Twitter Sentiment Analysis

Group 23

Kritartha Ghosh (A0191466X)
Pankaj Bhootra (A0144919W)
Ronald Lim (A0147890U)

Goal

- Gaining Sentiment Analysis insights on specific topics or hashtags on Twitter through the use of Spark and Machine Learning on a large dataset of Tweets.

Dataset Collection

- Source: **https://archive.org/details/archiveteam-twitter-stream-2018-10**
- **31,130,257** Tweets (~3.01 GB)
- All english tweets during the month of October, 2018
- No specific topic or further filtering when collecting Tweets
- The aim is to train a sentiment classifier model for english tweets of any topic

Tools & Technologies

- Azure Databricks
- Azure SQL Database Warehouse
- Apache Spark with Scala
- Spark MLLib (Machine Learning)
- Spark SQL (Data Management)
- Tableau (Data Visualization)

Tasks Pipeline

Step 1: Preprocessing of Raw Tweets

Step 2: Applying Hashing TF, IDF and Standard Scaling

Step 3: Training Machine Learning models

Step 4: Model Performance Comparison

Step 5: End-to-End Inferencing, Storing Results and Data Visualization

Data Preprocessing

Preprocessing is an important step to get better predictions.

Major steps are:

- Removing the usernames and hyperlinks
- Punctuations are also removed
- Camel case words are broken down into individual words
- Tweets are converted to lowercase
- Common contractions are expanded
- Stopwords and extra white spaces are removed
- Empty tweets are removed

Example of Data Preprocessing

Actual tweet:

RT @OriginalFunko: The Great War is here! #ForTheThrone #GameOfThrones #GoT @GameOfThrones https://t.co/kk5AzlqkJj

After preprocessing:

great war throne game thrones

TF-IDF

Term frequency-inverse document frequency (TF-IDF) is a feature vectorization method widely used in text mining to reflect the importance of a term to a document in the corpus.

Term Frequency (TF)

Term frequency is the number of times that a term appears in a document.

Inverse Document Frequency (IDF)

Document frequency is the number of documents that contain a particular term. Inverse document frequency is a numerical measure of how much information a term provides.

Standard Scaler

The StandardScaler standardizes features by scaling them to unit variance and/or removing the mean using column summary statistics on the samples in the training set.

This is a common preprocessing step which can improve model performance.

Machine Learning

Twitter Sentiment Analysis

Each tweet can have 10 different sentiments associated with it

MultiLabel Classification

- Anger
- Anticipation
- Disgust
- Fear
- Joy
- Sadness
- Surprise
- Trust
- Negative
- Positive

Machine Learning

Model Training Using Spark MLLib

The following models were experimented with:

- Deep Neural Network
- Logistic Regression
- Random Forest
- Naive Bayes
- Decision Tree

Multi-label Classification

The multiclass classification algorithms can be extended to work for multilabel classification by considering the classification problem as a set of binary outcomes.

Does this tweet belong to this class? Yes or No

This approach is similar to the implementation of OneVsRest in sklearn for multilabel classification.

We train 10 classifiers, with each classifier responsible for predicting if each tweet belongs to a class, ending up with 10 different labels.

Model Performance Comparison (Accuracy)

Sentiments	Logistic Regression	Naive Bayes	Decision Tree	Random Forest	Deep Neural Network
Anger	93.5 %	81.1 %	77.4 %	71.4 %	**98.6 %**
Anticipation	94.2 %	81.6 %	69.3 %	60.0 %	**98.1 %**
Disgust	94.6 %	80.8 %	84.7 %	79.9 %	**98.3 %**
Fear	93.0 %	82.1 %	76.2 %	71.5 %	**97.6 %**
Joy	94.8 %	81.9 %	72.5 %	63.5 %	**98.5 %**
Sadness	93.8 %	81.3 %	78.7 %	73.4 %	**98.4 %**
Surprise	95.5 %	82.0 %	84.9 %	78.9 %	**98.9 %**
Trust	92.8 %	82.2 %	63.0 %	57.4 %	**98.1 %**
Negative	90.4 %	81.5 %	59.0 %	58.8 %	**96.0 %**
Positive	92.3 %	83.4 %	65.0 %	65.0 %	**98.1 %**

