Comparative Study of Sentiment Analysis Techniques

Example on Global Warming Tweeter Data

1. Introduction

Natural Language Processing (NLP) is an important field in modern machine learning, with the goal to understand, analyse and manipulate human language. In this analysis, I am going to apply NLP techniques to perform sentiment analysis on global warming tweets, and at the same time try to explore various types of word embedding methods. The data was contributed by Kent Cavender-Bares and it is available for download from figure-eight.com webpage. This data set consists of 6090 twitter entries which were then evaluated for belief in the existence of global warming or climate change. The possible answers were “Yes” if the tweet suggests global warming is occurring, “No” if the tweet suggests global warming is not occurring. There are 1673 tweets with missing opinions.

There are three main stages in this sentiment analysis: 1. Exploratory data analysis (EDA) and data cleaning. 2. Preparing data for machine learning. 3. Applying machine learning algorithms.

2. Possible bias in the data

This data set was chosen because it is generously available online with relatively sufficient data entries. It also comes with labelled opinions, which makes the data processing a little easier. However, one of the major bias in the data is in the opinions part. Human judgement is used to access if a specific tweet is related or non-related to global warming/climate change. As human judgement is influenced by personal perspective, feelings and prior knowledge, hence the labelled opinions are subjective. Another source of bias comes from those tweets with missing labels on existence. There are 1673 out of 6090 tweets carrying no labels, accumulating to almost 31% of the entire data set. When we take a closer look at those tweets, some of them are kind of related to global warming or climate change, suggesting that people believe global warming is occurring. It suggests the group of tweets with missing labels contains important information too. It is possible to make good use of them by using semi-supervised learning method in the later part of the analysis.

3. EDA and data cleaning

Exploratory data analysis (EDA) is an important approach to analysing data sets in order to summarize their main characteristics. EDA is usually the first step when we look at the data to find out more information. Visualization or graphs are usually essential in supporting the interesting or useful information we have obtained from the data. The following steps have been performed in the process of exploratory data analysis and data cleaning.

1. Checked the dimensions of the data set.

The data set consists of 6090 data entries and 3 variables: tweet, existence, existence.confidence. As we are not going to use the last column, we will drop it.

	tweet	existence	existence.confidence
0	Global warming report urges governments to act...	Yes	1.0000
1	Fighting poverty and global warming in Africa ...	Yes	1.0000
2	Carbon offsets: How a Vatican forest failed to...	Yes	0.8786
3	Carbon offsets: How a Vatican forest failed to...	Yes	1.0000
4	URUGUAY: Tools Needed for Those Most Vulnerabl...	Yes	0.8087

Snapshot of the global warming tweets

2. Check duplicated entries and remove them

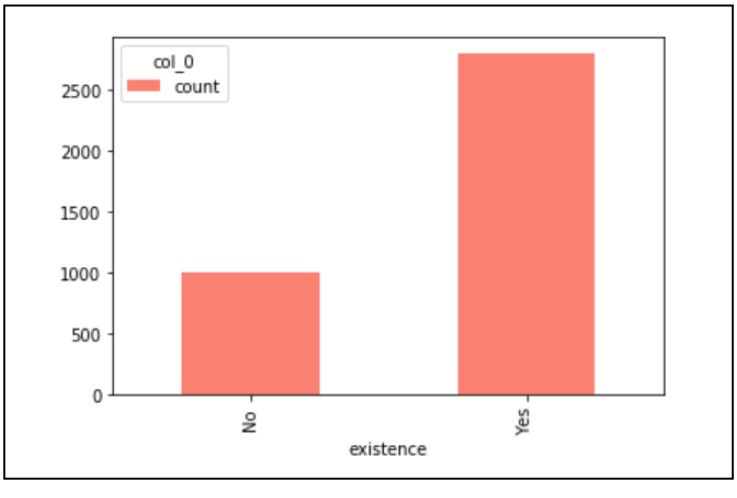
   It is common to see duplicated entries in twitter data due to extraction issues or human errors. We should keep only the first entry and remove all the rest, as duplicated entries could lead to overfitting if found in huge numbers. After removing the duplicates, we have 5471 entries left.

