken value here is 100 epochs. The model was tuned for the optimized results. But once the model is being converged even if the epochs are high, it doesn't make any significant differences.
-**Loss:** It can be a string or an instance. It is generally used on identifying the deviation/ accuracy rate of the training. It is helpful on the back propagation of the model. Here categorical cross entropy is used as the loss function.

4.8.2 Testing in ANN

The validation split ratio specifies the training and test datasets. Here it's divided into 80:20 for training and test data respectively. The system automatically takes the training datasets and trains the model and gives the training accuracy value. For each of the epochs it also calculates the accuracy with the test dataset. That will be given as validation accuracy.

ANN - Test Classification Report

	precision	recall	f1-score	support
0	0.83	0.89	0.86	12691
1	0.86	0.84	0.85	16144
2	0.85	0.79	0.82	11146
accuracy			0.84	39981
macro avg	0.84	0.84	0.84	39981
weighted avg	0.85	0.84	0.84	39981

Figure 6: ANN test results

The Figure 6 above shows the test classification report of the ANN. It is seen that the test accuracy of the ANN is 84 percent.

[11] https://www.tensorflow.org/api_docs/python/tf/keras/optimizers

5 Results and Evaluation

The main approach of the thesis was to predict the changes in the output accuracy of the sentiment analysis using various machine learning models. The selected two major approaches were the ANN and Random Forest methods. The results obtained from the study shows that the Random Forest has a better result compared to the ANN. It was also seen that the accuracy increased with higher datasets. On the analysis perspective 5 hundred thousand (5,00,000) data was collected from the Twitter account based on the Ukraine Russia war. For analysis/comparison of the scores between the RF and ANN proxy schemes with the NLTK sentiment analysis engine 2 hundred thousand (2,00,000) data was used. The RF model was able to get an accuracy of 88.86 percent.

In the ANN, the training dataset was able to get an accuracy of 96.7 percent and almost 84.5 percent in the validation (testing) dataset.

The loss function also shows similar results. For the 100 epochs in the training dataset, we have an error loss of less than 0.2 and in validation (testing) it's less than 2.5.

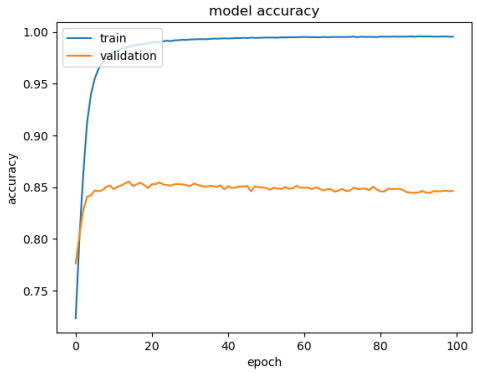

Figure 7: Model accuracy

The above Figure 7 shows the model accuracy for the train and validation(test) datasets with 100 epochs.

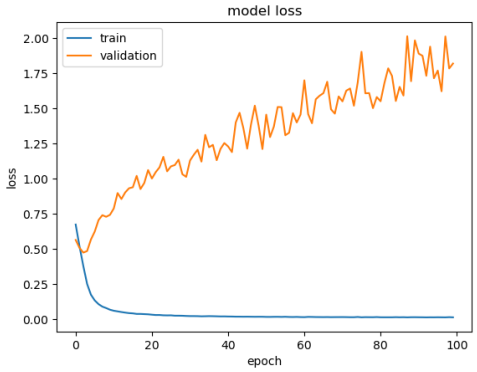

Figure 8: Model loss

The above Figure 8 shows the model loss for the train and validation(test) datasets with 100 epochs.

```
ANN - Train accuracy: 0.967
ANN - Test accuracy: 0.845
ANN - Train Classification Report
              precision    recall  f1-score   support

           0       0.96      0.98      0.97     51190
           1       0.97      0.97      0.97     64277
           2       0.97      0.96      0.96     44454

    accuracy                           0.97    159921
   macro avg       0.97      0.97      0.97    159921
weighted avg       0.97      0.97      0.97    159921

ANN - Test Classification Report
              precision    recall  f1-score   support

           0       0.83      0.89      0.86     12691
           1       0.86      0.84      0.85     16144
           2       0.85      0.79      0.82     11146

    accuracy                           0.84     39981
   macro avg       0.84      0.84      0.84     39981
weighted avg       0.85      0.84      0.84     39981
```

Figure 9: ANN train and test results

The Figure 9 above shows the train and test classification report of the ANN model evaluation. The accuracy of the train and test dataset is also determined.

