HEC MONTRÉAL

Use of Automatic Language Processing in a Legislative Context

By

Ye Xia

ye.xia@hec.ca

Thesis Supervisor

Gilles Caporossi

Gilles.caporossi@hec.ca

A Thesis

Presented to the Faculty of the Graduate School of HEC Montreal

In Partial Fulfillment of the Requirements for the Degree of

Master of Science (M. Sc.)

Department of Data Science and Business Analytics

September 2021

This thesis is submitted as part of my Master of Science program requirement at HEC Montreal.

This page is purposely left blank

Abstract

The online text contents on the governmental platform have made an important source for researchers to gauge government activity, determine their preference in sentiments or their opinions. Gaining insights into government behaviors enables us to better understand trends of authority, which in turn is helpful for making strategical business purposes such as investments, expansions.

The paper centers on how to effectively get insights from textual information of government portal: to investigate which are the most known and appropriate approaches for each sub-task, to implement the approaches in the experimented datasets, to analyze what is the weakness and strength of each of these approaches, to propose an improved method to tackle one of the weaknesses and to evaluate the experimented results of improved method.

This paper illustrates with the extraction of informative textual features for preparing the data, the use of robust NLP techniques such as tokenization for preparing the data, the implementation of supervised learning, deep learning, constituency parsing, unsupervised learning, and trend analysis for sentiment analysis, opinion detection, event detection, trend monitoring separately, the weakness of supervised learning due to noisy sources resulting in under performance of model and long computation time, the solution proposal of an improved method combining chunking by dependency parsing with supervised learning to address the weakness.

The results show that the dependency parsing helps to achieve high accuracies of models. It is possible to perform efficient signal extraction from online text data of authority based on improved method and use it to better know their trends of behavior.

Keywords: *Sentiment Analysis; Opinion Detection; Event Detection; Trend Monitoring; Constituency or Dependency Parsing; Machine Learning; Deep Learning; Trend Analysis, Canada Government or Parliament; Political Science.*

Table of Contents

List of abbreviations and acronyms

NLP: natural language processing

SVM: support vector machine

ML: machine learning

SA: sentiment analysis

VADER: Valence Aware Dictionary and sEntiment Reasoner

Glossary

Bags of Words	Bags of words are representations of text that describes the occurrence of words within a document. It is a method of feature extraction with text data for text modelling in natural language processing.
N-grams	N-grams are contiguous sequences of n items from a given sample of text or speech. For instance, “natural language processing” is a 3-grams.
Parts of Speech tags	Parts of speech tags are the properties of the words, which define their main context, functions, and usage in a sentence. Some of the commonly used parts of speech tags are Nouns, Verbs, Adjectives and Adverbs, etc.
Dependency-tree-based features	Dependency-tree-based features are triplets, phrases, or grammar dependency relationships, etc., extracted from directed graph representation also called dependency tree.
Naïve Bayes	Naïve Bayes is a probabilistic machine learning algorithm based on the Bayes Theorem, used in a wide variety of classification tasks.
SVM	Support-vector machine (SVM) is a supervised learning model with associated learning algorithms that analyze data for classification and regression analysis. SVM finds a hyperplane that creates a boundary between two classes of data to classify them.

Preface

This research is the result of a study of hard work and dedication. Several reasons have contributed to its start, development, and refinement in one way or another.

First and foremost, it has been written to fulfill the graduation requirements of master program of HEC Montreal. Second, it has been written to advance my knowledge in natural language processing. Third, it has been written to appreciate an opportunity to apply my knowledge into practical use. Lastly, it has been written to find insights and recommendations in the unexplored and open new doors for my professional career.

It requires no explanation that this work could not have been completed without the support of my supervisor Professor Gilles Caporossi to whom I would like to thank for his expertise and suggestion. His continuous and constructive guidance have shaped this dissertation for me.

Ye Xia

Acknowledgements

I would like to express my utmost gratitude to my thesis supervisor and academic advisor Professor Gilles Caporossi Department of Decision Sciences for providing me continuous support, guidance, and direction throughout this thesis research. He is one of the few Professors that have inspired me to go beyond and develop a further appreciation for natural language processing of my current study field.

I want to thank HEC Montreal for the opportunity to pursue a Master program in one of Canada's most prestigious research and academic institutions.

I am also grateful to my dear friend, Mr. Florian Carichon, a PHD student of Department of Decision Sciences for tremendous support, encouragement, and sharing his valuable suggestion throughout my academic and personal endeavors.

Lastly but not least, to my cousins and especially my incredible parents, the most important people in my life, I would like to show my appreciation for encouraging, sacrifice and supporting me you have given me through years. You are the only one constant in my life that drives me today and inspires me to be better every day and go beyond.

Thank you all for your unwavering support.

And to my readers, I would like to thank you for taking time to read my thesis. I hope this thesis would be an inspiration for you in your field of work.

I would like to end this chapter with a famous quote:

"Make your life a masterpiece; image no limitations on what you can be, have or do"

Chapter 1

Introduction and Problem Description

1.1 Introduction

It has been seen that data information has experienced exponential growth in modern world during these years. All types of machine automated systems are generating large amount of data in different forms like statistical, text, audio, video, sensor, and bio-metric data that emerges the term Big Data (Verma, Jai Prakash et al., 2016).

It is noticeable that text data is one of the important parts of big data because quite a few data information is available in textual form in databases. Big Text Data has significance in various perspective. Text data implies information about public opinions, product popularity, customer satisfaction, etc. This cannot be known if people do not have new technologies and architectures so that it becomes possible to extract value from it by capturing and analysis process (A. Katal et al., 2013).

The developed technology for big text data is called natural language processing (NLP). Natural Language Processing (NLP) is a theoretically motivated range of computational techniques for analyzing and representing naturally occurring texts at one or more levels of linguistic analysis for the purpose of achieving human-like language processing for a range of tasks or applications (Liddy, E.D., 2001). Natural language processing can handle many tasks. For example, information retrieval, Relation extraction, Named Entity Recognition (NER), Sentiment Analysis, Text classification or Document Classification, Part of Speech (POS), Keyword Extraction, etc.

But it seems that sentiment analysis, opinion detection, event detection and trend monitoring can turn high volumes of varied and rapidly changing data into meaningful insights when tremendous information from news, social media, blogs, and forums become available and attractive to explore in recent years (Fernández-Gavilanes,

Milagros, et al, 2016). Because sentiment analysis or opinion detection allow us to gain an overview of public opinion behind certain topics while event detection and trends monitoring helps to find events of public's interest and to see how they evolve. For instance, The Obama administration used sentiment analysis to gauge public opinion to policy announcements and campaign messages ahead of 2012 presidential election. So, being able to quickly see the sentiment behind everything from forum posts to news articles means being better able to strategize and to plan for the future (Brandwatch, 2021).

1.2 Problem Description

People are inundated with millions of information every day. It is an age of information bombardment, and it can foresee that the information can promote transformation in different parts of the world. Although the world lives with information, what the public can benefit from it and how they can convert it into effective way to utilize are the problems this paper should think about.

The public are always concerned about political or governmental information. News, policies, or regulations will all have an impact on people's lives. The residents keep up with the news trends and are exposed to information at the drop of a hat. However, they neither have the time to go all through nor have the scientific way to gather useful information to exploit.

The problem is quite evident that people need systematic approach to handle tremendous information from government or political science and find out the effective information relevant to them. If there is a way which can help to detect the threads of government with the collection of political news text, the public would benefit from acknowledging right away and making better decisions in their further events.

The paper can implement either sentiment analysis, or opinion detection, or event detection, or trend monitoring to provide insights for the public. However, things are not that simple - our problem has specialness:

It would be rather effortless to find customer opinion and foreknow event trend if the comments are explicitly expressed. For example, a product will be expected to suffer a serious defeat in sales when reviews for the product are negative. But news texts from government are implicitly presented. They could be factual statements, committee opinions, announcements of new regulations. It is not easy to identify government position which will influence on next decision. That means that to address this problem, it requires not only implementing various known approaches to discover the opinions, events or trends but also need more analysis to dive into those approaches (e.g., comparing the weakness and strength) in order to propose an appropriate methodology for getting meaningful insights from text information.

Because of the specialness of the problem, this paper will make two useful contributions to the knowledge in this area:

a) The paper implements several diversified but appropriate known approaches for four aspects: sentiment analysis, opinion detection, event detection and trend monitoring separately. The approaches are based on lexicon-based approach, supervised learning, unsupervised learning, deep learning, constituency parsing and dependency parsing, comparison analysis.

b) The paper introduces a methodology combing dependency parsing and supervised learning to help identifying opinions and the outputs after experiments are also attached.

And the document is organized as follows:

- The first part is an introduction presenting with the help of natural language processing such as sentiment analysis, opinion detection, event detection and trends monitoring, a great deal of text data information can be transformed to enlightening insights for the public. This part would also provide a description about the problem the paper needs to address and explain the specialness of this

problem.

- The second part is a literature review providing the information on what methodologies have been applied to tackle sentiment analysis, opinion detection, event detection and trends monitoring in current studies. And a description of the methods which is going to apply in this paper. It includes lexicon-based approach, machine learning, deep learning, constituency parsing, etc. This section will expatiate in each task on what are the methods, why are they, how to implement them and suggest possible reasons if when one method works better than the others.

- The third part is an overview of the datasets including source of data, nature of data, selection of data, collection of data, preparation of data and preprocessing of data.

- The fourth part demonstrates all the outputs from the implementation of applied approaches for tasks of sentiment analysis, opinion detection, event detection and trends monitoring. This part contains the results stemming from Vader and Textblob, supervised learning, feature clustering and cluster monitoring.

- The fifth part is an overall analysis of all the implemented methods. This section will talk about the weakness and strength of each of these approaches separately: automated sentimental classification tools for sentiment for sentiment analysis, supervised learning for opinion detection, unsupervised learning plus constituency parsing for event detection and clustering and trend analysis for trends monitoring.

- The sixth part is a proposition of the improved methodology and an explanation of what is the reason for the improved method and which weakness expected to tackle. Fundamentally, this section will introduce chunking by dependency parsing and how the dependency parsing will help and work together with supervised learning to detection opinions.

- The seventh part is a presentation of the experimental results. The section will consist of showing chunked text after using dependency parsing and demonstrating the performance of each model when using chunked text as input data. This section will also explain the possible reasons why one method works better than the others.

- The eighth part ends with conclusion providing key insights, limitation and possible extension can be tackled in the future research. The section will conclude on the impact of opinion detection in political science and other areas.

Chapter 2

Literature Review and Description of Methods

2.1 Introduction

Given this thesis is the study about getting effective information from selected data by trying methods. Sentiment analysis is a good way to find interesting insights for textual data. However, considering the specialty of our data, the paper will not only do the literature review on sentiment analysis, opinion detection, but also event detection and trend monitoring so that it can implement the most known approaches in the experiments.

These methods may either be supervised or unsupervised. Supervised learning is known by the use of labelled datasets to train algorithms that will classify the data or predict outcomes. In our case, this paper will use supervised learning for classification. For supervised learning to work, it needs a labeled set of data which can help the model to learn from and make correct decisions. In other words, raw datasets without labels needs to be labelled. Unsupervised learning is known to cluster and analyze unlabeled datasets (IBM, 2021). The algorithms specialize in discovering hidden patterns or data groupings without the need for human intervention (IBM, 2021).

This chapter will present an overview of the existing work in each of these aspects, and it is going to explain which method and for what reason for this paper is going to employ those methods for each part and explain how the method works.

2.2 Sentiment analysis

Sentiment analysis (SA) is drawing increasing attention in many fields. Marketing, finance, and the political and social sciences are all becoming the major applications areas to require sentiment information with the help of analytics tools (Amir Gandomi, and

Murtaza Haider, 2014). Feldman, Ronen (2013) described that sentiment analysis is defined as the task of finding the attitude of authors about specific entities. The decision-making process of people is affected by the comments formed by thought leaders and ordinary people. Therefore, the target of SA is to identify the sentiments they express and then classify their polarity (Walaa Medhat, et al (2014)). It expects to know the sentimental classification- positive, negative, or neural for the textual data input from a selected approach.

How to select an appropriate approach? In Walaa Medhat, et al (2014), sentiment classification techniques can be roughly divided into machine learning approach, lexicon-based approach, and hybrid approach.

The Machine Learning Approach (ML) applies ML algorithms and uses linguistic features. Machine learning based sentiment analysis or opinion mining typically includes the use of the formulation of a classification problem where the labels of the classifier refer to the sentiment expressed by a user on a particular topic. Features extracted may include bag-of-words or n-grams, Parts-of-Speech tags, dependency-tree-based features etc. Commonly used methods such as Naive Bayes, SVM, etc. are used as classification methods. The existing work applying machine learning techniques include V. Bobichev (2017) explored the task of sentiment analysis with Naïve Bayes, DMNBtext, NB Multinomial, SVM machine learning methods for Ukrainian and Russian news which is annotated by national technical university ‘KhPI’ students via an online interface. Abercrombie, Gavin, et al (2019) attempted to automatically do topic labelling with codes from a pre-existing coding scheme developed by political scientists for the annotation and analysis of political parties’ manifestos and further employed the models of Unigram overlap, cosine similarity, SVM, CNN and BERT to predict labels which would compare with labels from annotation process.

The Lexicon-based Approach relies on a sentiment lexicon, a collection of known and precompiled sentiment terms. Various publicly available lexicons are used for this purpose, each differing according to the context in which they were constructed. These

lexicons typically contain sentiment annotated words, usually in the form of a numeric score. Given some text, a "sentiment score" can be computed using the words occurring in the text and their respective scores in the lexicon concerned (Bhattacharjee, Kasturi, 2016). The existing work applying lexicon-based techniques comprises: another work was done by S. Taj (2019) who conducted sentiment analysis on BBC news articles by using a word stock dictionary with opinion words and match given set of words in a text for finding polarity (lexicon-based approach). Apoorv Agarwal, et al (2016) proposed to conduct opinion mining of news headlines using SentiWordNet. The instances of other techniques are Kim, Erin Hea-Jin, et al (2016) had an experiment of sentiment analysis on twitters and news articles with the query term 'Ebola' or 'Ebola virus' by applying the approach of n-gram Latent Dirichlet allocation (LDA) to identify topic trends and of SO-CAL developed by Taboada et al. (2011) to calculate topic-based sentiment score. David Vilares and Yulan He (2017) introduced Latent Argument Model (LAM) to identify a speaker's key arguments about certain topic in political debates. Examples include the Linguistic Inquiry and Word Count (LIWC) lexicon, the Multiple Perspective Question Answering (MPQA) lexicon, SentiStrength and SentiWordNet.

The hybrid Approach combines both approaches and is very common with sentiment lexicons playing a key role in the majority of methods.

This paper is going to select lexicon-based approach for sentiment analysis for two reasons:

a) The machine learning approach requires a training phase that is either conducted by the researchers themselves or by the sentiment software provider (Dhaoui, Chedia, et al. 2017).

b) For lexicon-based approach, numerous lexicon and rule-based sentiment analysis tools that is specifically attuned to sentiments expressed in social media have emerged and these packages are providing ready to use functionalities to perform

the analyzed process instead of building from scratch. For example, TextBlob and the Valence Aware Dictionary and sEntiment Reasoner (VADER).