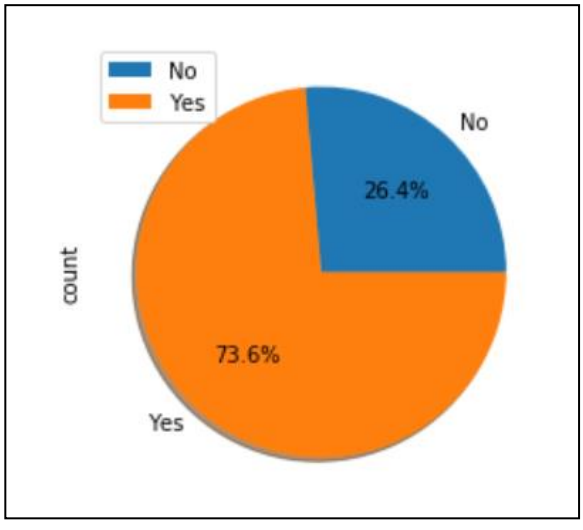
3. Check for missing values and perform imputation

   For the first column “tweet”, there are no missing entries. However, for the second column “existence”, we have 1673 tweets with missing opinions. We group these tweets and label them as “Missing”. In the first part of the analysis only tweets with “Yes” or “No” opinions are used to build the classifier. At this stage there are 3798 observations in the data set.

4. Summarize the count and percentage of each opinion

existence	col_0 count
No	1002
Yes	2796





Percentage of class distribution of tweets

From the count and percentage summary, we notice there are more (73.6%) tweets of "Yes" opinion about global warming existing, 26.4% are opinions on global warming not existing. Based on the information, this data set consists of imbalanced classes, hence we must be careful in selecting the right performance metrics for comparing model performances. In the case of imbalanced data, precision, recall and F1 score will be more appropriate compared to accuracy.

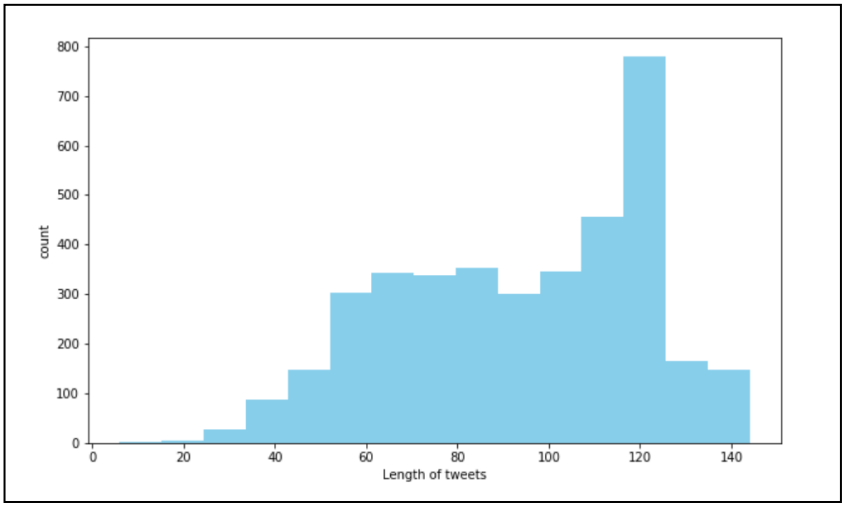
5. Clean up the tweets

   As tweets normally come with symbols like @ #, punctuations and website links, we need to remove these symbols and links because they have no use in the sentiment analysis later using machine learning algorithms.