Table 2: Execution time taken for testing the models and predicting the accuracy.

RF	**ANN**	**Sentiment analysis**
30.95s	8.81s	175.47s

The above Table 2 shows the execution time taken for the completion of the model creation and prediction with the accuracy by each model. The testing time of RF is more time consuming when compared to the ANN. The accuracy level for the Random Forest model is better compared to the ANN. But compared to the NLTK, RF took 30.95s. and ANN took 8.81s for testing datasets.

6 Future Work

The Main focus of the experiment was to predict the sentiment of the tweets using 2 different machine learning models, mainly by ANN and Random Forest. Various improvements can be incorporated in the future work on this paper. Some of the major improvements that can be incorporated on the future are listed below.

- **Keras tuning:** The artificial machine learning model used here is without proper tuning. The System will be able to produce more accurate results when its tuned prosperity and the results will be more reliable[12]. Keras is a hyperparameter tuning system where the library provides a wide range of options to be used for the proper tuning in various parts of the code like the model. Preprocessing post processing etc. The main tuning libraries available in the keras are Random Search, Hyperband, Bayesian Optimization, and Sklearn. Another advantage of using the keras model is the ability to be used in various visualization tools. The error and the loss functions can be properly modeled in the system for reference here.

[12] `https://keras.io/keras_tuner/`

- **Considering the live data model:** In the current system 2 hundred thousand (2,00,000) tweets were taken and the model was created using it and the predictions were done. But it will be more effective if the model is live. The live model is possible by using the live data and constantly dynamic model created in ANN. This will be more accurate, but the runtime will be very high compared to the traditional models.

7 Conclusion

The research focused on the analysis of the ANN and Random forest models' capability to match the scores of the NLTK . Both results from the ANN and RF were compared with the NLTK . The results from the evaluation shows that the ANN model better comparitively with respect to the RF model. But the time of computation of the ANN is much higher compared to the RF model.

Many challenges were mentioned in the initial stage of the thesis, many were tried on addressing in the proposal. The sarcasm of these tweets are mainly dependent on the emojis and emojis greatly define the emotions and sarcastic behavior. Here, importance has been given to the Twitter emojis and the tweets and thus utilizing the emojis makes sure the sarcasm is considered. But some tweets depend on the additional reference and interconnected meanings. Those are not addressed here.

The execution time of the ANN model is 8.81s and the RF model is 30.95s for the test datasets. By this, we can conclude that RF is more time consuming for the testing and ANN is faster. The time consumed by ANN for training the datasets is 9255.03s for 100 epochs. For training the datasets ANN can be more time consuming.

The accuracy of the RF is 88.86 percent and the accuracy of the ANN is 84.5 percent. It can be concluded that RF is having higher accuracy than the ANN machine learning model. ANN can be more accurate if it fed with more amount of data.

References

Abiodun, O. I., Jantan, A., Omolara, A. E., Dada, K. V., Mohamed, N. A. and Arshad, H. (2018). State-of-the-art in artificial neural network applications: A survey, *Heliyon* **4**(11): e00938.
URL: *https://www.sciencedirect.com/science/article/pii/S2405844018332067*

Alfarizi, M. I., Syafaah, L. and Lestandy, M. (2022). Emotional Text Classification Using TF-IDF (Term Frequency-Inverse Document Frequency) And LSTM (Long Short-Term Memory), *JUITA : Jurnal Informatika* **10**(2): 225.
URL: *https://jurnalnasional.ump.ac.id/index.php/JUITA/article/view/13262*

Ayvaz, S. and Shiha, M. (2017). The effects of emoji in sentiment analysis, *International Journal of Computer and Electrical Engineering* **9**: 360–369.