What are TextBlob and VADER and what are their edges? TextBlob and VADER are two typical open source packaged models to determine sentimental tendency for textual data. In Hutto and Gilbert (2014), VADER is described as a popular word list- and rule- based procedure which computes a continuous score for each text (ranging from negative to neutral to positive values) and appends a total sentiment score called compound [1](Jacobs, Arthur M., 2019). A. Amin, I. Hossain, et al, (2019) mentioned that VADER has a gold-standard sentiment lexicon which is significant to categorize semantic orientation for sentiment analysis models. Reza Hermansyah et al, (2020) mentioned TextBlob is one of many python libraries for processing in the Natural Language Processing. Textblob can be used to perform a variety of NLP tasks ranging from parts of speech to sentiment analysis, and language translation to text classification. A big advantage using TextBlob is that it provides a very easy interface and offers a lot of features like phase extraction, pos-tagging, sentiment analysis, etc. And Rajesh Bose et al, (2020) carried out work on sentiment Analysis on Tweeter Comments with Textblob. In the results it is observed that TextBlob approaches produce fruitful outcomes. The outcomes establish a solid relationship between twitter comments and at the apex of or downgrade sentiment polarity and opinion.

How do VADER and Textblob work? For lexicon-based approaches, a sentiment is defined by its semantic orientation and the intensity of each word in the sentence. This requires a pre-defined dictionary classifying negative and positive words. Generally, a text message will be represented by bag of words. After assigning individual scores to all

[1] The compound score is a normalized weighted composite score ranging from−1/negative to 1/positive. According to the authors, a word or text segment has a positive sentiment if its compound score >= 0.05, a negative one if it <= 0.05. If the score falls in between these two threshold values, the text is considered neutral.

the words, final sentiment is calculated by some pooling operation like taking an average of all the sentiments (Shah, Parthvi, 2020).

Textblob's lexicon dictionary in format of xml is based on WordNet3 lexical database of the English language containing about 150,000 words organized in over 115,000 synsets for a total of 207,000 word-sense pairs (Kohli, Pahul Preet Singh, 2020). The dictionary contains 2919 words, and each word has a value which is a dictionary of part-of-speech tags. Tags include polarity: negative vs positive ranging from -1 to 1; subjectivity: objective vs subjective ranging from 0 to 1 and intensity: modifies next word ranging from 0 to 1 (TextBlob Sentiment,2021). For example, the word "great" has four records in the dictionary as shown in below Fig. 'Great'.

Fig. 'Great'.

Word	Polarity	Subjectivity	Intensity
great	1.0	1.0	1.0
great	1.0	1.0	1.0
great	0.4	0.2	1.0
great	0.8	0.8	1.0

In TextBlob Sentiment,2021, it also shows the process how Textblob returns polarity of an input. i) If the input is only for word 'great', the calculation is the simple average of all the polarities. That is 0.8= (1.0+1.0+0.4+0.8)/4. ii) When the input is 'not great', the total polarity will be -0.4 with the polarity score of 'great' multiply by negation which is equal to -0.5. iii)When the input goes to 'very great', the total polarity score is the polarity score of 'great' multiply by intensity of word of 'very' (polarity:0.2, subjectivity:0.3, intensity:1.3) because 'very' is a modified word for 'great'. The final score is 0.8*1.3=1.04 but is shown 1.0 because of the range. iv) The total score is changed when

the input is changed to ‘not a very great’. The polarity is equivalent to -0.5*1/1.3*0.8=0.31. And Textblob will neglect one-letter words such as ‘a’, then the total score is the same as ‘not very great’. v) The final point about Textblob is it will ignore words it does not show in the dictionary. In short, the process of Textblob to calculate the score for the input is to find words and phrases it can assign polarity to and simply averages them to get the final score for the input.

In Hutto, C.J. & Gilbert, E.E. (2014), VADER’s dictionary is based on existing well-established sentiment lexicons (LIWC, ANEW, GI) and supplemented with additional lexical features commonly used to express sentiment in social media text. The dictionary has over 7,500 lexical features with validated valence scores that indicated both the sentiment polarity (positive/negative), and the sentiment intensity on a scale from –4 to +4. For example, the word “okay” has a positive valence of 0.9, “good” is 1.9, and “great” is 3.1, whereas “horrible” is –2.5, the frowning emoticon “:(” is –2.2, and “sucks” and “sux” are both –1.5.

They also described how to calculate the valence of an input sentence (Hutto, C.J. & Gilbert, E.E. (2014)). Vader has its rule called ‘heuristic’ to handle punctuation, capitalization, degree modifiers, contrastive conjunctions and polarity negation. The compound score is computed by summing the valence scores of each word in the lexicon for each category, adjusted according to the rules, and then normalized to be between -1 (most extreme negative) and +1 (most extreme positive). The pos, neu, and neg scores are ratios for proportions of text that fall in each category (so these should all add up to be 1... or close to it with float operation). Figure below shows the formula for normalization.

$$CompoundScore = \frac{x}{\sqrt{x^2 + alpha}}$$

Where x=sum of valence scores of constituent words, and alpha = normalization constant (default value is 15)

For example: i) the sentence input is ‘Sentiment analysis has been good’. Checking in the lexicon, the paper found good has a positive valence of 1.9 while the rest has neural valence of 0. The compound score is 0.44= (1.9+0)/$\sqrt{(1.9+0)^2+15}$. The count of positive words and of neural words in the sentence are 1 and 4 separately, which will be used to calculate positive, neural, and negative score according to the formula setting in Vader. So, the pos score is 0.42= (1.9+1)/ (1.9+0+1+4), the neu score is 0.58= (0+4)/ (1.9+0+1+4) and the neg score is 0. ii) If the input is ‘Sentiment analysis has been good!’, the compound score will change to 0.4926= (1.9+0+0.292)/ $\sqrt{(1.9+0+0.292)^2+15}$ because the valence for the exclamation mark is 0.292. And the pos score is 0.444= (1.9+0.292+1)/ (1.9+0.292+0+1+4), the neu score is 0.556= (0+4)/ (1.9+0.292+0+1+4) and the neg score is 0.

2.3 Opinion detection

Sentiment Analysis help to identify the sentiment expressed in a text then analyses it (Walaa Medhat et al, 2014) while opinion detection is the recognition of documents that that reflect an opinion, whatever their polarities - positive or negative (Belbachir, Faiza, and Bénédicte Le Grand, 2015). The detection and interpretation of these subjective comments is strategic for various organizational and business purposes, e.g., product and service benchmarking, ads placement or market intelligence (Belbachir, Faiza, and Bénédicte Le Grand, 2015). This paper does expect to detection opinions among a great of text information from government by relying on the advanced technologies.

How to select an appropriate technology? Bhattacharjee, Kasturi (2016) mentioned that in the field of opinion detection in general are based either on machine learning, or lexicons of words.

The paper is going to select machine learning methods for opinion detection in this paper because a) lexicons of words typically contain sentiment annotated words rather than opinionated text. Various publicly available lexicons are used for certain purpose, each

differing according to the context in which they were constructed. They are usually performed on sentiment analysis instead of opinion detection. b) Machine learning techniques performed well for opinion detection. In Bhattacharjee, Kasturi (2016), he indicated that publicly available sentiment lexicons such as SentiStrength and SentiWordNet do not perform well for opinion detection on Twitter users. He also showed that prior sentiment analysis methods using Maximum Entropy, Naive Bayes, SVM, k-NN based strategies label propagation, etc. address the problem of temporal opinion detection.

Further, in the field of machine learning, it includes supervised learning and unsupervised learning. However, this paper has decided to use supervised learning to detect opinions for the reason: 1) Our goal is to identify whether the sentence is an opinionized statement. In other words, it is a binary classification problem. One of the differences of supervised learning and unsupervised learning is that supervised learning knows number of classes while unsupervised learning does not. 2) There are quite a few models in supervised learning are good at classification problem while unsupervised learning is designed for clustering. 3) In supervised learning, there is a way to compare models by evaluating the result. It means best model can found to detect opinions.

When supervised learning has been confirmed, it needs to solve four points: a. Ground truth labels. b. Feature engineering c. Model selection d. Measurement to evaluate the model accuracy.

a. Ground truth labels: The paper assigned an opinioned label and a non-opinioned label to each of datasets by manually reading all of contents. Label 1 for sentences if the paper finds they are related to the government opinions (e.g., writer's thinking and believe, by other words, arguments need to be verified) and label 0 for the rest.

b. Feature extraction: Feature extraction is a process that identifies important features or attributes of the data (Lilian Hobbs et al., 2019). Feature extraction

increases the accuracy of learned models by extracting features from the input data. This phase of the general framework reduces the dimensionality of data by removing the redundant data. Of course, it increases training and inference speed (Eman A.Abdel Maksoud et al., 2019). For Natural Language Processing (NLP), feature extraction from text corpora is an important step, especially for Machine Learning (ML) techniques (Eman A.Abdel Maksoud et al., 2019). And features must be extracted from the textual data this paper deals within many NLP tasks, as ML algorithms cannot process text directly. The challenge is to select features that are informative and discriminative (Broda, Bartosz, et al., 2013). There are so many features extraction techniques such as Bags of Words, TF-IDF, word embedding, NLP based on features like word count, none count, etc. (Lilian Hobbs et al., 2019). They can help to convert text sentences into numeric vectors. Important feature vectors remain the most common and suitable signal representation for the classification problems (Lilian Hobbs et al., 2019). The paper has decided to use different ways to extract feature vectors so that to find a good one by comparing them in a model. The ways include Bags of Words, TF-IDF, word embedding, NLP based on features.

- Bags of Words: Bag of Words refers to the fact that this model does not take the order of the words into account. Instead, one can imagine that every word is put into a bag, where the ordering of the words gets lost (Eklund, Martin, 2018). Bag-of-Words is commonly used in clustering, classification, and topic modeling by weighing special words and relevant terminologies (Eiki, 2019). The most common one is to simply count the number of occurrences of each word within a document and keep the result in a vector (referred to as a count vector) (Mehdi Allahyari et al., 2017), which will be the method of Bags of Words in this paper.

- TF-IDF: Term Frequency-Inverse Document Frequency (TF-IDF) is an information retrieval technique that can be used to determine the relevance of terms in documents in relation to a query (Zach CHASE et al.,2014). In this case it can be used for feature extraction by determining which terms in a document are most distinguishing for that document. It was proven to be both simple and effective for feature extraction (Eklund, Martin, 2018). TF-IDF usually performs better in machine learning models (PURVA HUILGOL, 2020) and 83% of text-based recommender systems uses TF-IDF (BEEL, Joeran et al., 2016). TF-IDF is a "calculation method" to score an importance of words in a document. How to calculate TF-IDF? TF-IDF consists of two steps, first calculating the term frequency (TF), and then calculating the inverse document frequency (IDF). The final step is to multiply TF part and the IDF part for a certain term, from which the paper gets a measure of how distinguishing that term is (Eklund, Martin, 2018).

  To understand Term Frequent (TF), formula is used as below:

$$tf_{t,d} = \frac{n_{t,d}}{\sum_k n_{k,d}}$$

Where $n_{t,d}$ is the number of times that term t occurs in document d, and $n_{k,d}$ is the number of occurrences of every term in document d.

  To understand inverse document frequency (IDF), formula is as below:

$$idf_t = log\frac{|D|}{|D_t|}$$

Where $|D|$ is the total number of documents, and $|D_t|$ is the number of documents where the term t appears.

Also, TF-IDF Vectors can be generated at different levels of input tokens (words, characters, n-grams) (Uddin, Moahammad Nasir, 2021) :

i. Word Level TF-IDF: Matrix representing tf-idf scores of every term in different documents

ii. N-gram Level TF-IDF: N-grams are the combination of N terms together. This Matrix representing tf-idf scores of N-grams

iii. Character Level TF-IDF: Matrix representing tf-idf scores of character level n-grams in the corpus

- Word Embedding: Words are represented in a real valued vector space when using word embedding models (Tobias Schnabel et al., 2015). A good word embedding would ideally represent words in such a way that two different words with similar semantic meanings would have similar vector space representations (Eklund, Martin, 2018). There are several popular implementations such as word2vec, GloVe, FastText etc. This paper will use FastText pre-trained word vectiors: wiki-news-300d-1M.vec which has 1-million-word vectors trained on Wikipedia 2017, UMBC webbase corpus and statmt.org news dataset (16B tokens) to do word embedding. FastText is an extension of the word2vec model. Instead of learning vectors for words directly, FastText represents each word as an n-gram of characters. So, Word2vec and GloVe both fail to provide any vector representation for words for rare and out of dictionary words. This is another advantage of FastText (SANJANA REDDY,2019).

c. Model selection: Supervised learning has various types of classifiers with respective benefits and drawbacks. The paper is going to employ Multinomial Naive Bayes, Logistic Regression, Support Vector Machine, Random Forest,

XGboost, Neural Network, Convolutional Neural Networks， Recurrent Neural Network (RNN)- Long short-term memory (LSTM)， Recurrent Neural Network (RNN)- Gate Recurrent Unit (GRU), RNN-Bidirectional, CNN-Bidirectional with feature vectors as inputs because they are good classifiers but need to have a classifier with best performance by comparing their accuracy score.

- Multinomial Naive Bayes: Multinomial Naive Bayes algorithm is a probabilistic learning method that is mostly used in Natural Language Processing (NLP). The algorithm is based on the Bayes theorem and predicts the tag of a text such as a piece of email or newspaper article. It calculates the probability of each tag for a given sample and then gives the tag with the highest probability as output (Shriram, 2021). While Naive Bayes classifier is a general term which refers to conditional independence of each of the features in the model, Multinomial Naive Bayes classifier is a specific instance of a Naive Bayes classifier which uses a multinomial distribution for each of the features. And the multinomial distribution describes the probability of observing counts among a number of categories, and thus multinomial naive Bayes is most appropriate for features that represent counts or count rates (Jake VanderPlas, 2016).

- Logistic Regression: Logistic regression is an advanced Linear regression technique used for classifying both linear and non-linear data (O. Aborisade and M. Anwar, 2018). Logistic Regression works by taking input and multiplied the input value with weight value. It is a classifier that learns what features from the input that are the most useful to discriminate between the different possible classes (S. T. Indra et al., 2016). It is commonly used to model data with binary responses (O. Aborisade and M. Anwar, 2018) and it is a powerful way of modelling binomial outcome as well (Shah, Kanish, et al, 2020).

- Support Vector Machine: SVMs were developed by V. Vapnik et al. based on the structural risk minimization principle from statistical learning theory (Joachims, Thorsten,2001). Support Vector Machine (SVM) is a supervised machine learning algorithm which can be used for both classification and regression challenges. The idea behind SVMs is to find a linear hyperplane (or decision boundary) which separates the data-points of one class from the rest, and to do this in such a way that maximizes the margin between them (Shah, Kanish, et al, 2020). Support Vector Machines (SVMs) have built-in overfitting protection, which is one of the reasons why they have been proven to work well for text categorization (Joachims, Thorsten,2001).

- Random Forest: Random Forest is a popular classification method which is an ensemble of a set of classification trees. Due to its algorithmic simplicity and prominent classification performance for high dimensional data, random forest has become a promising method for text categorization (Baoxun Xu et al., 2012). A Random Forest model comprises a set of decision trees each of which is trained using random subsets of features. Given an instance, the prediction by the RF is obtained via majority voting of the predictions of all the trees in the forest. (Islam, Md Zahidul, et al., 2019).

- XGboost: eXtreme Gradient Boosting (XGBoost) is a scalable machine learning system for tree boosting that is widely used by data scientists and provides state-of-the-art results on many problems. XGBoost supports various weighted classification and rank objective functions, as well as user defined objective function. And it is able to solve real world scale problems using a minimal number of resources (Chen, Tianqi, 2016). The basic principle behind XGboost is to combine multiple Decision trees with lower accuracy into a model with higher accuracy. The XGBoost algorithm adopts the idea of gradient descent in the generation of each tree. Based on the tree generated in the previous step, it iterates to the direction of the minimum given objective

function. Through the iteration of multiple decision trees, the loss error is continuously reduced, and the prediction model is finally obtained. The split nodes of each decision tree are constructed in accordance with the criteria of CART (regression) tree, and the least square loss and logarithmic function are commonly used (Z. Qi, 2020).