6. Explore relationship between opinion and tweet length

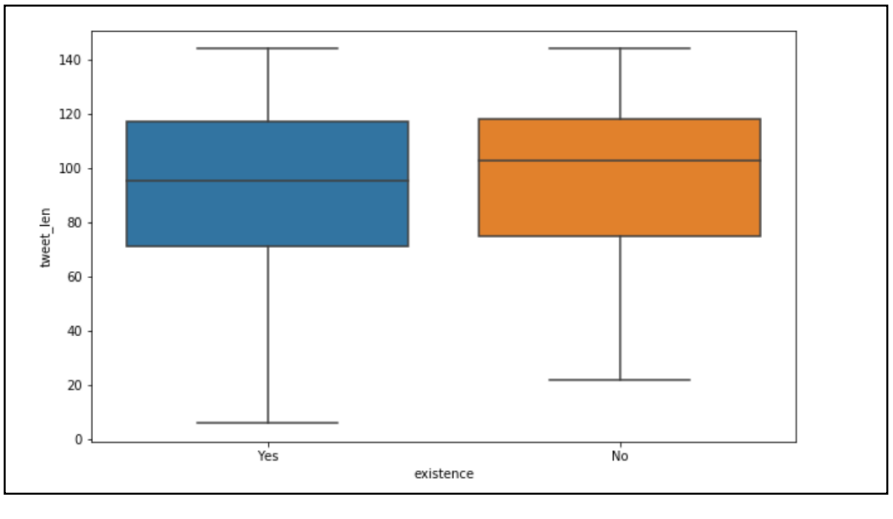
   Does the length of each tweet tell us anything about the writer's opinion on global warming? To find the answer we will look into the details of tweet length. Graph 3 below shows the distribution of the tweets' length. From the distribution, we see most of the tweets are in the

range of 20 to 140 characters in length which is considered a relatively large range. The peak was at 120 characters and accounts for about 21%, which is computed by $\frac{800}{3798}$ * 100 = 21%.



Distribution of tweet length

To explore the relationship between tweet lengths and different opinions, boxplot has been used to illustrate the distribution of tweet lengths in various opinion classes, as shown in the graph below. Overall the tweet lengths in different opinion classes do not differ much. The medium lengths of both “Yes” and “No” classes are about 100 characters. This could be explained by the length limit (140 characters) set by Twitter back then.



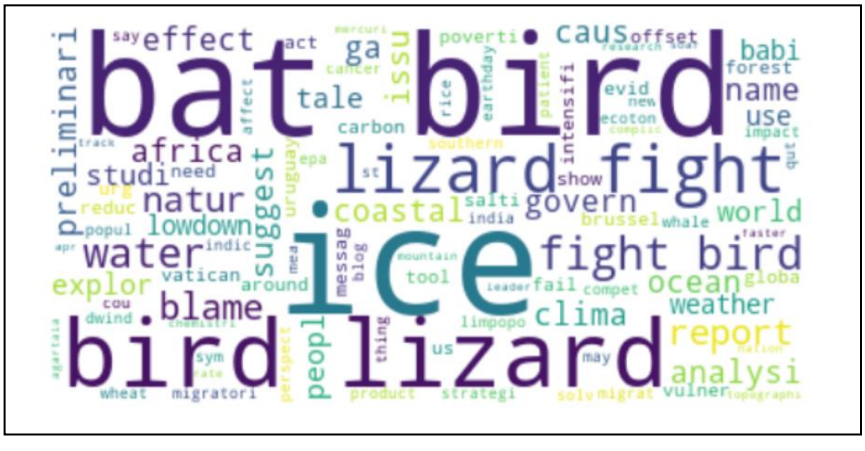
Distribution of tweet length in different opinion classes