Model Performance Comparison (Avg. Accuracy)

Metric	Logistic Regression	Naive Bayes	Decision Tree	Random Forest	Deep Neural Network
Average Accuracy	93.5 %	81.8 %	73.1 %	68.0 %	**98.1 %**

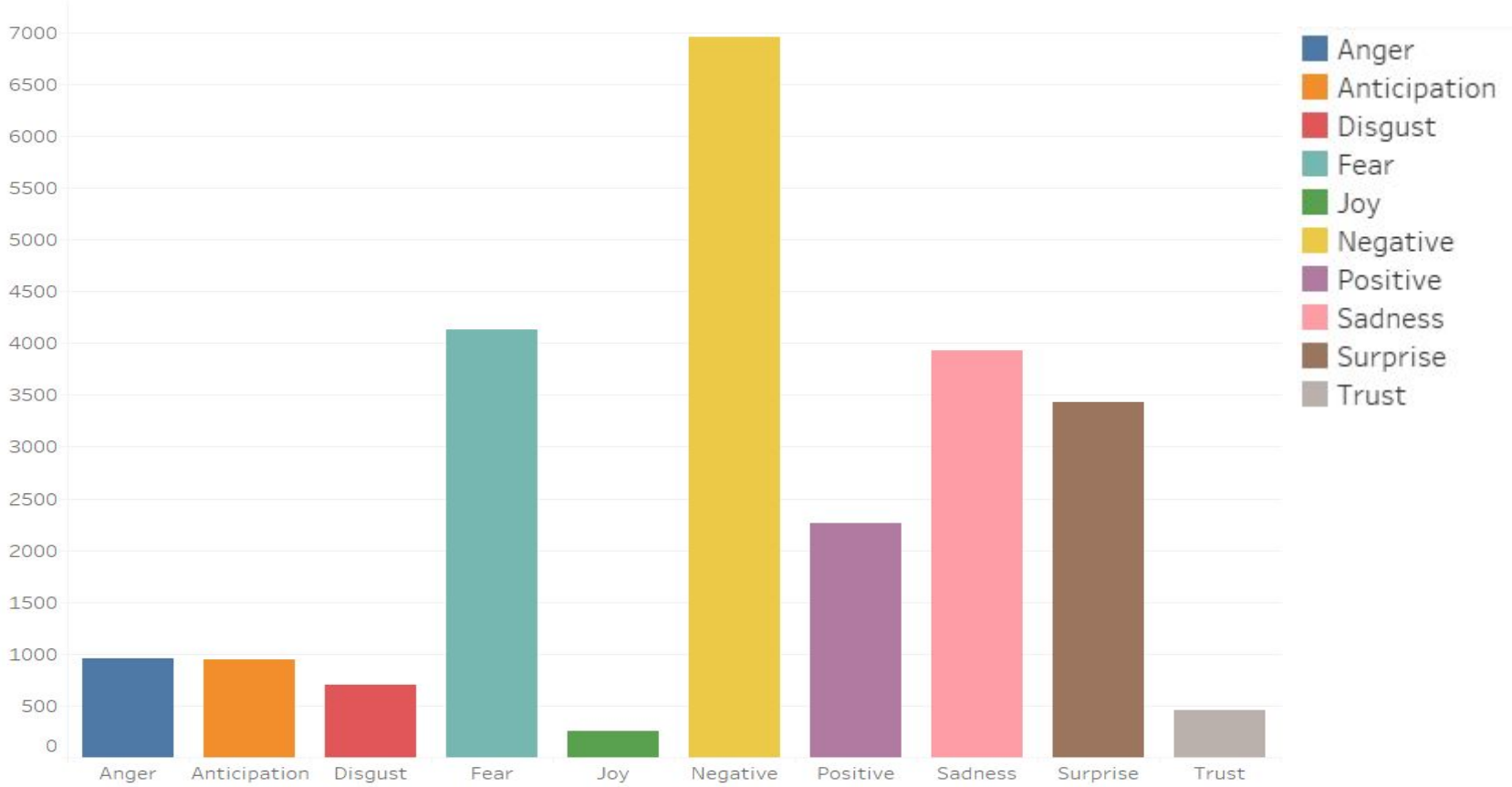
Setting up Azure SQL Database

- To store results for Sentiment Analysis of Tweets for a given hashtag.
- Azure SQL Database can be easily set up and used with Spark SQL, which provides OLTP and OLAP APIs for the same.
- The end-to-end process yields two sets of results: the number of tweets for each sentiment (anger, anticipation, etc.) and the top 150 most frequent words occurring in the tweets.
- Each result is stored in separate tables in the SQL database.
- Eg. for #GoT tweets, the results are stored in tables **got_preds** (for sentiments found) and **got_wordcounts** (for most frequent words found).