- Neural Network: Neural network is a popular classification method, it can handle linear and nonlinear problems for text categorization, and both of linear and nonlinear classifier can achieve good results. Neural networks have been widely applied by many researchers to classify the text documents with different types of feature vectors. The neural network is trained with back-propagation algorithm. It is consistently repeated the weight learning process to classify and predict a class of samples. The inputs are the components of the document vector, and the outputs are the document categories (F. Harrag and E. El-Qawasmah, 2009). In our experiment, there have one input layer, one hidden layer with active function “relu”, and output layer with active function “sigmoid”. For the compile setting, the optimizer is Adam and loss function is binary cross entropy.

- Convolutional Neural Networks: Convolutional Neural Networks, known as CNN, is a category of Neural Networks that uses a multilayer perceptron variation that is designed for minimal preprocessing (N I Widiastuti, 2019). In Jacovi, Alon, et al., 2020, they mentioned that convolutional Neural Networks (CNNs), originally invented for computer vision, have been shown to achieve strong performance on text classification tasks (Bai et al., 2018; Kalchbrenner et al., 2014; Wang et al., 2015; Zhang et al., 2015; Johnson and Zhang, 2015; Iyyer et al., 2015) as well as other traditional Natural Language Processing (NLP) tasks (Collobert et al., 2011), even when considering relatively simple one-layer models (Kim, 2014). Convolution Neural Network (ConvNets) involves a series of filters of different sizes and shapes which convolve (roll

over) the original sentence matrix to reduce it into further low dimension matrices. In text classification ConvNets are being applied to distributed and discrete word embedding. The down sampling technique used in convolutional neural network is L2 Regularization. CNN utilizes an activation function which helps it run in kernel (i.e) high dimensional space for neural processing (Zain Amin et al., 2018). For the architecture in this paper, it has one input layer, two embedding layers (one layer for embedding, the other for spatial dropout1-D), one convolutional layer with 'relu', one pooling layer for GlobalMaxPool1D, two output layers (one layer for dense with 'relu' and dropout, the other is final output for dense with 'sigmoid'). For the compile setting, the optimizer is Adam and loss function is binary cross entropy.

- Recurrent Neural Network (RNN)- Long short-term memory (LSTM): A recurrent neural network (RNN) is a class of artificial neural networks where connections between nodes form a directed graph along a temporal sequence. Recurrent neural network (RNN) is one of two main architectural types of deep neural networks besides CNN, and they have been widely used in natural language processing projects. RNN is good at modeling units in sequence while CNN can extract features that do not change position. RNN scans the data input mode so that all data parameters of each time step are shared. Each time step will not only receive the input of the current time, but also receive the output of the previous time, thereby successfully using the input and input of the past information to assist in the judgment of the current moment. For text classification task, RNN represented all sentences to a vector by pre-training, and then, RNN encode all sentences vector and predict the class of sentence. Although this model can perform text classification well, gradient vanishing and exploding problem often occur when training RNN. To the gradient problem, Long Short-Term Memory (LSTM) and Gated Cyclic Unit (GRU) were proposed (H. Hu et al, 2020). LSTM is an artificial recurrent neural network (RNN) architecture capable of learning capable of learning

capture the long-term and short-term dependencies in a sequence. LSTM maintains a separate storage unit internally and only updates and discloses its contents when deemed necessary by forget gate, the input gate and the output gate (H. Hu et al, 2020). Models LSTM has shown outstanding results in many domains such as language modeling, tagging problem, and sequence-to sequence predictions (Sari, Winda Kurnia, et al., 2020). In our experiment, RNN-LSTM model includes one input layer, two embedding layers (one layer for embedding, the other for spatial dropout1-D), one LSTM layer, two output layers (one layer for dense with 'relu' and dropout, the other is final output for dense with 'sigmoid'). For the compile setting, the optimizer is Adam and loss function is binary cross entropy.

- Recurrent Neural Network (RNN)- Gate Recurrent Unit (GRU): To address the gradient problem occurred in RNN, Long Short-Term Memory (LSTM) and Gated Cyclic Unit (GRU) were proposed (H. Hu et al, 2020). LSTM was proposed by Hochreiter et al. in 2019 and consists of an input gate, a forgotten gate, an output gate, and a memory cell. GRU was proposed by chung et al. in 2014, which combines the forgotten gate and the input gate in LSTM into an update gate, and without a separate memory cell. Compared with LSTM, the GRU structure is simpler, and the convergence time required for the same task is shorter, but the training effect is similar to LSTM (Luo, Li-xia., 2019). Models GRU are effective in the task of text classification because of their capability to remember long time dependencies and efficiently capture the semantics between words. GRU approaches are especially useful for sequential datasets (Zulqarnain, M., et al., 2019). In this paper, RNN-GRU architecture includes one input layer, two embedding layers (one layer for embedding, the other for spatial dropout1-D), one GRU layer, two output layers (one layer for dense with 'relu' and dropout, the other is final output for dense with 'sigmoid'). For the compile setting, the optimizer is Adam and loss function is binary cross entropy.

- RNN-Bidirectional: A bidirectional recurrent neural network (BRNN) is an extension of a regular recurrent neural network (RNN). Bidirectional Recurrent Neural Network (BiRNN) connects two hidden layers in opposite directions to the same output. The output layer can get information from past (backwards) and future (forward) states simultaneously. BiRNN increases the amount of input information available to the network (Gangwar, Akhilesh Kumar, and Vadlamani Ravi, 2020). Bidirectional GRU consists of forward and backward GRU, which considers the contextual information on text sequence (Lu, Guangquan, et al., 2020). The Bi-directional GRU resolves the feed-forward model's limited ability by extracting limitless contextual information from the front and back. Then, the hidden state values for both aspects and context are averaged separately, giving an initial representation of both, which will be used later to calculate the vectors of attention (Abdelgwad, Mohammed M., et al, 2021). BiGRU is the most advanced RNN and is less complex compared to BiLSTM. BiGRU works as a better window-based feature extractor (Kumar, and Vadlamani Ravi, 2020). For RNN-Bidirectional GRU architecture in this paper, it includes one input layer, two embedding layers (one layer for embedding, the other for spatial dropout1-D), one Bidirectional GRU layer, two output layers (one layer for dense with 'relu' and dropout, the other is final output for dense with 'sigmoid'). For the compile setting, the optimizer is Adam and loss function is binary cross entropy.

- RCNN-Bidirectional: Recurrent Conventional neural network with bidirectional GRU is combined two main network architectures: convolutional neural networks and recursive/recurrent neural networks, and along with Bidirectional GRU based recurrent neural network (RNN). Convolutional neural network aims to generalize the local and consecutive context of the relation mentions, while recurrent neural networks adaptively accumulate the context information in the whole sentence via memory units, thereby encoding

the global and possibly inconsecutive patterns for relation classification (Xie P et al., 2018). Bidirectional GRU is responsible for long term dependency, vanishing gradient and exploding gradient. RCNN-Bidirectional GRU combines with multiple attention mechanisms made the network focus on and learn different focuses in the training process through different attention mechanisms which is expected to improve classification accuracy and faster convergence speed (Zhang, Jingren, et al, 2019). In this paper, the architecture includes one input layer, two embedding layers (one layer for embedding, the other for spatial dropout1-D), one Bidirectional GRU recurrent layer, one convolutional layer, one pooling layer and two output layers (one layer for dense with 'relu' and dropout, the other is final output for dense with 'sigmoid'). For the compile setting, the optimizer is Adam and loss function is binary cross entropy.

d. Measurement to evaluate the model accuracy: It needs a metrics to tell us each model's performance (whether it is good or bad) by measuring its accuracy. Scikit-learn contains many built-in functions for analyzing the performance of models. One of the metrics is accuracy_score. The accuracy_score in sklearn is taking as inputs the actual labels and the predicted labels. It is the measure of all the correctly identified cases. To be more detailed, it is the fraction of samples predicted correctly as shown below:

$$\text{Accuracy} = \frac{\text{True Positive + True Negative}}{\text{(True Positive + False Positive + True Negative + False Negative)}}$$

The reason why choose accuracy_score as measurement because it is one of the more obvious metrics. Our problem is with all the classes equally important, so accuracy_score is usually a good start. And there is another additional benefit of this measurement: it would be easy to understand for non-technical stakeholders.

2.4 Event Detection

Opinion detection is detecting user opinions on a topic in which the conversation evolves over time (Bhattacharjee, Kasturi et al, 2016) while event detection is the process of analyzing event streams in order to discover sets of events matching patterns of events in an event context (Mellin et al., 2009). Online new event detection and tracking which is part of topic detection and tracking (TDT) was first studied by Allan et al. in 1998. Mining online news for events was a hot topic in information retrieval during the last decade (Unankard, Sayan, 2015).

What are the existing works the researchers have? They include F. Jiang et al (2009) proposed unsupervised clustering of object trajectories to detect unusual video events. In 2015, Pohl, D. et al proposed clustering approaches for sub-event detection in the context of social media crisis management. Zeinab Ghaemi and Mahdi Farnaghi (2019) applied spatial clustering on geotagged tweets providing the ability to discover events and their locations. Moreover, Ena, O. et al (2016) introduced a systematic technology trend monitoring (TTM) methodology based on an analysis of bibliometric data. TTM process is composed of formation of a list of terms, data scanning in databases and collections, data clusterization, identification and description of trends, creation of a trends database. In 2020, Leonid Gokhberg et al suggested an advanced text-mining for trend analysis of Russia's extractive industries. This methodology consisted of four main stages: primary natural language processing, syntactic-semantic analysis, topic modeling, classification and clustering to link dominant discussions (e.g., climate change vs rural development) and to flag key trends.

Other works are composed of - Lam, W., et al (2001) proposed contextual analysis to detect topically related stories from a continuous stream of news. Linmei Hu et al (2017) presented an adaptive online event detection for online news event detection. And L.Dey et al (2009) proposed a stock market analysis system that analyzes financial news items to identify and characterize major events that impact the market by using Latent Dirichlet

Allocation (LDA) based topic extraction mechanism for the events identification and kernel k means algorithm for the topic-document data clustering.

In this paper, it can be expected to detect the occurrences of events and categorize them for event detection. How to select an appropriate technology? Mellin et al., (2009) mentioned that the event patterns and the event contexts define event types. If a set of events matching the pattern of an event type is discovered during the analysis, then subscribers of the event type should be signaled. The analysis typically entails filtering and aggregation of events.

The paper selects feature clustering for event detection because a) Feature clustering has evolved to be a powerful method for clustering text documents. Feature clustering is one such algorithm that allows documents with pair wise semantic relatedness to be grouped together. Each document will be identified by a minimal number of features or words; hence the overall dimensionality could be reduced by drastic amounts (Dev, Divya D., and Merlin Jebaruby, 2014). b) This method works to find the real distribution of words in the text documents. Experimental results do show that the proposed method is much better when compared against several other clustering methods. The distinguished clusters are identified by a unique group of top keywords, obtained from the documents (Dev, Divya D., and Merlin Jebaruby, 2014).

Specially, this paper selects K-means for clustering and Constituency Parsing for feature extraction since 1) K-means scales to large datasets, guarantees converge, easily adapts to new examples and generalizes to clusters of different shapes and sizes 2) Constituency Parsing is an attractive technique for use in information extraction because it can provide a representation: the sub phrases based on grammatical structure that is convenient to use in any further layers of analysis.

How does feature clustering work? It will collect the content summary of each article for the first step and implement dependency parsing for the second step to extract the name words from article summary and employ the K-means to do clustering as the final step.

a) Collecting summaries of news: It can be found a brief under article header in each news of our dataset. The brief describes what is the news about including when, where, who, what, why or how. This paper collected summaries of 11 articles to do feature clustering experiment.

b) Constituency Parsing for noun phrases: constituency parsing based on the formalism of context-free grammars. The sentence is divided into constituents, that is, sub-phrases that belong to a specific category in the grammar such as verb phrases or noun phrases. Prior research by Shakira et al. (2014) has shown that by monitoring specific known keywords in Twitter, it is possible to perform event detection. These event-monitoring systems usually track a fixed set of manually chosen NPs (Chua, Freddy Chong Tat, et al, 2012). SpaCy has a good parser which can power the sentence boundary detection, and lets you iterate over base noun phrases, or "chunks". (SpaCy, 2020). The paper is going to use SpaCy to chunk a noun plus the words describing the noun.

c) K-means: Clustering is an unsupervised learning technique which has no need for labels. It is the process of grouping data samples together into clusters based on a certain feature that they share. Text clustering is the application of cluster analysis to text-based documents. K-means is a clustering algorithm, which can take data points input and groups them into K clusters. It is easy to implement and is able to identify unknown groups of data from complex data set. The k-Means algorithm works as a hard cluster algorithm, and each data point is deterministically assigned to a specific cluster. The main idea is to define k centroids, one for each cluster. In order to achieve this, the algorithms rearrange an initial non-overlapping cluster setting into k optimal clusters by moving the data objects from one cluster to another such that a certain homogeneity criterion is optimized (Beumer, Lisa, 2020).

2.5 Trend Monitoring

Event detection is to capture content associated with various types of events including both planned (e.g., government projects or committee activities) and unplanned (e.g., natural disaster and epidemics) events (Shakira BanuKaleel, 2015), while trend monitoring is to trace the information about the behavioral changes of events over the time. For instance, trends can illustrate the events popularity over time in media platform (Shakira BanuKaleel, 2015).

Trend monitoring has been widely applied in many contexts. When conducted by experts, trend monitoring will increase your organization's situational awareness. This reduces uncertainty, giving you the ability to anticipate both your competition and market forces and adapt your goals accordingly (Cipher, 2021). It is expected to know how the event trend is changing overtime by trend monitoring.

How to select an appropriate technology? Poppe, Olga, et al. (2019) mentioned Streaming applications from cluster monitoring to detect and aggregate event trends. The paper selects clustering monitoring for event trends tracing in this paper because a) The task of event detection has been focused on in chapter 4.3. Therefore, it requires finding an approach to monitor these detected events so as to conduct real-time analysis and management. b) Regarding a trend monitoring approach for event, it should combine both event detection and event trending tasks. It relies on clustering technique to detect events from massive collection of unstructured data which contains different types of events. The continued work should leverage the availability of discovered events and existing technique for comparing the event situations during different periods.

How does cluster monitoring work? The monitoring work is comparing the old cluster with new cluster which comes from old events and new events to find the trend change at different time. For example, there have a clustering output from previous collected 11 sentences and 5 new events happened when the time goes by. A new clustering process will be proceeded because of the occurrence of new events. As a result, the paper will

have a new output from clustering of 16 events. By comparison (new vs old), it can monitor how the clusters change over time.

Chapter 3

Description of the Data

After literature review, an introduction of the data will be provided in this chapter to indicate what kind of data will be used and how the data will be managed in this paper.

3.1 Source of data

People do care about news, policies, or regulations from government because government actions have influence in every way of people's lives and eager to obtain insightful ideas from masses of information. Because of this, this paper is going to conduct the research based on government portal.

Governmental data sources could be many and varied. The paper selects the source of data coming specifically from Parliament[2] website since a) This is an official website that that examines what the government is doing, makes new laws, holds the power to set taxes and debates the issues of the day. b) This is a website provided and employed by a consulting company dedicating in offering professional service in political science. This means this website is not only authoritative but also receives a wide range of audience. c) This website is comprehensive. It has a great number of contents ranging from house of Commons, house of Lords, Bills and Legislation to News from Parliament.