7. Text mining on the tweets

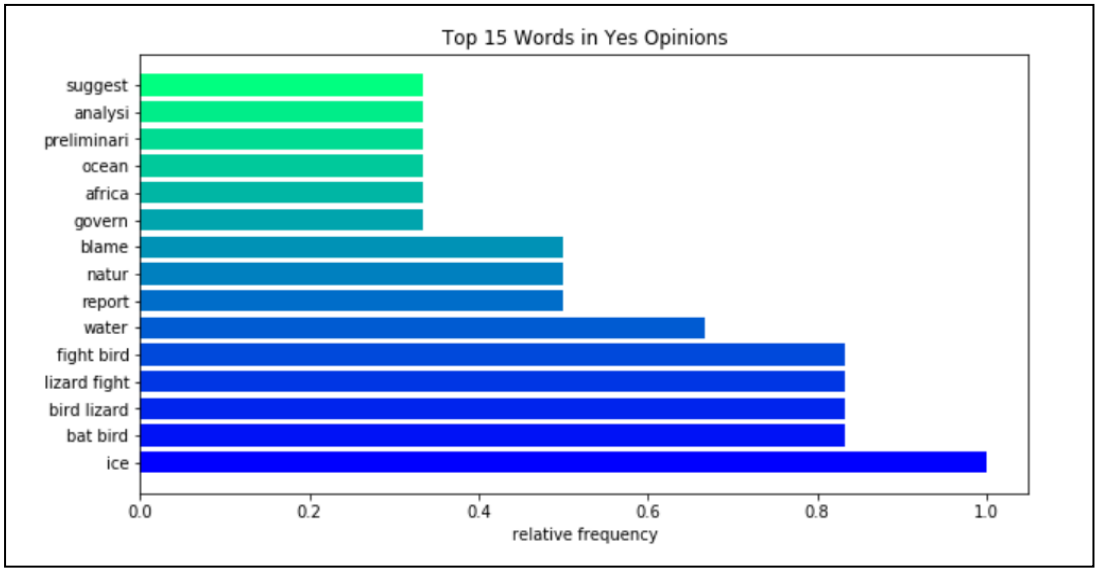
   The data preparation includes text mining techniques such as tokenizing the sentences, stemming and lemmatization, removing stop-words.

8. Top words in both classes

   In this step, we filter data with “Yes” and “No” opinions to create two independent sets of data, followed by applying the technique of Word Cloud to display the top words with highest occurrence rate in each class. As “global, warm, climate, change” appear equally frequently in both classes, it will be better to filter them out for better comparison on other key indicating words. Here below shows the results.