Hung, P. D., Hung, N. D. and Diep, V. T. (2022). Url classification using convolutional neural network for a new large dataset, *Cooperative Design, Visualization, and Engineering: 19th International Conference, CDVE 2022, Virtual Event, September 25–28, 2022, Proceedings*, Springer-Verlag, Berlin, Heidelberg, p. 103–114.
URL: *https://doi.org/10.1007/978-3-031-16538-2*$_1$1

Hönings, H., Knapp, D., Nguyn, B. C., Richter, D., Williams, K., Dorsch, I. and Fietkiewicz, K. J. (2022). Health information diffusion on Twitter: The content and design of WHO tweets matter, *Health Information & Libraries Journal* **39**(1): 22–35.
URL: *https://onlinelibrary.wiley.com/doi/10.1111/hir.12361*

Ikeda, D., Takamura, H., Ratinov, L.-A. and Okumura, M. (2008). Learning to shift the polarity of words for sentiment classification, *Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-I.*
URL: *https://aclanthology.org/I08-1039*

Karthika, P., Murugeswari, R. and Manoranjithem, R. (2019). Sentiment analysis of social media network using random forest algorithm, *2019 IEEE International Conference on Intelligent Techniques in Control, Optimization and Signal Processing (INCOS)*, pp. 1–5.

Kharde, V. and Sonawane, S. (2016). Sentiment analysis of twitter data: A survey of techniques, *International Journal of Computer Applications* **139**: 5–15.

Krouska, A., Troussas, C. and Virvou, M. (2016). The effect of preprocessing techniques on twitter sentiment analysis, *2016 7th International Conference on Information, Intelligence, Systems & Applications (IISA)* pp. 1–5.

Medhat, W., Hassan, A. and Korashy, H. (2014). Sentiment analysis algorithms and applications: A survey, *Ain Shams Engineering Journal* **5**(4): 1093–1113.
URL: *https://www.sciencedirect.com/science/article/pii/S2090447914000550*

Murthy, D., Allu, S., Andhavarapu, B. and Bagadi, M. (2020). Text based sentiment analysis using lstm, *International Journal of Engineering Research and* **V9**.

PERKINS, J. (2017). *NATURAL LANGUAGE PROCESSING: python and nltk*, PACKT Publishing, Place of publication not identified. OCLC: 1005111877.

Pozzi, F. A., Fersini, E., Messina, E. and Blanc, D. (2013). Enhance polarity classification on social media through sentiment-based feature expansion, *WOA@AI*IA*.

Prathyusha, K. S. and Reddy, B. E. (2021). Normalization Methods for Multiple Sources of Data, *2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS)*, IEEE, Madurai, India, pp. 1013–1019.
URL: *https://ieeexplore.ieee.org/document/9432142/*

Raschka, S. (2016). *Python machine learning: unlock deeper insights into machine learning with this vital guide to cutting-edge predictive analytics*, Community experience distilled, Packt Publishing open source, Birmingham Mumbai.

Sandagiri, S., Kumara, B. and Kuhaneswaran, B. (2020). Ann based crime detection and prediction using twitter posts and weather data, *2020 International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy (ICDABI)*, pp. 1–5.

Thakkar, A., Mungra, D., Agrawal, A. and Chaudhari, K. (2022). Improving the performance of sentiment analysis using enhanced preprocessing technique and artificial neural network, *IEEE Transactions on Affective Computing* **13**(4): 1771–1782.

Wang, S.-C. (2003). Artificial Neural Network, *Interdisciplinary Computing in Java Programming*, Springer US, Boston, MA, pp. 81–100.
URL: *https://doi.org/10.1007/978-1-4615-0377-4*$_5$