3.2 Nature of data and Selection of data

3.2.1 Nature of data

The Parliament website is publishing order papers, votes and proceedings, oral questions

[2] Available at https://www.parliament.uk/

Rota, business papers from both Houses, draft bills, and featured news, etc. All the data information from the website are recorded in textual form and updated every single day. It is convenient to browser all the published information on the web page. Most of sections need to be viewed in HTML with some of parts can be also viewed in PDF and some are possible to download in text format. However, all the text data information is pure records which cannot be directly used as model data.

3.2.2 Selection of data

Although there have rich data information on the website, featured news will be the input data in this research paper attributing to a) The public are paying more attention to news than other resources all the time. It would provide great help if the technology could transform plain texts into insights. b) The subjects of featured news have diversity ranging from agriculture, crime, culture, education to economy and so on. People can benefit from insights from different areas if the datasets are informative. c) News is updated faster than other resources. It is possible to proceed prompt analysis and generate real-time insights based on quick update of news. The paper will attach an example to explain what the featured news looks like.

Title: Industrial Strategy Challenge Fund needs results-focussed 'overhaul' to deliver for UK economy and taxpayer

The Industrial Strategy Challenge Fund (ISCF) was set up to help "address some of the complex issues" the UK economy faces, including long-term low productivity and living standards. Managed by UK Research and Innovation (UKRI) under the Business department (BEIS), it is designed around four 'grand challenges': future mobility; clean growth; artificial intelligence and data; and the ageing society.

The ISCF is a key element in achieving the government's ambitious target for the UK to spend 2.4% of GDP on R&D by 2027, but "this was challenging before the outbreak of COVID-19 and is more so now". BEIS has not yet made clear how it will meet the target and is "insufficiently focused on what it is expected to deliver in terms of benefit to the UK".

The Committee has “concerns about the Fund’s clarity of purpose including the multiple projects now being funded”. By January 2021 over 1,600 projects had benefited from funding of £1.2 billion to support innovation in some of the most complex issues faced by the UK. Businesses and other bodies have contributed almost £600 million in “co-investment”, but the Committee says the financial support “is currently concentrated in certain parts of the country and larger organisations have recently received an increasing proportion of funding”. The Committee is concerned this risks undermining future performance by overlooking ideas from elsewhere and smaller businesses.

Although UKRI can point to good performance in beginning to tackle the various chosen challenges and in involving industry in the selection of challenges to support, its objectives for the Fund overall are input focused. Government must “track the number of jobs delivered over time against job creation ambitions if it is to properly demonstrate its economic impact”.

Structural issues in the Fund’s design - such as a lengthy approvals processes and the industry ‘co-investment’ requirements - similarly “need an overhaul if it is to play an important role in helping re-build the UK economy post-pandemic”.

Meg Hillier MP, Chair of the Public Accounts Committee, said:

“The message on the ISCF is a recurring one for too many programmes across Government - they are too focussed on inputs, on ticking boxes and distributing funds, rather than on outcomes. Throwing more taxpayers’ money at the UK’s notorious, long term productivity and opportunity problems, yet again without a clear, integrated plan or measures of proof that it’s working, reinforces the strong and unfortunate impression of ‘government by announcement’. Show us the government by results.”

3.3 Collection of data

Data collection is the process of gathering raw text data information from data source. Featured news is published on the web page and cannot be downloaded in text format. It is necessary to rely on web crawling to collect data. Web crawling also known as web scraping has been used to extract web data in many fields while web crawling tool allows to accomplish automation process for web data extraction.

The paper selects a tool called “Scraper” to do web crawling among numerous web-

crawlers.

Scraper is a Chrome extension with limited data extraction features, but it is helpful for making online research. It also allows exporting the data to Google Spreadsheets. This tool is intended for beginners and experts. You can easily copy the data to the clipboard or store it to the spreadsheets using OAuth. Scraper can auto-generates XPaths for defining URLs to crawl. (Octoparse, 2021). The Scraper has pros for our data collection task: 1) free of Cost & Good Documentation and Video Tutorials available. 2) ideal for simple and light data extraction jobs. 3) it can extract data from multiple pages.

How to use collect data by using web scraper? It has four steps: first step is to install the extension of web scraper, second step is to visit the source website, third step is to create sitemap. Sitemap is plan or traversal map to collect data from the website in web scraper. The next step is to create selectors. Selectors as its name suggests, are html elements that contain data and links to navigate. The last step is to start scraping the website by scrape option in Sitemap and export to csv file (Pro Web Scraping, 2016).

3.4 Preparation of data and preprocessing of data

3.4.1 Preparation of data

The raw data contains features that is waiting to be extracted for our models. It is critical to extract useful features because machine learning model learns the feature and label values that given by us and predicting the value of previously unseen, new feature value's corresponding label value (Tekyaygil, Fethi, 2020). The paper will include Bags of Words, TF-IDF and word embedding in terms of feature extraction and has introduced them with more details in 2.3.b for preparing the data.

3.4.2 Preprocessing of data

Raw data and features cannot be sent through a model if they are not transformed into an understandable format. It requires data split, label-encoding, tokenization, sequence padding and embedding matrix building.

- Data split: Data splitting is the act of partitioning available data into two portions;usually for cross-validatory purposes. One portion of the data is used to develop a predictive model and the other to evaluate the model's performance (Picard, Richard R., and Kenneth N. Berk, 1990). The purpose of splitting data is to avoid overfitting which is paying attention to minor details noise which are not necessary and only optimizes the training dataset accuracy (Farsan Rashid, 2018). The paper will use train and test split function of model selection method in Sklearn to split the data.

- Label-Encoding: Label Encoding is a popular encoding technique for handling categorical variables. In this technique, each label is assigned a unique integer based on alphabetical ordering (ALAKH SETHI, 2020). This is done because our machine learning model doesn't understand string characters and therefore there should be a provision to encode them in a machine-understandable format (Banerjee, Sagnik, 2020). In this paper, label-encoding is processed by LabelEncoder of preprocessing in Sklearn library and fit transformation function to return encoded labels of data input.

- Tokenization: Tokenization is essentially splitting a phrase, sentence, paragraph, or an entire text document into smaller units, such as individual words or terms. Each of these smaller units are called tokens (SHUBHAM SINGH, 2019). The process of tokenization could help to understand the meaning of the text that can easily be interpreted by analyzing the words present in the text. Once there have a list of words, this paper can also use statistical tools and methods to get more insights into the text. For example, the paper can use word count and word frequency to find out important of word in that sentence or document (Satish Gunjal, 2021). For the experiment, the paper will use tensorflow keras.preprocessing.text.Tokenizer to do text tokenization.

- Sequence Padding: Padding is a special form of masking where the masked steps are at the start or the end of a sequence. Sequence Padding is a process to encode sequence data into contiguous batches: in order to make all sequences in a batch fit a given standard length, it is necessary to pad or truncate some sequences. It needs padding for sequence because all the neural networks require to have inputs that have the same shape and size. It also allows to design deeper networks and improve performance by keeping information at the boarders. In our paper, it will use keras.preprocessing.sequence.pad_sequences to pad the sequence data.

- Embedding Matrix Building: An embedding matrix is a list of all words and their corresponding embeddings. It is consisted of a matrix of embedding vectors. The concept of an embedding matrix is an attempt to solve relationship representation problem. With embedding matrix, it could keep the size of each vector much smaller so that the memory is highly reduced and the computationally will be efficient. In this paper, it will use python Numpy to build embedding matrix.

Chapter 4

Implementation of Known Approaches

The paper has presented the known approaches in the Chapter 2 and have introduced the data in Chapter 3. In this chapter, the paper will show the outputs from implementation of these approaches.

4.1 Vader and Textblob outputs for Sentiment analysis

The points that what are the VADER and TextBlob, why they are chosen, and how they work have been explained in Chapter 2. Here are the results coming from tools of Vader and Textblob.

Fig. Vader Result

	Text	Scores	Compound	comp_score
0	Claims that Covid-19 testing gives too many false positive results have led some to believe they shouldn't be relied on to shape responses to the pandemic.	{'neg': 0.0, 'neu': 0.874, 'pos':0.126, 'compound': 0.5574}	0.5574	pos
1	The argument stems from a genuine issue with medical tests about the risk of false positive results.	{'neg': 0.217, 'neu': 0.613, 'pos': 0.17, 'compound': 0.0}	0	neu
2	It's certainly true that some tests which show as positive may be false positives.	{'neg': 0.0, 'neu': 0.45, 'pos': 0.55, 'compound': 0.9042}	0.9042	pos
3	It's equally true that some of the negative results will be false negatives.	{'neg': 0.211, 'neu': 0.629, 'pos': 0.16,	-0.2263	neg

		'compound': -0.2263}		
4	This Insight examines what can affect test results and explain why we can't always apply general mathematic principles to specific situations (like Covid-19)	{'neg': 0.0, 'neu': 1.0, 'pos': 0.0, 'compound': 0.0}	0	neu
5	For a test to pick up a high proportion of genuine cases, it depends on a combination of the sensitivity and specificity of the test and the population being tested.	{'neg': 0.0, 'neu': 1.0, 'pos': 0.0, 'compound': 0.0}	0	neu
6	Sensitivity looks at how well a test can identify true positive cases.	{'neg': 0.0, 'neu': 0.485, 'pos': 0.515, 'compound': 0.8176}	0.8176	pos
7	A test that is very sensitive will flag up almost everyone who has a disease and won't give you many false negative results	{'neg': 0.156, 'neu': 0.844, 'pos': 0.0, 'compound': -0.5719}	-0.5719	neg
8	Specificity looks at a test's ability to correctly give a negative result for people who really don't have the disease.	{'neg': 0.168, 'neu': 0.727, 'pos': 0.105, 'compound': -0.34}	-0.34	neg
9	So, a Covid-19 test with 90% specificity will correctly tell 90 people in every hundred that they don't have Covid-19 but will tell 10 others that they have Covid-19 when they don't	{'neg': 0.0, 'neu': 1.0, 'pos': 0.0, 'compound': 0.0}	0	neu
10	Although we can't give an exact figure for the number of false positives, there's no evidence to suggest it's high enough to make test results useless.	{'neg': 0.158, 'neu': 0.694, 'pos': 0.148, 'compound': -0.0772}	-0.0772	neg

11	The main problem with claims about huge amounts of false positives is that they’re based on the assumption we know the prevalence of Covid-19 in the population being tested.	{'neg': 0.078, 'neu': 0.756, 'pos': 0.166, 'compound': 0.4588}	0.4588	pos
12	It assumes that because Office for National Statistics surveillance shows that 0.1% of the population is infected with Covid-19, this can be used as pre-test probability for Covid-19 tests.	{'neg': 0.103, 'neu': 0.897, 'pos': 0.0, 'compound': -0.4939}	-0.4939	neg
13	However, most testing is undertaken with symptomatic people or those who have been in contact with someone who has tested positive.	{'neg': 0.0, 'neu': 0.847, 'pos': 0.153, 'compound': 0.5574}	0.5574	pos

Fig. Textblob Result

	Text	**scores**	**comp_score**
0	Claims that Covid-19 testing gives too many false positive results have led some to believe they shouldn’t be relied on to shape responses to the pandemic.	0.109090909	pos
1	The argument stems from a genuine issue with medical tests about the risk of false positive results.	0.056818182	pos
2	It’s certainly true that some tests which show as positive may be false positives.	0.059090909	pos
3	It’s equally true that some of the negative results will be false negatives.	-0.116666667	neg

4	This Insight examines what can affect test results and explain why we can't always apply general mathematic principles to specific situations (like Covid-19)	0.025	pos
5	For a test to pick up a high proportion of genuine cases, it depends on a combination of the sensitivity and specificity of the test and the population being tested.	0.28	pos
6	Sensitivity looks at how well a test can identify true positive cases.	0.288636364	pos
7	A test that is very sensitive will flag up almost everyone who has a disease and won't give you many false negative results	-0.0175	neg
8	Specificity looks at a test's ability to correctly give a negative result for people who really don't have the disease.	-0.05	neg
9	So, a Covid-19 test with 90% specificity will correctly tell 90 people in every hundred that they don't have Covid-19 but will tell 10 others that they have Covid-19 when they don't	0	pos
10	Although we can't give an exact figure for the number of false positives, there's no evidence to suggest it's high enough to make test results useless.	-0.098	neg
11	The main problem with claims about huge amounts of false positives is that they're based on the assumption we know the prevalence of Covid-19 in the population being tested.	0.055555556	pos
12	It assumes that because Office for National Statistics surveillance shows that 0.1% of the population is infected with Covid-19, this can be used as pre-test probability for Covid-19 tests.	0	pos
13	However, most testing is undertaken with symptomatic people or those who have been in contact with someone who has tested positive.	0.363636364	pos

4.2 Supervised learning outputs for opinion detection

As expressed in Chapter 2, when supervised learning has been confirmed, it needs to do manual ground truth labelling. Below figure gives some examples of manual labels.

Sentences	Labels
Nurses have played a “vital” role in caring for people during the COVID-19 outbreak despite nearly 40,000 unfilled nursing post vacancies which had already put staff and services under strain.	1
the Department for Health & Social Care (DHSC) does not understand the nursing needs of the NHS	1
the COVID-19 outbreak is just the latest illustration of the risks and problems inherent in this strategy.	0
Even before the pandemic, the NHS’ own numbers show tens of thousands of their nurses were leaving the profession every year	0
This is not good enough for the over-stretched NHS workforce”.	1
The pace of progress on increasing the number of NHS nurses was already “too slow”, with efforts to increase the numbers in undergraduate nursing degrees – which anyway take years to come to come to fruition - having “signally failed”.	0
It is vital that the NHS protects the mental health and well-being of nurses who have contributed so much during the COVID-19 outbreak	0
We fully recognise that the NHS is reeling under the strain of Covid-19, with staff unsure how they will cope with the second wave that it seems clear already upon us.	0
It must press on with coherent plans to get the nursing workforce back to capacity, under the kind of working conditions that can encourage hard-won, hard-working nurses to stay in our NHS and care homes.”	1
None of the actual “Restoration and Renewal Programme” work has yet begun	0

In January 2018, Parliament approved the Programme to repair the Palace of Westminster and consider “wider objectives” such as improving accessibility and providing educational facilities.	0
It poses a very real risk to health and safety in its current state.	1
The restrictions of the pandemic may provide an opportunity in this context.	1

And the paper has shown how each model works from previous chapter, below figure is showing the model accuracy after implementing all the models.

Model Name	Naïve_Bayes	Naïve_Bayes	Naïve_Bayes	Naïve_Bayes
Feature Extraction	Count Vectors	Word Level TF-IDF	N-gram Level TF-IDF	Character Level TF-IDF
Model Accuracy	0.6111	0.4444	0.3889	0.4444
Model Name	Logistic Regression	Logistic Regression	Logistic Regression	Logistic Regression
Feature Extraction	Count Vectors	Word Level TF-IDF	N-gram Level TF-IDF	Character Level TF-IDF
Model Accuracy	0.5556	0.3889	0.3889	0.3889
Model Name	SVM	Random Forest	Random Forest	xgboost
Feature Extraction	N-gram Level TF-IDF	Count Vectors	Word Level TF-IDF	Count Vectors
Model Accuracy	0.3889	0.3889	0.5556	0.7222
Model Name	xgboost	xgboost	Neural Network	CNN

Feature Extraction	Word Level TF-IDF	Character Level TF-IDF	N-gram Level TF-IDF	NA
Model Accuracy	0.5556	0.4444	0.6667	0.6667
Model Name	RNN-LSTM	RNN-GRU	RNN-Bidirectional	RCNN-Bidirectional
Feature Extraction	NA	NA	NA	NA
Model Accuracy	0.6667	0.6667	0.6667	0.6667

It can find the Xgboost with count feature vectors have the best performance of all the models. This is probably because Xgboost is an ensemble method that works by boosting trees. Xgboost makes use of a gradient descent algorithm which is to correct the previous mistake done by the model, learn from it and its next step improves the performance. Then the previous results are rectified, and performance is enhanced.