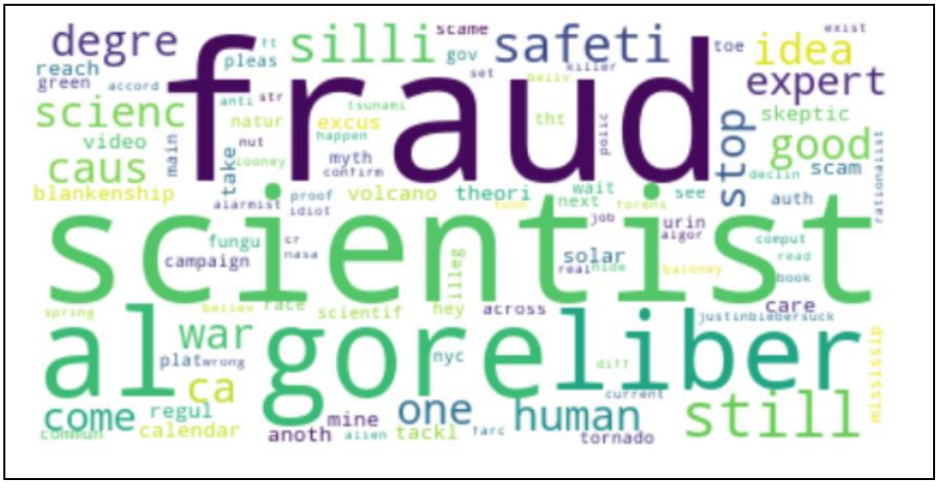


Word Cloud of top words in “Yes” opinion

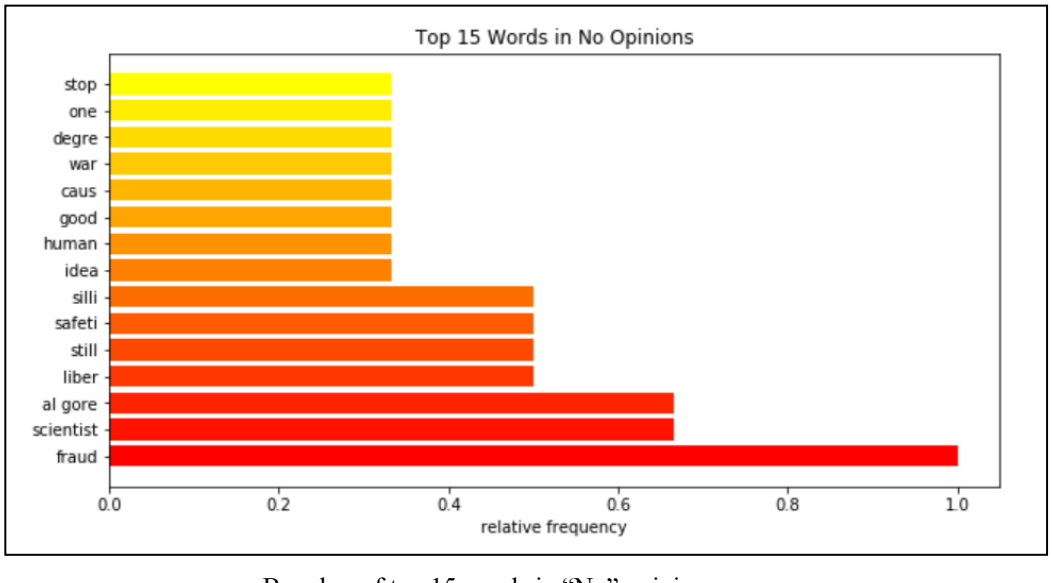


Bar char of top 15 words in “Yes” opinion

Combining information from the two graphs, we have a better idea of the top 15 most frequent words in “Yes” data set. The word “ice” has the highest occurrence rate and the reason is rather obvious – global warming and climate change results in icebergs melting. The 2nd – 5th high frequency words are all about animals – bat, bird and lizard. You might wonder why these animal names have such high frequency rates of appearing in global warming related tweets. After some research, I have become aware of the fact that global warming and climate changes cause certain species of bats, birds and lizards to be endangered. Other top words such as water, nature, ocean are important indicators of the changes brought by global warming.



Word Cloud of top words in “No” opinion



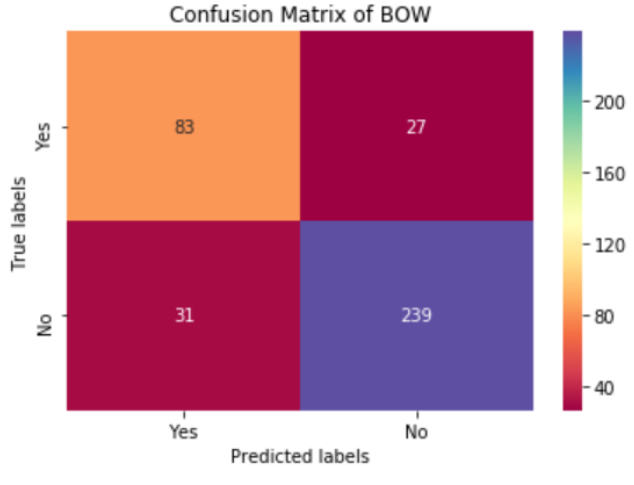
Bar char of top 15 words in “No” opinion

Similarly the most prominent word in “No” opinion data is “fraud”, which reflects how those who do not believe in global warming think about climate change.

4. Machine Learning

4.1 Bag of Words

The first algorithm used is Bag of Words (BOW) method to extract text features together with Logistic Regression. In this model, we use the tokenized words for each observation and find out the frequency of each token. Out of 3798 tweets with ‘Yes’ or ‘No’ opinions, 90% is used for model training and the remaining 10% for testing. With BOW we have the prediction result as shown below. Since it is an imbalanced data, we will prefer to use precision, recall and F1 score as the performance metrics, instead of accuracy. We have about 85% Precision, Recall and F1 score in overall, by taking the weighted averages from both ‘Yes’ and ‘No’ groups.