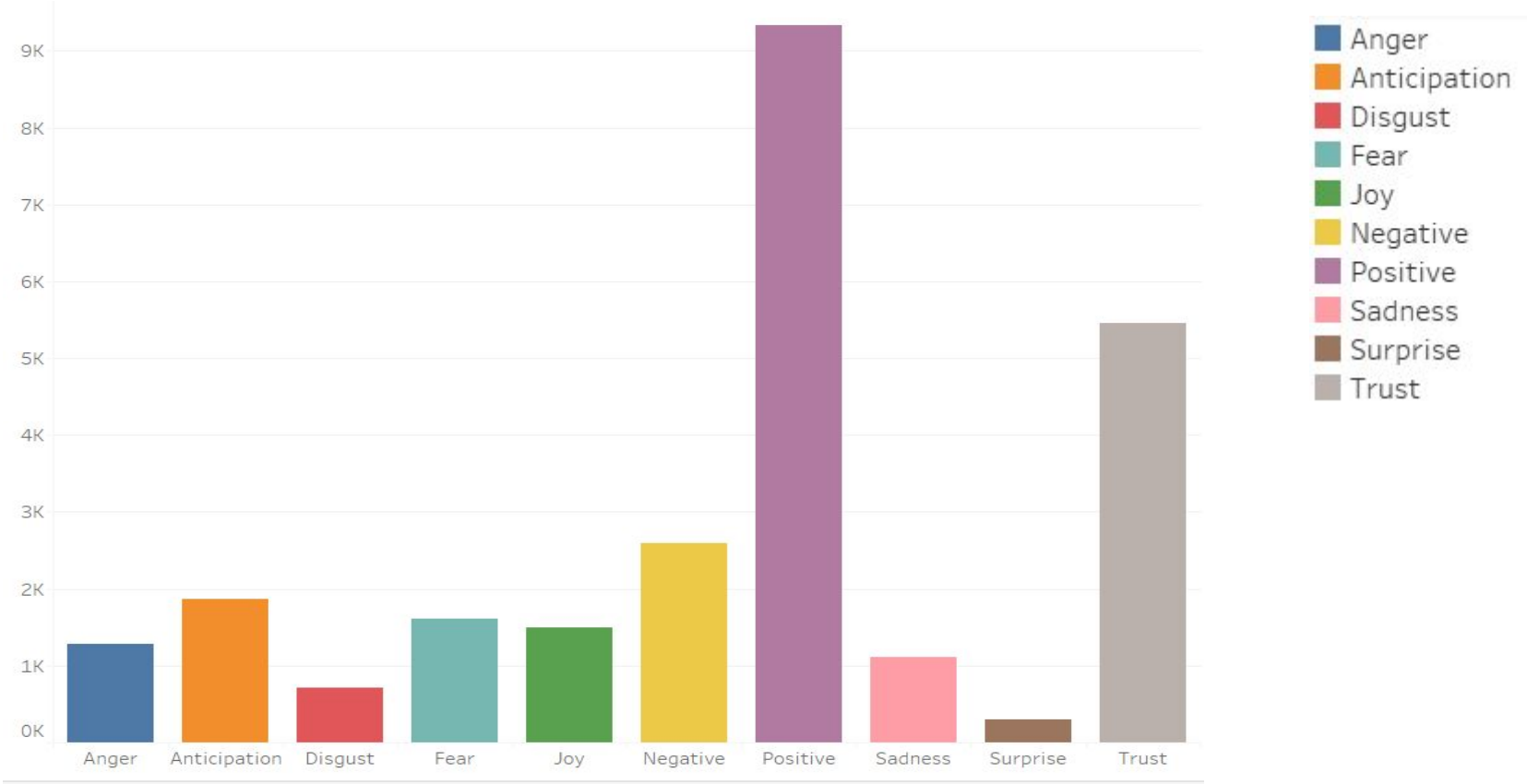
Data Visualization using Tableau

- With the results stored in an Azure SQL database, the table can be queried for data visualization.
- Tableau is used as it can easily connect to remote SQL databases and generate visualizations for the results of Sentiment Analysis.
- Eg. for #GoT, the tables **got_preds** (for sentiments found) and **got_wordcounts** (for most frequent words found) are accessed remotely from Tableau and then various Tableau features are used to create bar charts and word clouds respectively.