GBM is a boosting method, which builds on weak classifiers. The idea is to add a classifier at a time, so that the next classifier is trained to improve the already trained ensemble. Notice that for RF each iteration the classifier is trained independently from the rest.

4.3 Feature clustering outputs for Event Detection

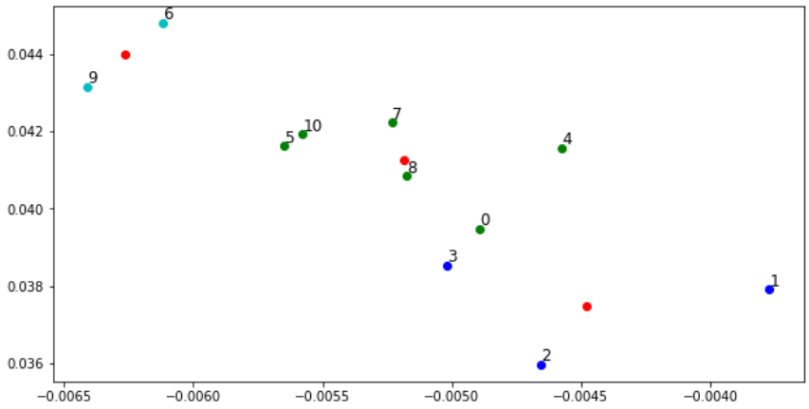
The paper has explained feature clustering has three steps processes in previous chapter. The below figure is the output for second step from SpaCy noun chunking process.

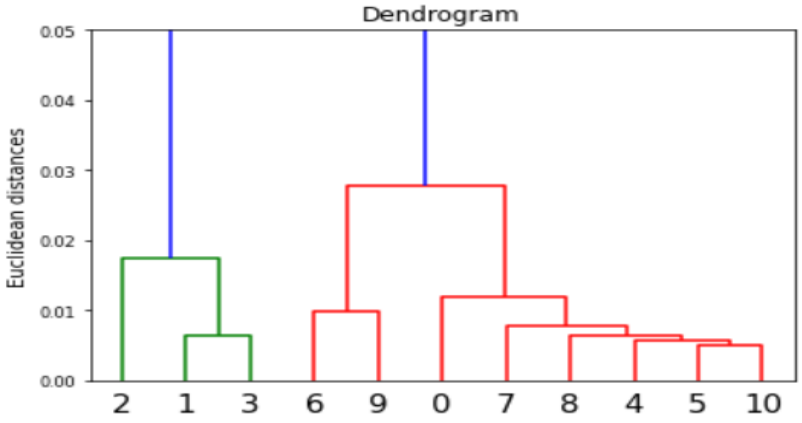
Sentences	Chunked Text
In its report published today, the Public Accounts Committee says that despite being involved in a 2016 cross-government exercise on dealing with a pandemic, the	the Public Accounts Committee ', 'a 2016 cross-government exercise ', 'a pandemic', 'the Department ',

Department for Education (DfE) had ‘no plan’ and was ‘unprepared’ for the challenges of Covid-19.	'Education ', 'DfE', '‘no plan', 'the challenges '
The Public Accounts Committee today reports concerns that the BBC, ‘at a critical juncture’ in its history, appears ‘complacent and unconcerned’ in the face of a series of commercial and financial challenges	The Public Accounts Committee ', 'concerns ', 'the BBC', 'a critical juncture', 'its history', 'the face ', 'a series ', 'commercial and financial challenges'
In its report published today the Public Accounts Committee says Government has set ambitious targets to phase out new petrol and diesel cars by 2030 and for all new cars to be zero-emission from 2035, but with just 11% of new car registrations for ultra-low emission cars in 2020 it will be a “huge challenge” to get this to 100% in the next 14 years.	its report ', 'the Public Accounts Committee ', 'Government ', 'ambitious targets ', 'new petrol ', 'diesel cars ', 'all new cars ', 'zero-emission ', 'just 11% ', 'new car registrations ', 'ultra-low emission cars ', 'it ', 'a “huge challenge', 'this to 100% ', 'the next 14 years'
The Industrial Strategy Challenge Fund (ISCF) was set up to help “address some of the complex issues” the UK economy faces, including long-term low productivity and living standards. Managed by UK Research and Innovation (UKRI) under the Business department (BEIS), it is designed around four ‘grand challenges’: future mobility; clean growth; artificial intelligence and data; and the ageing society.	The Industrial Strategy Challenge Fund ', 'ISCF', 'the complex issues', 'the UK economy ', 'long-term low productivity ', 'living standards', 'UK Research ', 'Innovation ', 'UKRI', 'the Business department ', 'BEIS', 'it ', 'four ‘grand challenges', 'future mobility', 'clean growth', 'artificial intelligence ', 'data', 'the ageing society'
Public Accounts Committee today warns HM Treasury and HMRC have a “very limited view of the role of tax”, with a “limited understanding of the environmental impact” of taxes and were unable to explain to the Committee “how the tax system is used in achieving the government’s environmental goals”.	Public Accounts Committee ', 'HM Treasury ', 'HMRC ', 'a “very limited view ', 'the role ', 'tax', 'a “limited understanding ', 'the environmental impact', 'taxes', 'the Committee ', 'the tax system ', 'the government', 'environmental goals'

In its report published today the Public Accounts Committee warns that the Ministry of Defence' (MoD) "neglect" of the accommodation for more than half of the Armed Forces is a risk to retention of service personnel and ultimately "directly undermines operational capability". In 2020 29% of service personnel living in the Single Living Accommodation (SLA) said accommodation was a factor increasing their intention to leave.	its report ', 'the Public Accounts Committee ', 'the Ministry ', 'Defence', '(MoD', 'the accommodation ', 'more than half ', 'the Armed Forces ', 'a risk ', 'retention ', 'service personnel ', 'operational capability', '2020 29% ', 'service personnel ', 'the Single Living Accommodation (SLA', 'accommodation ', 'a factor ', 'their intention '
In its report published today the Public Accounts Committee says the Government's "quickly drawn up", centrally-directed scheme to support those most vulnerable to covid-19 disease who were instructed to "shield" at home "suffered from the problems of poor data and a lack of joined up systems that we see all too often in government programmes".	its report ', 'the Public Accounts Committee ', 'the Government', 'centrally-directed scheme ', 'disease ', 'who ', 'home ', 'the problems ', 'poor data ', 'a lack ', 'systems ', 'we ', 'government programmes'

Below figure is the K-means effect for the final step to form three clusters and hierarchy dendrogram which indicates a clearer visualization in form of a tree showing the order and distances of merges during the hierarchical clustering.

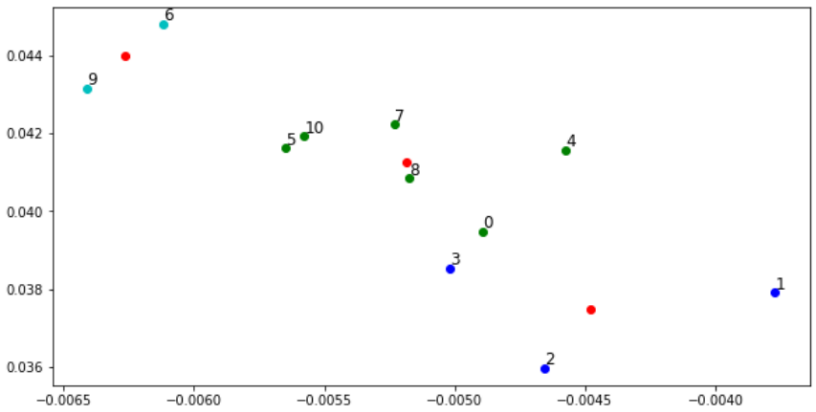


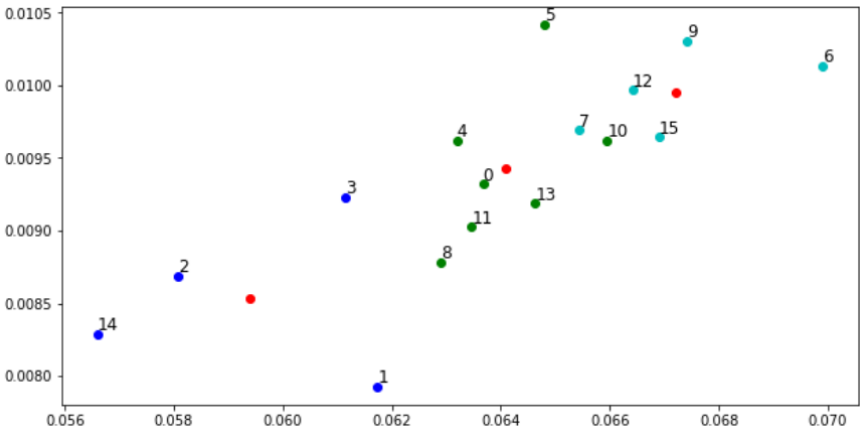


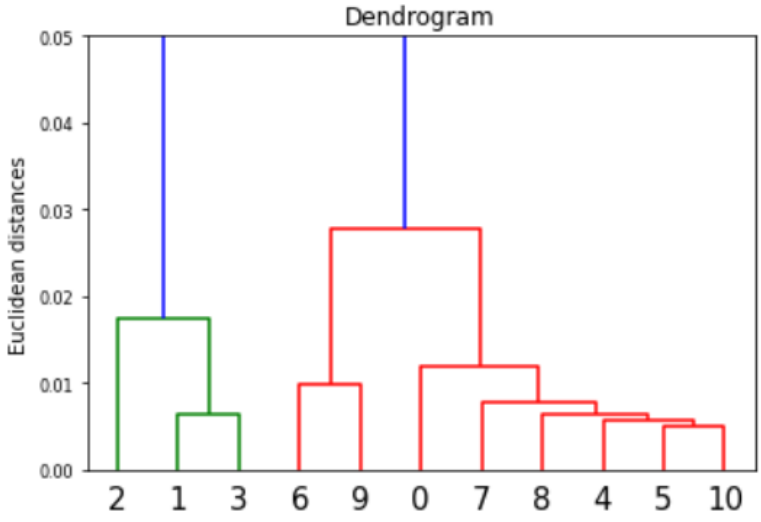
What it can find is different kinds of events represented by diverse colors. The light blue cluster is explaining related to pandemic, the deep blue cluster is about challenges, the green cluster is for the Public Accounts Committee. It also tells what is the closest one from the other one in a cluster. For example, point 10 has the shortest distance to point 5 in their cluster.

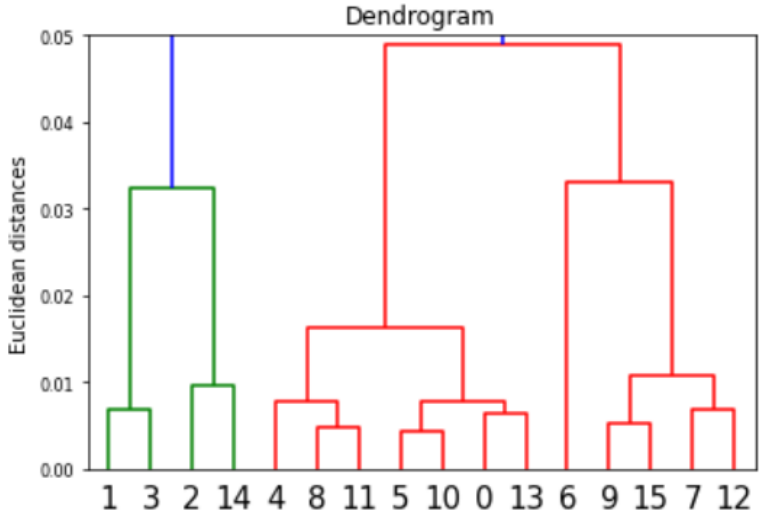
4.4 Cluster monitoring outputs for Trend Monitoring

As mentioned before, the paper is using cluster monitoring to find the trend change at different time. Fig below shows the comparison of before and after and hierarchy dendrogram shares a clearer comparison. It can conclude the clusters become larger with the number of events increasing. It can also see that the green cluster has the growing trend.




0.0105
0.0100
0.0095
0.0090
0.0085
0.0080
0.056
0.058
0.060
0.062
0.064
0.066
0.068
0.070
5
9
6
12
7
10
15
4
0
3
13
11
8
2
14
1


Dendrogram
Euclidean distances
0.05
0.04
0.03
0.02
0.01
0.00
2 1 3 6 9 0 7 8 4 5 10


Dendrogram
Euclidean distances
0.05
0.04
0.03
0.02
0.01
0.00
1 3 2 14 4 8 11 5 10 0 13 6 9 15 7 12

Chapter 5

Analysis of experimented approaches

The paper has demonstrated how to proceed the most known approaches for sentiment analysis, opinion detection, event detection, trend monitoring and received experimental results. This section will compare these methods (i.e., Automated sentiment analyzed tools, Supervised Learning and Deep Learning, Unsupervised Learning and Constituency Parsing, Clustering and Trend Analysis) and interpret the comparison by going through the weakness and strength of each of these approaches.

5.1 The automated analyzed tools for sentiment

In this paper, it applies the automated sentiment analyzed tools, e.g., VADER and Textblob for doing sentiment analysis. The advantages of using these tools are:

1) They contain reliable framework to produce classification. For instance, in Hutto and Gilbert (2014), VADER is described as a popular word list- and rule-based procedure which computes a continuous score for each text (ranging from negative to neutral to positive values) and appends a total sentiment score called compound[3] (Jacobs, Arthur M., 2019). Also, A. Amin, I. Hossain, et al, (2019) mentioned that VADER has a gold-standard sentiment lexicon which is significant to categorize semantic orientation for sentiment analysis models. Based on the information from the Internet, VADER and Textblob are the most popular methods and packages and have been widely used in many cases in the market.

[3] The compound score is a normalized weighted composite score ranging from−1/negative to 1/positive. According to the authors, a word or text segment has a positive sentiment if its compound score >= 0.05, a negative one if it <= 0.05. If the score falls in between these two threshold values, the text is considered neutral.

2) It is simplified manipulation to obtain the results of classification. With pre-set analyzer, all to do is feeding the textual data into the model and will output the result in a short time. The entire evaluating process does not require additional processing.

However, these tools have certain disadvantages on the other side:

1) The limited coverage of sentiment words. The existing tools have been found to be more successful for dealing with social media texts. For example, VADER is more sensitive to sentiment expressions in social media contexts though it can also be generalized to other domains (Shihab Elbagir and Jing Yang, 2019). When looking at the experimented result, it can find the classification from the tools may not quite convincible. That is mainly because our dataset is not composed of the words of dictionary the tools contain. And lexicon approach relies on a lexicon of words with pre-calculated polarity (Zahoor, S., and R. Rohilla., 2020). The limitation of covering sentiment words in other domain in the dictionary of words thereby will have a significant impact on evaluation process.

2) Failed to recognize spelling error or special expression. The existing tools have not been able to distinguish misspellings and grammatical mistakes, yet which may cause the analysis to overlook important words or usage. And it would impact on accuracy - possible misclassification if discriminating jargon, nomenclature, memes, or turns of phrase are used. (Jayson DeLancey, 2020).