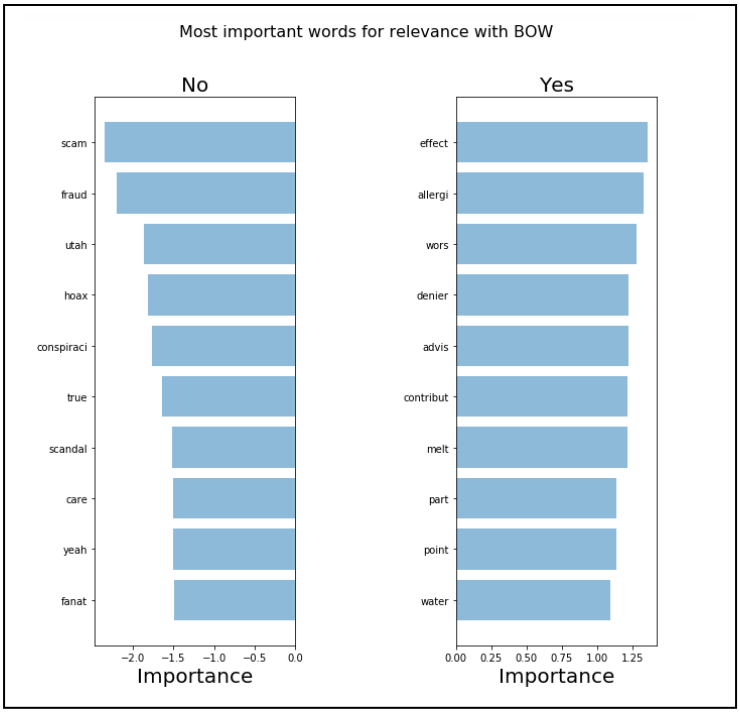


	BOW
Precision	0.849
Recall	0.847
F1	0.848
Accuracy	0.847

Graph 9. Bag of Words model performance

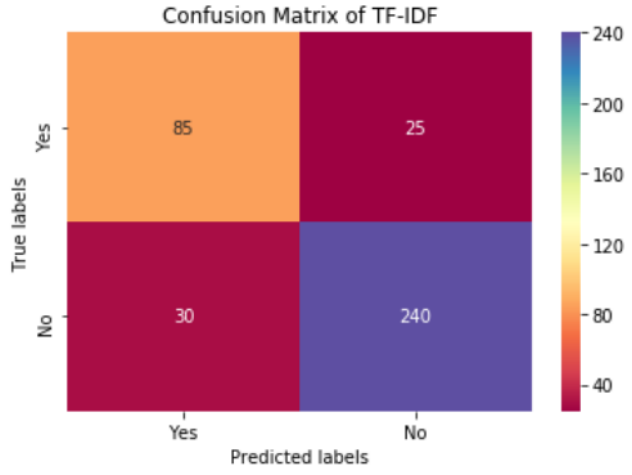
Although the results look fairly decent, we will take a deeper look at the important words picked up by the algorithm to examine the reliability of BOW model here. The below pictures show the top 10 most important words picked by the BOW algorithm for both ‘Yes’ and ‘No’ groups. By

scanning through the words, most of the important words in both groups are relevant to what we are expecting, except a few irrelevant words like ‘yeah’, ‘fanat’ in ‘No’ group and ‘point’, ‘part’ in ‘Yes’ group. Overall the BOW algorithm has done a fairly good job in picking up relevant indicators for respective opinions.