#brexit Tweets for each Sentiment

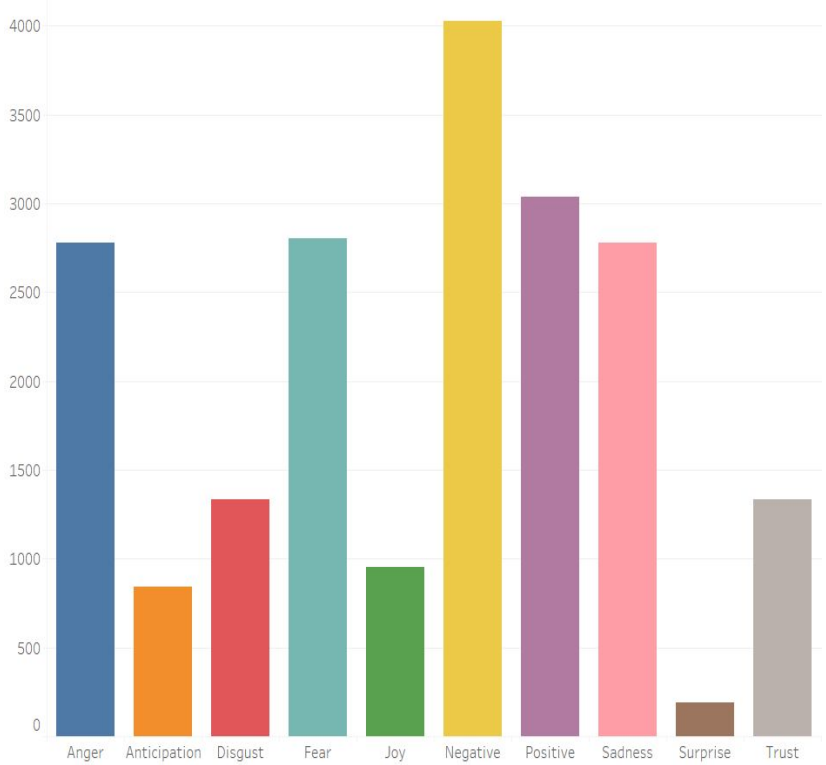


#GoT Tweets for each Sentiment

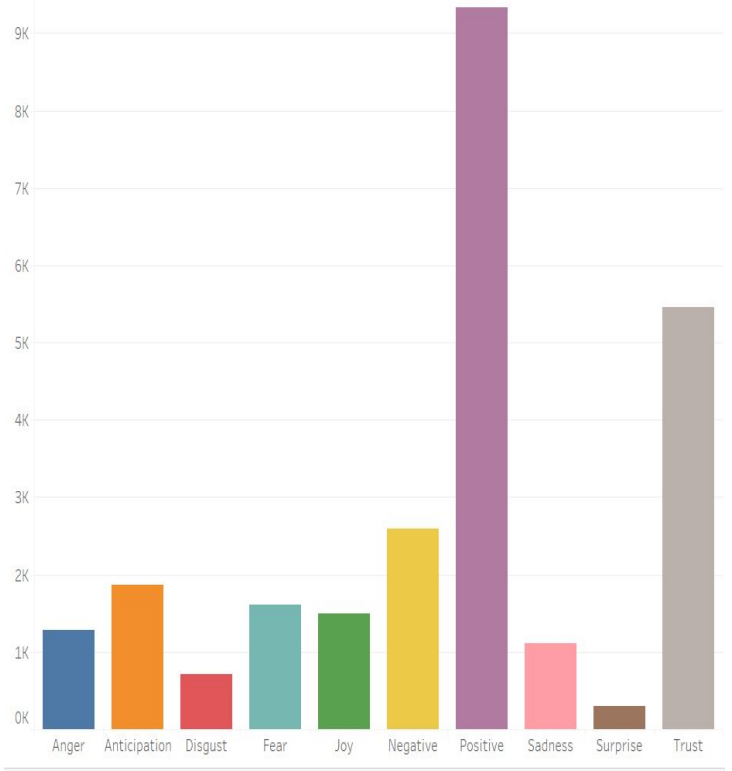