5.2 Supervised learning and Deep learning

For opinion detection, the paper employs supervised learning models and deep learning models in this paper. There are quite a few benefits of supervised learning and deep learning models and with two state-of-art techniques, it is allowed to identify valuable opinion among a great amount of textual information:

1) No human intervention needed to identify trends and patterns. Machine learning

technology including supervised learning is a branch of artificial intelligence with which systems can learn from data, detect patterns, and make decision without much human intervention. Its capability to analyze bigger, more complex data gives an organization to automatically identify what they want. It is a good practice to get insights for complex and a great amount of textual information by using machine and deep learning models.

2) Produce predicted or classified result with good accuracy for unseen data. Supervised learning is major practical models of machine learning, which are good at classification and regression problems, and can provide high accuracy. They are trained until it can detect the underlying patterns and relationships between the input data and the output labels, enabling it to yield accurate labeling results when presented with never-before-seen data. As supervised learning has various algorithms, it is able to select the best one by comparison. The paper has around 70% accuracy with the best model to output the classified result.

3) More advanced models to provide more accurate result. The paper not only performs machine learning techniques but also deep learning framework. Deep learning is part of a broader family of machine learning methods based on artificial neural networks with representation learning. It is known as using multiple layers to progressively extract higher-level features from the raw input. It can expect to provide more accurate result to tell positive or negative about keywords.

Nevertheless, supervised learning and deep learning have room to improve:

1) Unwanted data downs the efficiency and improvement. Supervised learning can produce misclassification if input data has unwanted data which will be then used into feature extraction. This would prevent improvement of supervised learning to proceed classification process when the input data is mixed with unwanted data.

2) Computation time can be vast for supervised learning. Not an algorithm works

best for all problems. In machine learning, there is no one algorithm works best for every problem. As a result, it needs to try many different algorithms for our problem and select the one with best performance. It also implies that it is unable to deliver the results until experiment completion. Our experiment took five minutes to finish all the training and text process for more than one hundred sentences. When it is imbued with a great amount of textual information, the computation time will be much longer.

3) Human annotation requires human energy and time. Supervised learning need labels. Human annotation is slow to apply and is hard to ensure validity and reliability (Claire Cardie & John Wilkerson, 2008). Because of supervised learning models requires training data with labels, they are more difficult to build than unsupervised learning algorithms, even though supervised learning algorithms off-the-shelf are being increasingly implemented in the public domain for a wide variety of text tasks.

4) More preconditions for deep learning and hard to interpret how the models work. Deep learning is known to require a great deal amount of data and GPUs machine so that it can achieve good performance. And deep learning is kind of black box, so, it is not easy to interpret the training process.

5.3 Unsupervised learning and Constituency Parsing

In terms of event detection, in this paper unsupervised learning is chosen to identify events. Unsupervised learning and constituency parsing has been found with many strengths:

1) Beneficial to clustering if keywords of the event have been showed. Constituency parsing would help to extract noun phrases of the sentences, which is beneficial to improve the accuracy of clustering process.

2) Implementation without labels. Unsupervised learning can implement on datasets

without labels. It not only can save the process of costly annotation for large datasets but still is able to find structure with its input. Clustering can help to find the internal group. If the sentences are similar, they would gather closely with each other.

3) Easy to demonstrate the results through visualization. It is obviously shown in the graph that sentences with proximity idea are clustered to form a class while the rest of sentences are stayed in other places to represent another class. From visualized graphs, events are easily detected based on clusters. The extra point for unsupervised learning is that it can be relied on to find outliers of data. If there is a point which has long distance from others, it would be worthy to have it explored.

But everything is a double-edged sword. Unsupervised learning has its drawbacks:

1) No specific way to evaluate the model. Since no labels are provided, there is no specific way to compare model performance in most unsupervised learning methods. It cannot be known how many classes the data will divide into before the experiment.

2) Hard to keep class information with time. Clusters would change when there are more data involving. It would not be beneficial to keep same class information while increasing the volume of data.

5.4 Unsupervised learning and Constituency Parsing

Regarding trend monitoring, clustering realized by unsupervised learning and trend analysis for clusters are introduced in this paper to watch the event trends. It can find certain advantages when using these methods:

1) Clusters indicate a trend： The paper can benefit from taking notice of the trend when a cluster is growing due to trend monitoring. When a cluster grows fast and

big, there has been an increasing trend for this event, and vice versa. It is worthy to watch the trends and respond to these trends in line with business goals.

2) Comparison keeps periodical monitoring: trend analysis allows to compare similarities and differences across periods. By performing comparison at different time, it can know how many events occurred and can see how the old events evolve so that the process of events trend monitoring is fulfilled.

Whereas, for every plus there is a minus. This method has disadvantages:

1) Historical data has limitation and possible time complexity: Historical data may not be an accurate representation of a trend. And when dealing with a large dataset, conducting a clustering technique will crash the computer due to a magnitude of computational load and memory limits.

2) Lacks consistency. Class information is likely to change when the experiment is going through new gathering process. As a result, it is neither effortless to observe clustering change at different points of time nor convenient to determine the cause of a trend.

Chapter 6

Suggested improvements

The analysis for experimented approaches has disclosed the weakness and strength of each of these approaches. It also has provided us with insights into exploring tasks namely sentiment analysis, opinion detection, event detection and trend monitoring with more meaningful ways in methodology.

6.1 Introduction

It has known that supervised NLP models use the best approximating mapping learned during training to analyze unforeseen input data (never seen before) to accurately predict the corresponding output. Supervised learning models have advantages of typically capable of achieving excellent levels of performance (Aisera, 2020).

However, it has disadvantages of producing incorrect misclassification and long computation time when unwanted data is involved, which might cause an unfriendly input-output mapping preventing from improving performance. It is found that chunking with the help of dependency parsing may help to handle with unwanted data when constituency parsing is being done to extract names which is beneficial for clustering process to recognize events.

If could combine chunking by means of dependency parsing with supervised learning models, it is possible to expect unwanted data will be removed, input and output mapping will be adjusted with representative data, model performance will be improved, and computation time will be reduced accordingly.

6.2 Proposed Method

This paper proposes to maintain models of supervised learning and deep learning but

with the blend of chunking by dependency parsing for classification. It will be different from traditional models: supervised learning will continue to work on making future predictions on labeled data and dependency parsing tool will focus on extract the main components from long sentences for input data before feeding into the models. Ultimately, the paper can compare the accuracy before and after, and see how the performance of models have been improved. The introduction will start with chunking for relationship, dependency parsing and models.

6.2.1 Chunking for relationship

Chunking got a lot of attention when syntactic parsing was predominantly driven by constituency parsing (Tjong Kim Sang and Buchholz, 2000). Chunking in constituency parsing is a process of extracting phrases from unstructured text, which means analyzing a sentence to identify the constituents (Noun Groups, Verbs, verb groups, etc.) However, it does not specify their internal structure, nor their role in the main sentence. Many studies have applied noun-chunking or verb-chunking for certain purpose. That is because chunking can break sentences into phrases that are useful to yield meaningful results. This is also shown in experimented clustering process to detect events. The representative phrases are getting closer if they are similar with each other, which as result to form an event cluster. This also brings better results if the input sentences removed irrelevant individual words.

But in this chapter, it will use chunking to explore relationship. This is mainly because the internal structure of a sentence will tell the writer's main speaking ideas if the sentence is long and complex. And our datasets have quite a few complex sentences. For example, the sentence: "The Committee expects a report back from the Cabinet Office, by September 1st, on Government's progress on a "second wave ready" plan." The sentence is shown to be very long. However, the main idea the writer wants to convey is 'The committee expects a report back'. As said, the model can identify the trends and pattern without human intervention. If feeding all sentence into the model, the model will digest all the information and learn to identify the trends and pattern, which will impact on model

performance- possible low accuracy and misclassification due to irrelevant information.

How chunking for relationship works? The paper will introduce dependency parsing to achieve the goal.

6.2.2 Dependency parsing

Dependency parsing is the process of analyzing the grammatical structure of a sentence based on the dependencies between the words in a sentence. The reason to choose dependency parsing because 1) Dependency parsing has ability to deal with languages that are morphologically rich and have a relatively free word order (Daniel Jurafsky et all, 2021). By doing so, the paper can find the "head" words and words, which modify those heads even there are not word order. 2) Further, according to Covington (2001), dependency links are close to the semantic relationships needed for the next stage of interpretation. It means it can tell what the subjects and objects of a verb from dependency parsing, which is useful in applications like information extraction.

How to chunk based on dependency parsing? Chunking works on top of POS tagging, it uses pos-tags as input and provides chunks as output. There are also a standard set of Chunk tags like Noun Phrase (NP), Verb Phrase (VP), etc. As said, the paper is eager to find semantic relationships between words and locate the main structure of the sentence (e.g., subjects, verbs, objects of a verb). SpaCy has a good parser which not only provides POS tagging for chunking noun or verb phrases but also offers dependency tagging (DEP). Therefore, it can easily chunk the targeted components of the sentence by tags of SpaCy.

6.2.3 Supervised learning models and Deep learning models

The paper will employ the same dataset and modelling including all the sentences and all kinds of old models (i.e., supervised learning and deep learning) for new experiment with our proposed method so that it can compare the results before and after and know how the method will help to manage unwanted data and further to improve the accuracy. For the method details of how to do the modelling, they will be provided in Chapter 2.

Chapter 7

Experimental Results

In this section, the paper outlines in detail the results from dependency parsing and report the model results obtained. This paper also demonstrated the comparison with and without the proposed method.

7.1 Chunked text from dependency parsing

As said, the dataset has 69 sentences, and it is going to use dependency parser to chunk for each sentence. The chunked text will carry the subject, predicate and possible object or object of a preposition of the sentence. The labels follow the same rule without change. That is if the sentence is describing reality, what has happened, that means it existed or exists in the world, it will be put 0. It will be labelled 1 if the sentence is indicating writer's thinking and believe, by other words, arguments need to be verified. Figure below is the examples of chunked texts of some sentences.

Sentences	Chunked Text	Labels
Nurses have played a "vital" role in caring for people during the COVID-19 outbreak despite nearly 40,000 unfilled nursing post vacancies which had already put staff and services under strain.	Nurses have played a " vital " role	1
the Department for Health & Social Care (DHSC) does not understand the nursing needs of the NHS	the Department for Health & Social Care (DHSC) does not understand the nursing needs	1
the COVID-19 outbreak is just the latest illustration of the risks and problems inherent in this strategy.	the COVID-19 outbreak is just the latest illustration	0

Even before the pandemic, the NHS' own numbers show tens of thousands of their nurses were leaving the profession every year	the NHS's own numbers show tens of thousands of their nurses were leaving the profession	0
This is not good enough for the over-stretched NHS workforce".	This is not good enough	1
The pace of progress on increasing the number of NHS nurses was already "too slow", with efforts to increase the numbers in undergraduate nursing degrees – which anyway take years to come to come to fruition - having "signally failed".	The pace of progress was already "too slow"	0
It is vital that the NHS protects the mental health and well-being of nurses who have contributed so much during the COVID-19 outbreak	It is vital that the NHS protects the mental health	1
the Committee is also concerned that the necessary safeguards being put in place to protect Black, Asian and minority ethnic staff, who are disproportionately affected by COVID-19, restrict their work experience and career progression.	the Committee is concerned that	0
After a highly critical PAC report earlier this year, when the Committee Chair said care homes had been "thrown to the wolves" in the pandemic, today's report finds "the nursing needs of social care remain an unaddressed afterthought for the Department of Health & Social Care."	today's report finds that	0
Vacancies for nurses in social care increased from 4% in 2012-13 to 10% in 2018-19, while the number of registered nursing posts in social care has fallen by 20% since 2012-13.	Vacancies increased from 4% in 2012-13 to 10% in 2018-19	0
we are facing an emerging crisis in nursing	we are facing an emerging crisis	0

We fully recognise that the NHS is reeling under the strain of Covid-19, with staff unsure how they will cope with the second wave that it seems clear already upon us.	We fully recognise that	0
It must press on with coherent plans to get the nursing workforce back to capacity, under the kind of working conditions that can encourage hard-won, hard-working nurses to stay in our NHS and care homes.”	It must press on	1
None of the actual “Restoration and Renewal Programme” work has yet begun	None has yet begun	0
In January 2018, Parliament approved the Programme to repair the Palace of Westminster and consider “wider objectives” such as improving accessibility and providing educational facilities.	In January 2018, Parliament approved the Programme	0
It poses a very real risk to health and safety in its current state.	It poses a risk	1
The restrictions of the pandemic may provide an opportunity in this context	The restrictions may provide an opportunity	1

7.2 Model results with and without proposed method

As said, the modelling process is following the old way which is detailed explained in Chapter 2. The difference is the models are fed with chunked text as input instead of the whole sentence. To demonstrate the improvement, the paper attaches two tables so that it can compare the accuracy before and after. The prior and latter figure represent the model results with and without proposed method separately.

Figure- the model results with proposed method:

Model Name	Naïve_Bayes	Naïve_Bayes	Naïve_Bayes	Naïve_Bayes
Feature Extraction	Count Vectors	Word Level TF-IDF	N-gram Level TF-IDF	Character Level TF-IDF
Model Accuracy	0.72222222	0.72222222	0.5	0.72222222
Model Name	Logistic Regression	Logistic Regression	Logistic Regression	Logistic Regression
Feature Extraction	Count Vectors	Word Level TF-IDF	N-gram Level TF-IDF	Character Level TF-IDF
Model Accuracy	0.72222222	0.77777778	0.55555556	0.77777778
Model Name	SVM	Random Forest	Random Forest	xgboost
Feature Extraction	N-gram Level TF-IDF	Count Vectors	Word Level TF-IDF	Count Vectors
Model Accuracy	0.5	0.72222222	0.83333333	0.83333333
Model Name	xgboost	xgboost	Neural Network	CNN
Feature Extraction	Word Level TF-IDF	Character Level TF-IDF	N-gram Level TF-IDF	NA
Model Accuracy	0.72222222	0.66666667	0.5	0.5
Model Name	RNN-LSTM	RNN-GRU	RNN-Bidirectional	RCNN-Bidirectional
Feature Extraction	NA	NA	NA	NA
Model Accuracy	0.5	0.5	0.5	0.5

Figure- the model results without proposed method:

Model Name	Naïve_Bayes	Naïve_Bayes	Naïve_Bayes	Naïve_Bayes
Feature Extraction	Count Vectors	Word Level TF-IDF	N-gram Level TF-IDF	Character Level TF-IDF
Model Accuracy	0.6111	0.4444	0.3889	0.4444
Model Name	Logistic Regression	Logistic Regression	Logistic Regression	Logistic Regression
Feature Extraction	Count Vectors	Word Level TF-IDF	N-gram Level TF-IDF	Character Level TF-IDF
Model Accuracy	0.5556	0.3889	0.3889	0.3889
Model Name	SVM	Random Forest	Random Forest	xgboost
Feature Extraction	N-gram Level TF-IDF	Count Vectors	Word Level TF-IDF	Count Vectors
Model Accuracy	0.3889	0.3889	0.5556	0.7222
Model Name	xgboost	xgboost	Neural Network	CNN
Feature Extraction	Word Level TF-IDF	Character Level TF-IDF	N-gram Level TF-IDF	NA
Model Accuracy	0.5556	0.4444	0.6667	0.6667
Model Name	RNN-LSTM	RNN-GRU	RNN-Bidirectional	RCNN-Bidirectional
Feature Extraction	NA	NA	NA	NA
Model Accuracy	0.6667	0.6667	0.6667	0.6667

7.3 Conclusion for experiment results

The model accuracy obtained from the proposed method in machine learning has been improved. Specially, Naïve Baye with count vectors features is improved from 0.61 to 0.72, with Word Level TF-IDF is increased from 0.44 to 0.72, with N-gram Level TF-IDF is advanced from 0.39 to 0.5, with Character Level TF-IDF is enhanced from 0.44 to 0.72. Logistic regression with count vectors features is improved from 0.56 to 0.72, with Word Level TF-IDF is increased from 0.39 to 0.78, with N-gram Level TF-IDF is advanced from 0.39 to 0.56, with Character Level TF-IDF is enhanced from 0.39 to 0.78.