4.1 Term Frequency – Inverse Document Frequency (TF-IDF)

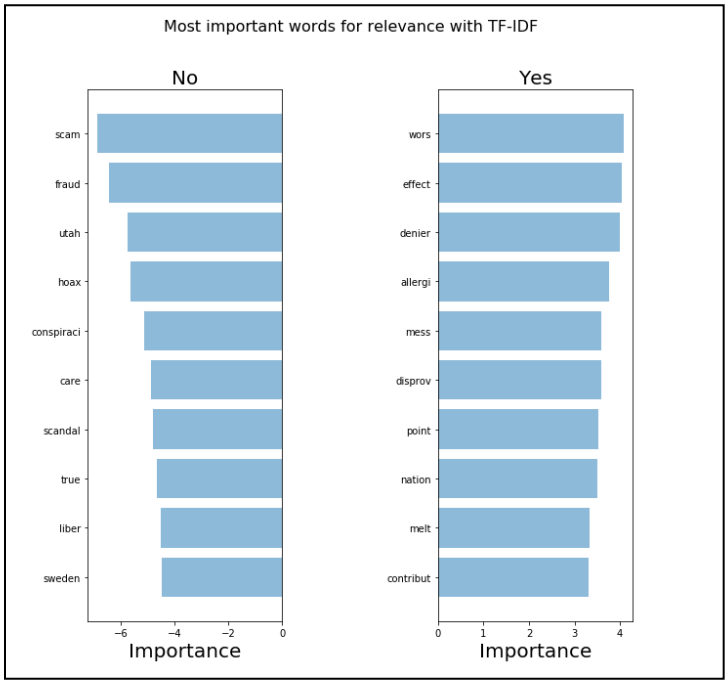
TF-IDF is another method of extracting features from text. It is a scoring measure reflecting how relevant or meaningful a word is to a document in a corpus. If a frequent word in a document (TF) also appear often in other documents, it implies that this word is just a frequent word in the whole corpus but not relevant or meaningful with respect to the specific document (IDF). As many studies suggest, TF-IDF algorithm is able to pick up more relevant words than BOW, hence we will try TF-IDF too and compare the performances. The results of TF-IDF is summarized as follows in Graph 10. The results of TF-IDF method turns out to be slightly improved.



	BOW	TF-IDF
Precision	**0.849**	**0.857**
Recall	**0.847**	**0.855**
F1	**0.848**	**0.856**
Accuracy	**0.847**	**0.855**

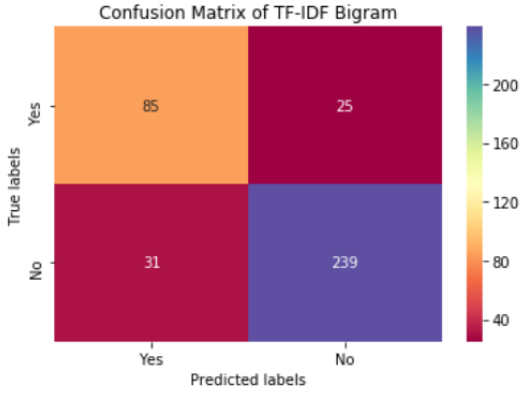
Graph 10. TF-IDF model performance

Although we do not see much improvement in the performance, we are still curious to find out if TF-IDF algorithm picks up more relevant words as compared to BOW. Here are the top 10 important words in TF-IDF model. We notice the irrelevant words ‘yeah’, ‘fanat’ from ‘No’ group in BOW model have been replaced by two new words ‘liber’, ‘sweden’. For ‘Yes’ group, ‘advis’, ‘part’, ‘water’ have been replaced by ‘mess’, ‘disprov’, ‘nation’. The new words seem a little more relevant to what we are predicting. That possibly explains the 0.01% improvement in the results.



4.3 Bigram with TF-IDF

TF-IDF Bigram has also been used to explore if any important bigram vocabulary exists in the tweets, that helps picking up important words for prediction. The results are shown below. We do not see improvement in performance and the TD-IDF Bigram algorithm only picks up a few important bigram vocabularies, e.g. ‘stop global’, ‘chang climat’.



	BOW	TF-IDF	TF-IDF BIGRAM
Precision	**0.849**	**0.857**	**0.855**
Recall	**0.847**	**0.855**	**0.853**
F1	**0.848**	**0.856**	**0.854**
Accuracy	**0.847**	**0.855**	**0.853**

4.4 Semi-supervised Learning

At the beginning we left out a group of tweets with no labels and group them as ‘Missing’. We would like to make good use of them too. Here we apply the TF-IDF model trained with ‘Yes’ and ‘No’ groups to predict the labels of ‘Missing’ group. After that these two groups of data are combined to build a new model and make predictions. The result shows improvement of about 0.4% in Precision, Recall and F1 score.