SVM with N-gram Level TF-IDF is improved from 0.39 to 0.5 while Random Forest with count vectors features is increased from 0.39 to 0.72 and with Word Level TF-IDF is enhanced from 0.56 to 0.83. Xgboost with count vectors features is improved from 0.72 to 0.83, with Word Level TF-IDF is increased from 0.56 to 0.72, with Character Level TF-IDF is enhanced from 0.44 to 0.67. However, the model accuracy obtained from the proposed method in deep learning has been weakened. Neural Network with N-gram Level TF-IDF is reduced from 0.67 to 0.5 while other deep learning models including CNN, RNN-LSTM, RNN-GRU, RNN-Bidirectional, CNN-Bidirectional is decreased from 0.67 to 0.5.

Given the statistical results, chunking with dependency parsing contributes to the improvement of machine learning model performance but no contribution to deep learning models.

Chapter 8

Conclusion and future work

8.1 Conclusion

This thesis discussed the importance of analyzing online text data to gain a deeper understanding of writer opinions and thoughts using data mining, machine learning and deep learning techniques. The challenges associated with extracting meaningful information from online text include the specialty of the nature of data. As described, in detail the research problems undertaken as part of the author's master program, and the methods proposed to address them. First, it implemented the appropriated and most known methods: popular automatic classification tools for sentiment analysis, supervised learning and deep learning for opinion detection, constituency parsing and unsupervised learning for event detection, clustering and trend analysis for trend monitoring. Keeping in mind there is no perfect algorithm, it was able to analyze the weakness and strength of each of these approaches.

Second, the paper addressed the weakness of supervised learning which is unwanted data will have impact on input and output mapping, model performance and computation time. A method introducing chunking by dependency parsing into models was developed. Our method is able to decrease unwanted data and improve performance of models. It obtained high accuracies for supervised learning models, and simultaneously uncovered the proposed method is possibly used for other fields.

8.2 Future work

A dependency parsing framework combined with supervised learning models provides an improved approach in this paper. The improved approach is more sophisticated (and therefore more precise) than the simplistic approach of applying models. However, there

are some limitations of the experiment. a) One of the difficulties is human annotation. How to distinguish between facts and opinions, since actual reports can affect the information fusion process that enables a polarity analyzer to classify opinions (Chaturvedi, Iti, et al., 2018). Specially, what is the rule to differentiate two groups for annotating the labels of the dataset and if the manual labels will have an impact on the model accuracy and effectiveness. b) Due to a lack of time, the quantity of datasets is limited. As such, deep learning models cannot demonstrate their capability during the experiment because it only has 70 sentences for experiment which is far low from quantity requirements of deep learning models and increasing the volume of dataset to thousands level is not easy to accomplish. c) Dependency parsing can help machine learning to produce more accurate results. But it seems it cannot help deep learning to acquire high accuracy. What is the reason? What if includes massive data into dataset? Will the result be different? Will the chunking by dependency parsing facilitate deep learning similar as machine learning? The conclusion should be made based on experiments.

This is an improved approach developed in political domain. It is envisaged that this methodology could bring more insights to professional analysts as well as facilitate the public to get the intention of authorities. Future work includes: i) to increase the number of datasets substantially so that the paper can do the comparison between machine learning models and deep learning models. It is possible one of the deep learning models will beat the champion and become the best model. ii) the proposed method should be experimented on other tasks. In this paper, the proposed method is merely working on one task goal which is to detect opinions. It could also be used for tasks like sentiment analysis, event detection and event trends monitor. iii) the proposed method should be tested on other kinds of texts so that it can be known the method could be expanded to apply in other application areas.

The analysis of sentiment or opinions, the detection of events and the monitor of event trends play a central role in work in political communication (Lori Young & Stuart Soroka, 2012). With confidence, the improved approach indicated in this dissertation is applicable

and conducive for political or governmental context. But it is also expected to be improved and advanced for the sake of different areas.

References

1. Verma, Jai Prakash, and Smita Agrawal. “Big Data Analytics: Challenges And Applications For Text, Audio, Video, And Social Media Data.” International Journal on Soft Computing, Artificial Intelligence and Applications, vol. 5, no. 1, 2016, pp. 41–51.

2. A. Katal, M. Wazid and R. H. Goudar, "Big data: Issues, challenges, tools and Good practices," 2013 Sixth International Conference on Contemporary Computing (IC3), 2013, pp. 404-409, doi: 10.1109/IC3.2013.6612229.

3. Liddy, E.D. 2001. “Natural Language Processing.” In Encyclopedia of Library and Information Science, 2nd Ed. NY. Marcel Decker, Inc.

4. Fernández-Gavilanes, Milagros, et al (2016). Unsupervised Method for Sentiment Analysis in Online Texts. Expert Systems with Applications, vol. 58, Oct. 2016, pp. 57–75. ScienceDirect, doi:10.1016/j.eswa.2016.03.031.

5. Brandwatch, “Sentiment Analysis: How Does It Work? Why Should We Use It?”, https://www.brandwatch.com/blog/understanding-sentiment-analysis/. Accessed 16 June 2021.

6. What Is Unsupervised Learning? https://www.ibm.com/cloud/learn/unsupervised-learning. Accessed 8 June 2021.

7. Amir Gandomi, and Murtaza Haider (2014). Beyond the Hype: Big Data Concepts, Methods, and Analytics. International Journal of Information Management, vol. 35, no. 2, Apr. 2015, pp. 137–44. www.sciencedirect.com, doi:10.1016/j.ijinfomgt.2014.10.007.

8. Feldman, Ronen. “Techniques and Applications for Sentiment Analysis.” Communications of the ACM, vol. 56, no. 4, Apr. 2013, pp. 82–89. April 2013, doi:10.1145/2436256.2436274.

9. Walaa Medhat, Ahmed Hassan, Hoda Korashy “Sentiment Analysis Algorithms and Applications: A Survey.” Ain Shams Engineering Journal, vol. 5, no. 4, Dec. 2014, pp. 1093–113. www.sciencedirect.com, doi:10.1016/j.asej.2014.04.011.

10. V. Bobichev, O. Kanishcheva and O. Cherednichenko, "Sentiment analysis in the Ukrainian and Russian news," 2017 IEEE First Ukraine Conference on Electrical and Computer Engineering (UKRCON), 2017, pp. 1050-1055, doi: 10.1109/UKRCON.2017.8100410.

11. Abercrombie, Gavin, et al. “Policy Preference Detection in Parliamentary Debate Motions.” Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL), Association for Computational Linguistics, 2019, pp. 249–59. ACLWeb, doi:10.18653/v1/K19-1024.

12. Bhattacharjee, Kasturi. Opinion Detection, Sentiment Analysis and User Attribute Detection from Online Text Data. UC Santa Barbara, 2016. escholarship.org, https://escholarship.org/uc/item/4x85k62h.

13. S. Taj, B. B. Shaikh and A. Fatemah Meghji, "Sentiment Analysis of News Articles: A Lexicon based Approach," 2019 2nd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET), 2019, pp. 1-5, doi: 10.1109/ICOMET.2019.8673428.

14. A. Agarwal, V. Sharma, G. Sikka and R. Dhir, "Opinion mining of news headlines using SentiWordNet," 2016 Symposium on Colossal Data Analysis and Networking (CDAN), 2016, pp. 1-5, doi: 10.1109/CDAN.2016.7570949.

15. Kim, Erin Hea-Jin, et al. “Topic-Based Content and Sentiment Analysis of Ebola Virus on Twitter and in the News.” Journal of Information Science, vol. 42, no. 6, Dec. 2016, pp. 763–81. SAGE Journals, doi:10.1177/0165551515608733.

16. David Vilares, Yulan He. Detecting Perspectives in Political Debates; Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 1573–1582

17. Dhaoui, Chedia, et al. “Social Media Sentiment Analysis: Lexicon versus Machine Learning.” Journal of Consumer Marketing, vol. 34, no. 6, Jan. 2017, pp. 480–88. Emerald Insight, doi:10.1108/JCM-03-2017-2141.

18. Hutto, C.J. & Gilbert, E.E. (2014). VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text. Eighth International Conference on Weblogs and Social Media (ICWSM-14). Ann Arbor, MI, June 2014.

19. Jacobs, Arthur M. (2019) “Sentiment Analysis for Words and Fiction Characters From the Perspective of Computational (Neuro-)Poetics.” Frontiers in Robotics and AI, vol. 6, July 2019, p. 53. DOI.org (Crossref), doi:10.3389/frobt.2019.00053.

20. A. Amin, I. Hossain, A. Akther and K. M. Alam (2019), "Bengali VADER: A Sentiment Analysis Approach Using Modified VADER," 2019 International Conference on Electrical, Computer and Communication Engineering (ECCE), Cox'sBazar, Bangladesh, 2019, pp. 1-6, doi: 10.1109/ECACE.2019.8679144.

21. Hermansyah, Reza, and Riyanarto Sarno. “Sentiment Analysis about Product and Service Evaluation of PT Telekomunikasi Indonesia Tbk from Tweets Using TextBlob, Naive Bayes K-NN Method.” 2020 International Seminar on Application for Technology of Information and Communication (ISemantic), 2020, pp. 511–16. IEEE Xplore, doi:10.1109/iSemantic50169.2020.9234238.

22. Rajesh Bose, P. S. Aithal, Sandip Roy. “Sentiment Analysis on the Basis of Tweeter Comments of Application of Drugs by Customary Language Toolkit and TextBlob Opinions of Distinct Countries”.International Journal of Emerging Trends in Engineering Research, 8(7), July 2020, 3684 – 3696

23. Shah, Parthvi. “My Absolute Go-To for Sentiment Analysis — TextBlob.” Medium, 6 Nov. 2020, https://towardsdatascience.com/my-absolute-go-to-for-sentiment-analysis-textblob-3ac3a11d524.

24. Kohli, Pahul Preet Singh. “Sentiment Analysis - Methods and Pre-Trained Models Review.” Pahul Preet Singh Kohli, 1 Mar. 2020, https://pahulpreet86.github.io/sentiment-analysis-methods-and-pre-trained-models-review/.

25. TextBlob Sentiment: Calculating Polarity and Subjectivity. https://planspace.org/20150607-textblob_sentiment/. Accessed 8 June 2021.

26. Belbachir, Faiza, and Bénédicte Le Grand. “Opinion Detection: Influence Factors.” 2015 IEEE 9th International Conference on Research Challenges in Information Science (RCIS), 2015, pp. 522–23. IEEE Xplore, doi:10.1109/RCIS.2015.7128918.

27. Lilian Hobbs, Susan Hillson, Shilpa Lawande, Pete Smith, “The Impact of Features Extraction on the Sentiment Analysis.” Procedia Computer Science, vol. 152, Jan. 2019, pp. 341–48. www.sciencedirect.com, doi:10.1016/j.procs.2019.05.008.

28. Eman A.Abdel Maksoud, Sherif Barakat, MohammedElmogy “Medical Images Analysis Based on Multilabel Classification.” Machine Learning in Bio-Signal Analysis and Diagnostic Imaging, Jan. 2019, pp. 209–45. www.sciencedirect.com, doi:10.1016/B978-0-12-816086-2.00009-6.

29. Broda, Bartosz, et al. “Fextor: A Feature Extraction Framework for Natural Language Processing: A Case Study in Word Sense Disambiguation, Relation Recognition and Anaphora Resolution.” Computational Linguistics: Applications, edited by Adam Przepiórkowski et al., Springer, 2013, pp. 41–62. Springer Link, doi:10.1007/978-3-642-34399-5_3.

30. Eklund, Martin. Comparing Feature Extraction Methods and Effects of Pre-Processing Methods for Multi-Label Classification of Textual Data. 2018. www.diva-portal.org, http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-231438.

31. eiki. “Feature Extraction in Natural Language Processing with Python.” Medium, 14 Jan. 2019, https://medium.com/@eiki1212/feature-extraction-in-natural-language-processing-with-python-59c7cdcaf064.

32. Mehdi Allahyari et al. “A brief survey of text mining: Classification, clustering and extraction techniques”. In: arXiv preprint arXiv:1707.02919 (2017).

33. Zach CHASE, Nicolas Genain, and Orren Karniol-Tambour. “Learning Multi-Label Topic Classification of News Articles”. In: (2014).

34. PURVA HUILGOL “BoW Model and TF-IDF For Creating Feature From Text.” Analytics Vidhya, 27 Feb. 2020, https://www.analyticsvidhya.com/blog/2020/02/quick-introduction-bag-of-words-bow-tf-idf/.

35. BEEL, Joeran, Bela GIPP, Stefan LANGER, Corinna BREITINGER, 2016. Research-paper recommender systems : a literature survey. In: International Journal on Digital Libraries. 17(4), pp. 305-338. ISSN 1432-5012. eISSN 1432-1300. Available under: doi: 10.1007/s00799-015-0156-0

36. Uddin, Moahammad Nasir. Supervised Text Classification | Codementor. https://www.codementor.io/@nasiruddin630/supervised-text-classification-w9bp90nwf. Accessed 11 June 2021.

37. Tobias Schnabel et al. “Evaluation methods for unsupervised word embeddings”. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. 2015, pp. 298–307.

38. SANJANA REDDY. GloVe and FastText — Two Popular Word Vector Models in NLP | SAP Conversational AI Blog. 3 Apr. 2019, https://cai.tools.sap/blog/glove-and-fasttext-two-popular-word-vector-models-in-nlp/.

39. Shriram "Multinomial Naive Bayes Explained: Function, Advantages & Disadvantages, Applications in 2021." UpGrad Blog, 3 Jan. 2021, https://www.upgrad.com/blog/multinomial-naive-bayes-explained/.

40. Jake VanderPlas, Python Data Science Handbook 2016 | Python Data Science Handbook. https://jakevdp.github.io/PythonDataScienceHandbook/. Accessed 11 June 2021.

41. O. Aborisade and M. Anwar, "Classification for Authorship of Tweets by Comparing Logistic Regression and Naive Bayes Classifiers," 2018 IEEE International Conference on Information Reuse and Integration (IRI), 2018, pp. 269-276, doi: 10.1109/IRI.2018.00049.

42. S. T. Indra, L. Wikarsa and R. Turang, "Using logistic regression method to classify tweets into the selected topics," 2016 International Conference on Advanced Computer Science and Information Systems (ICACSIS), 2016, pp. 385-390, doi: 10.1109/ICACSIS.2016.7872727.

43. Shah, Kanish, et al. "A Comparative Analysis of Logistic Regression, Random Forest and KNN Models for the Text Classification." Augmented Human Research, vol. 5, no. 1, Mar. 2020, p. 12. Springer Link, doi:10.1007/s41133-020-00032-0.

44. Joachims, Thorsten. "A Statistical Learning Learning Model of Text Classification for Support Vector Machines." Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Association for Computing Machinery, 2001, pp. 128–36. ACM Digital Library, doi:10.1145/383952.383974.

45. Baoxun Xu, Xiufeng Guo, Yunming Ye, Jiefeng Cheng. “An Improved Random Forest Classifier for Text Categorization” JOURNAL OF COMPUTERS, VOL. 7, NO. 12, DECEMBER 2012 doi:10.4304/jcp.7.12.2913-2920

46. Islam, Md Zahidul, et al. “A Semantics Aware Random Forest for Text Classification.” Proceedings of the 28th ACM International Conference on Information and Knowledge Management, Association for Computing Machinery, 2019, pp. 1061–70. ACM Digital Library, doi:10.1145/3357384.3357891.

47. Chen, Tianqi, and Carlos Guestrin. “XGBoost: A Scalable Tree Boosting System.” Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Association for Computing Machinery, 2016, pp. 785–94. ACM Digital Library, doi:10.1145/2939672.2939785.

48. Z. Qi, "The Text Classification of Theft Crime Based on TF-IDF and XGBoost Model," 2020 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA), 2020, pp. 1241-1246, doi: 10.1109/ICAICA50127.2020.9182555.

49. F. Harrag and E. El-Qawasmah, "Neural Network for Arabic text classification," 2009 Second International Conference on the Applications of Digital Information and Web Technologies, 2009, pp. 778-783, doi: 10.1109/ICADIWT.2009.5273841.

50. N I Widiastuti Convolution Neural Network for Text Mining and Natural Language Processing 2019 IOP Conf. Ser.: Mater. Sci. Eng. 662 052010

51. Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. 2018. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. CoRR, abs/1803.01271.

52. Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. In Proceedings of the 52nd

Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 1: Long Papers, pages 655–665. The Association for Computer Linguistics.

53. Peng Wang, Jiaming Xu, Bo Xu, Cheng-Lin Liu, Heng Zhang, Fangyuan Wang, and Hongwei Hao. 2015. Semantic clustering and convolutional neural network for short text categorization. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 2: Short Papers, pages 352–357. The Association for Computer Linguistics.

54. Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada, pages 649–657.

55. Rie Johnson and Tong Zhang. 2015. Effective use of word order for text categorization with convolutional neural networks. In NAACL HLT 2015, The 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Denver, Colorado, USA, May 31 - June 5, 2015, pages 103–112. The Association for Computational Linguistics.

56. Mohit Iyyer, Varun Manjunatha, Jordan L. BoydGraber, and Hal Daume III. 2015. Deep unordered ´composition rivals syntactic methods for text classification. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers, pages 1681–1691. The Association for Computer Linguistics.

57. Ronan Collobert, Jason Weston, Leon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel P. Kuksa. 2011. Natural language processing (almost) from scratch. Journal of Machine Learning Research, 12:2493–2537.

58. Yoon Kim. 2014. Convolutional neural networks for sentence classification. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL, pages 1746–1751. ACL.

59. Zain Amin, Muhammad, and Noman Nadeem. “Convolutional Neural Network: Text Classification Model for Open Domain Question Answering System.” ArXiv E-Prints, Sept. 2018, p. arXiv:1809.02479.

60. H. Hu, M. Liao, C. Zhang and Y. Jing, "Text classification based recurrent neural network," 2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC), 2020, pp. 652-655, doi: 10.1109/ITOEC49072.2020.9141747.

61. Sari, Winda Kurnia, et al. Sequential Models for Text Classification Using Recurrent Neural Network. Atlantis Press, 2020, pp. 333–40. www.atlantis-press.com, doi:10.2991/aisr.k.200424.050.

62. Luo, Li-xia. “Network Text Sentiment Analysis Method Combining LDA Text Representation and GRU-CNN.” Personal and Ubiquitous Computing, vol. 23, no. 3, July 2019, pp. 405–12. Springer Link, doi:10.1007/s00779-018-1183-9.

63. Zulqarnain, M., et al. Efficient Processing of GRU Based on Word Embedding for Text Classification. 2019. Semantic Scholar, doi:10.30630/joiv.3.4.289.

64. Gangwar, Akhilesh Kumar, and Vadlamani Ravi. “A Novel BGCapsule Network for Text Classification.” ArXiv:2007.04302 [Cs], July 2020. arXiv.org, http://arxiv.org/abs/2007.04302.

65. Lu, Guangquan, et al. “Multi-Task Learning Using a Hybrid Representation for Text Classification.” Neural Computing and Applications, vol. 32, no. 11, June 2020, pp. 6467–80. Springer Link, doi:10.1007/s00521-018-3934-y.

66. Abdelgwad, Mohammed M., et al. “Arabic Aspect Based Sentiment Analysis Using Bidirectional GRU Based Models.” ArXiv:2101.10539 [Cs], Mar. 2021. arXiv.org, http://arxiv.org/abs/2101.10539.

67. Xie P, Wang G, Zhang C, Chen M, Yang H, Lv T, Sang Z, Zhang P. Bidirectional Recurrent Neural Network And Convolutional Neural Network (BiRCNN) For ECG Beat Classification. Annu Int Conf IEEE Eng Med Biol Soc. 2018 Jul;2018:2555-2558. doi: 10.1109/EMBC.2018.8512752. PMID: 30440929.

68. Zhang, Jingren, et al. “Feature Fusion Text Classification Model Combining CNN and BiGRU with Multi-Attention Mechanism.” Future Internet, vol. 11, no. 11, Nov. 2019, p. 237. www.mdpi.com, doi:10.3390/fi11110237.

69. Mellin, Jonas, and Mikael Berndtsson. “Event Detection.” Encyclopedia of Database Systems, edited by LING LIU and M. TAMER ÖZSU, Springer US, 2009, pp. 1035–40. Springer Link, doi:10.1007/978-0-387-39940-9_506.

70. Unankard, Sayan (2015). “Event detection in social networks.” PhD Thesis, School of Information Technology and Electrical Engineering, The University of Queensland. https://doi.org/10.14264/uql.2015.545

71. F. Jiang, Y. Wu and A. K. Katsaggelos, "A Dynamic Hierarchical Clustering Method for Trajectory-Based Unusual Video Event Detection," in IEEE Transactions on Image Processing, vol. 18, no. 4, pp. 907-913, April 2009, doi: 10.1109/TIP.2008.2012070.

72. Pohl, D., Bouchachia, A. & Hellwagner, H. Social media for crisis management: clustering approaches for sub-event detection. Multimed Tools Appl 74, 3901–3932 (2015). https://doi.org/10.1007/s11042-013-1804-2

73. Ghaemi, Zeinab, and Mahdi Farnaghi. “A Varied Density-Based Clustering Approach for Event Detection from Heterogeneous Twitter Data.” ISPRS International Journal of Geo-Information, vol. 8, no. 2, Feb. 2019, p. 82. www.mdpi.com, doi:10.3390/ijgi8020082.

74. Ena, O., Mikova, N., Saritas, O. et al. A methodology for technology trend monitoring: the case of semantic technologies. Scientometrics 108, 1013–1041 (2016). https://doi.org/10.1007/s11192-016-2024-0

75. Leonid Gokhberg, Ilya Kuzminov, Elena Khabirova, Thomas Thurner “Advanced Text-Mining for Trend Analysis of Russia’s Extractive Industries.” Futures, vol. 115, Jan. 2020, p. 102476. www.sciencedirect.com, doi:10.1016/j.futures.2019.102476.

76. Lam, W., et al. “Using Contextual Analysis for News Event Detection.” International Journal of Intelligent Systems, vol. 16, no. 4, 2001, pp. 525–46. Wiley Online Library, doi:https://doi.org/10.1002/int.1022.

77. Linmei Hu, Bin Zhang, Lei Hou, Juanzi Li “Adaptive Online Event Detection in News Streams.” Knowledge-Based Systems, vol. 138, Dec. 2017, pp. 105–12. www.sciencedirect.com, doi:10.1016/j.knosys.2017.09.039.

78. L. Dey, A. Mahajan and S. M. Haque, "Document Clustering for Event Identification and Trend Analysis in Market News," 2009 Seventh International Conference on Advances in Pattern Recognition, 2009, pp. 103-106, doi: 10.1109/ICAPR.2009.84.

79. Dev, Divya D., and Merlin Jebaruby. “Text Clustering Using Novel Hybrid Algorithm.” Intelligent Information and Database Systems, edited by Ngoc Thanh Nguyen et al., Springer International Publishing, 2014, pp. 11–20. Springer Link, doi:10.1007/978-3-319-05476-6_2.

80. Shakira BanuKaleel, AbdolrezaAbhari “Cluster-Discovery of Twitter Messages for Event Detection and Trending.” Journal of Computational Science, vol. 6, Jan. 2015, pp. 47–57. www.sciencedirect.com, doi:10.1016/j.jocs.2014.11.004.

81. Chua, Freddy Chong Tat, et al. “Community-Based Classification of Noun Phrases in Twitter.” Proceedings of the 21st ACM International Conference on Information and Knowledge Management, Association for Computing Machinery, 2012, pp. 1702–06. ACM Digital Library, doi:10.1145/2396761.2398501.

82. SpaCy (2020). “Linguistic Features·SpaCy Usage Documentation.” Linguistic Features, https://spacy.io/usage/linguistic-features/. Accessed 30 Dec. 2020.

83. Beumer, Lisa. “Evaluation of Text Document Clustering Using K-Means.” Theses and Dissertations, May 2020, https://dc.uwm.edu/etd/2349.

84. Cipher, Everything You Need to Know About Trend Monitoring. https://www.cipher-sys.com/blog/trend-monitoring-positioning-your-business-for-growth-and-disruption-in-competitive-markets. Accessed 14 June 2021.

85. Poppe, Olga, et al. “Event Trend Aggregation Under Rich Event Matching Semantics.” Proceedings of the 2019 International Conference on Management of Data, Association for Computing Machinery, 2019, pp. 555–72. ACM Digital Library, doi:10.1145/3299869.3319862.

86. Octoparse, Top 20 Web Crawling Tools to Scrape the Websites Quickly. http://www.octoparse.com/blog/top-20-web-crawling-tools-for-extracting-web-data. Accessed 17 June 2021.

87. Pro Web Scraping, “How to Scrape Yellow Pages with Web Scraper.” 28 Mar. 2016, http://prowebscraping.com/how-to-scrape-yellow-pages-with-web-scraper-chrome-extension/.

88. Tekyaygil, Fethi. “Extracting Feature From Raw Data in Machine Learning.” Medium, 20 Apr.2020,https://ai.plainenglish.io/extracting-feature-from-raw-data-in-machine-learning-8f5cfee7c874.

89. Picard, Richard R., and Kenneth N. Berk. “Data Splitting.” The American Statistician, vol. 44, no. 2, 1990, pp. 140–147. JSTOR, www.jstor.org/stable/2684155. Accessed 22 June 2021.

90. Farsan Rashid, In Machine Learning, What’s the Purpose of Splitting Data up into Test Sets and Training Sets? - Quora. https://www.quora.com/In-machine-learning-what-s-the-purpose-of-splitting-data-up-into-test-sets-and-training-sets. Answered on 2018, Accessed 22 June 2021.

91. Farsan Rashid, In Machine Learning, What’s the Purpose of Splitting Data up into Test Sets and Training Sets? - Quora. https://www.quora.com/In-machine-learning-what-s-the-purpose-of-splitting-data-up-into-test-sets-and-training-sets. Answered on 2018, Accessed 22 June 2021.

92. Banerjee, Sagnik. “Difference between Label Encoding and One Hot Encoding.” H2S Media, 31 Aug. 2020, https://www.how2shout.com/science/difference-between-label-encoding-and-one-hot-encoding.html.

93. Banerjee, Sagnik. “Difference between Label Encoding and One Hot Encoding.” H2S Media, 31 Aug. 2020, https://www.how2shout.com/science/difference-between-label-encoding-and-one-hot-encoding.html.

94. Satish Gunjal, Tokenization in NLP. https://kaggle.com/satishgunjal/tokenization-in-nlp. Accessed 26 June 2021.

95. Shihab Elbagir and Jing Yang (2019) Twitter Sentiment Analysis Using Natural Language Toolkit and VADER Sentiment, Proceedings of the International MultiConference of Engineers and Computer Scientists 2019 IMECS 2019, March 13-15, 2019, Hong Kong

96. Zahoor, S., and R. Rohilla. (2020) "Twitter Sentiment Analysis Using Lexical or Rule Based Approach: A Case Study." 2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), 2020, pp. 537–42. IEEE Xplore, doi:10.1109/ICRITO48877.2020.9197910.

97. Jayson DeLancey "Pros and Cons of NLTK Sentiment Analysis with VADER." CodeProject, 29 May 2020, https://www.codeproject.com/Articles/5269447/Pros-and-Cons-of-NLTK-Sentiment-Analysis-with-VADE.

98. Claire Cardie & John Wilkerson (2008) Text Annotation for Political Science Research, 5:1, 1-6, DOI: 10.1080/19331680802149590

99. "Supervised NLP and Unsupervised NLP Approach | Aisera." Aisera - AI Service Management (AISM), 23 July 2020, https://aisera.com/blog/unsupervised-and-supervised-nlp-approach/.

100. Erik F. Tjong Kim Sang and Sabine Buchholz. (2000). Introduction to the CoNLL-2000 shared task: Chunking. In Proceedings of the 2nd Workshop on Learning language in logic and the 4th conference on Computational Natural Language Learning.

101. Daniel Jurafsky, James H. Martin "Speech and Language Processing." https://web.stanford.edu/~jurafsky/slp3/. Accessed 6 June 2021.

102. Covington, M. A. (2001). A fundamental algorithm for dependency parsing. Proceedings of the 39th Annual ACM Southeast Conference, pp. 95–102.

103. Chaturvedi, Iti, et al. (2018). Distinguishing between Facts and Opinions for Sentiment Analysis: Survey and Challenges. Information Fusion, vol. 44, Nov. 2018, pp. 65–77. www.sciencedirect.com, doi:10.1016/j.inffus.2017.12.006.

104. Lori Young & Stuart Soroka (2012) Affective News: The Automated Coding of Sentiment in Political Texts, Political Communication, 29:2, 205-231, DOI: 10.1080/10584609.2012.671234

105. Medhat, Walaa, et al. (2014) "Sentiment Analysis Algorithms and Applications: A Survey." Ain Shams Engineering Journal, vol. 5, no. 4, Dec. 2014, pp. 1093–113. www.sciencedirect.com, doi:10.1016/j.asej.2014.04.011.

106. Hutto, C. J., and Gilbert, E. E. (2014). "VADER: a parsimonious rule-based model for sentiment analysis of social media rext," in Eighth International Conference onWeblogs and Social Media (ICWSM-14), Ann Arbor, MI.

107. Taboada, M, Brooke, J, Tofiloski, M. Lexicon-based methods for sentiment analysis. Computational Linguistics 2011; 37(2): 267–307.

108. @inproceedings{mikolov2018advances title={Advances in Pre-Training Distributed Word Representations}, author={Mikolov, Tomas and Grave, Edouard and Bojanowski, Piotr and Puhrsch, Christian and Joulin, Armand}, booktitle={Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)}, year={2018}}

109. Jacovi, Alon, et al. "Understanding Convolutional Neural Networks for Text Classification." ArXiv:1809.08037 [Cs], Apr. 2020. arXiv.org, http://arxiv.org/abs/1809.08037.

110. Chung J, Gulcehre C, Cho KH et al (2014) Empirical evaluation of gated recurrent neural networks on sequence modeling. Eprint Arxiv 1412:3555

111. J. Allan, R. Papka, and V. Lavrenko. On-line new event detection and tracking. In SIGIR ’98: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, August 24-28 1998, Melbourne, Australia, pages 37–45, 1998