	BOW	TF-IDF	TF-IDF BIGRAM	Semi Supervised
Precision	**0.849**	**0.857**	**0.855**	**0.888**
Recall	**0.847**	**0.855**	**0.853**	**0.889**
F1	**0.848**	**0.856**	**0.854**	**0.889**
Accuracy	**0.847**	**0.855**	**0.853**	**0.889**

When we extract the most important words from the model, we notice more relevant words like ‘environment’, ‘issu’, ‘climat’, ‘post’ are picked up by the model for prediction, and that explains the improvement in the prediction results in test data.

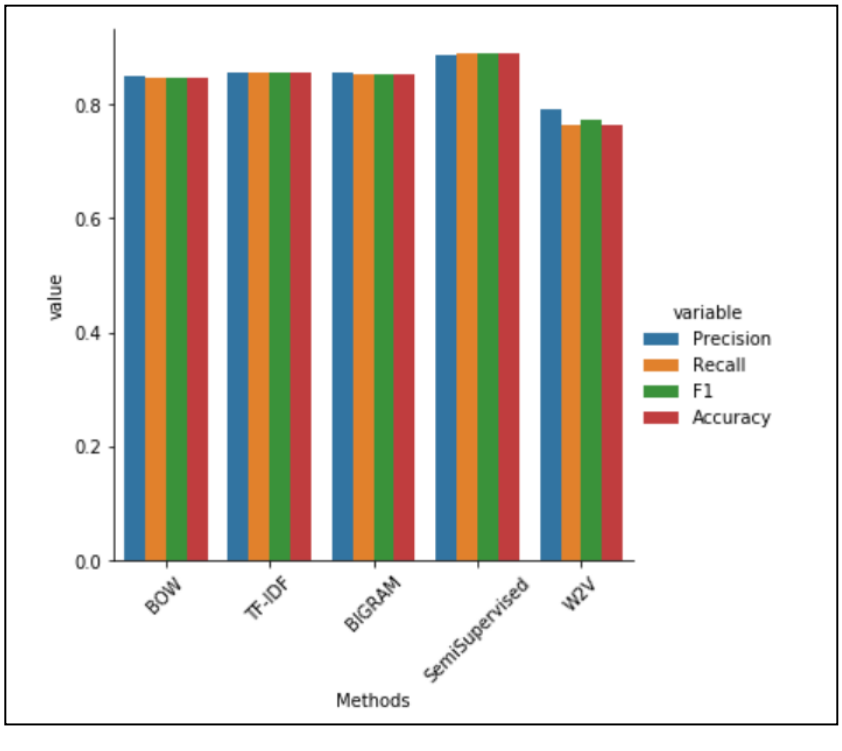
4.5 Word2Vec

In the last part of the analysis we want to explore more advanced deep learning algorithm on sentiment analysis. Word2Vec is a good choice because it is a method to efficiently create word embeddings though n-dimensional vector representation. The usefulness of Word2Vec is to group the vectors of similar words in the vectorspace. Given enough data, Word2Vec can make fairly accurate guesses about the meaning of a specific word based on the context and other words appeared around it. The results are shown below. However, it does not perform as good as other models, likely due to relatively smaller data set.

	BOW	TF-IDF	TF-IDF BIGRAM	Semi Supervised	Pretrained W2V
Precision	**0.849**	**0.857**	**0.855**	**0.888**	**0.792**
Recall	**0.847**	**0.855**	**0.853**	**0.889**	**0.763**
F1	**0.848**	**0.856**	**0.854**	**0.889**	**0.773**
Accuracy	**0.847**	**0.855**	**0.853**	**0.889**	**0.763**

5. Conclusion and Future Works

The histogram below shows the performance comparison of all the models we have performed. The semi-supervised learning model has the best performance among all. However there are a few suggestions regarding how to improve the reliability and accuracy of the models. As semi-supervised learning usually produces good result when the size of unlabelled data is much larger compared to that of labelled data, we could add more unlabelled tweets to the model to verify if that produces good results too. To boost the performance of Word2Vec we could possibly apply XGBoost to see if that helps bring up the result.



References:

Ameisen, E. (Jan 24, 2018). How to solve 90% of NLP problems: a step-by-step guide. Medium. Retrieved from https://blog.insightdatascience.com/how-to-solve-90-of-nlp-problems-a-step-by-step-guide-fda605278e4e