How sentiments for #GoT Tweets changed...

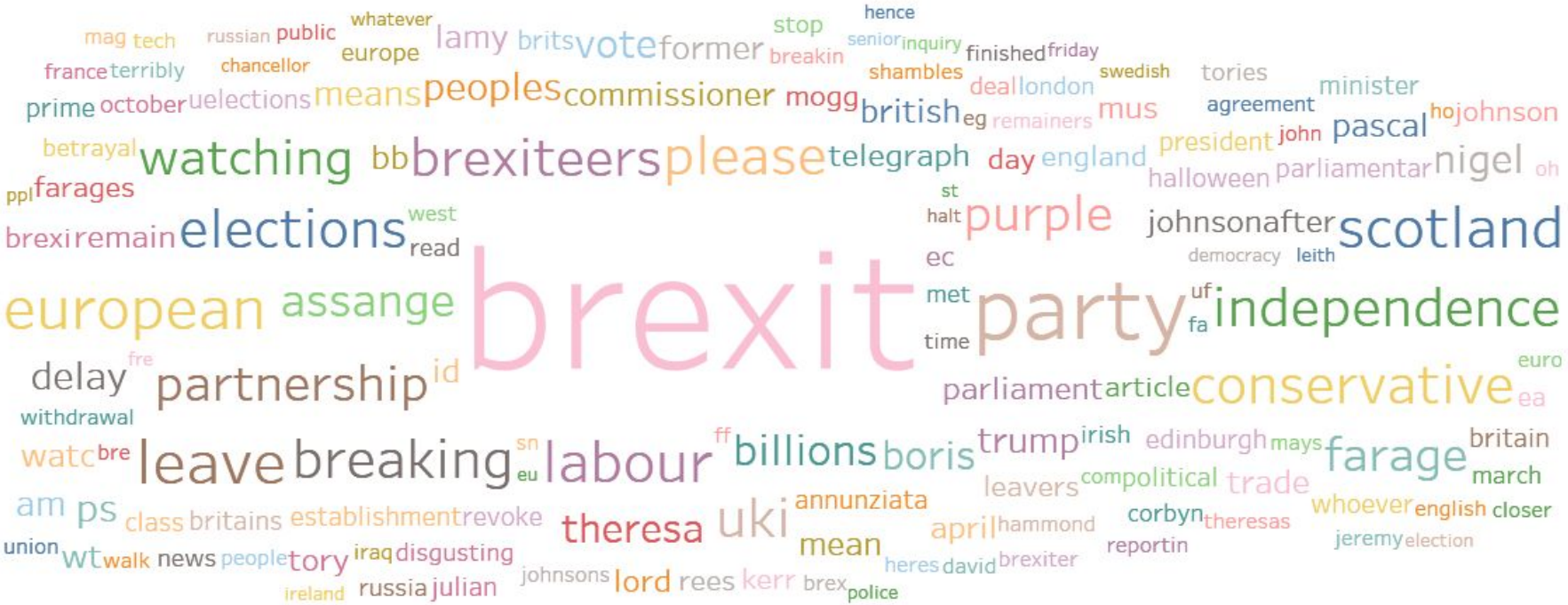
Before Season 8 premiere...



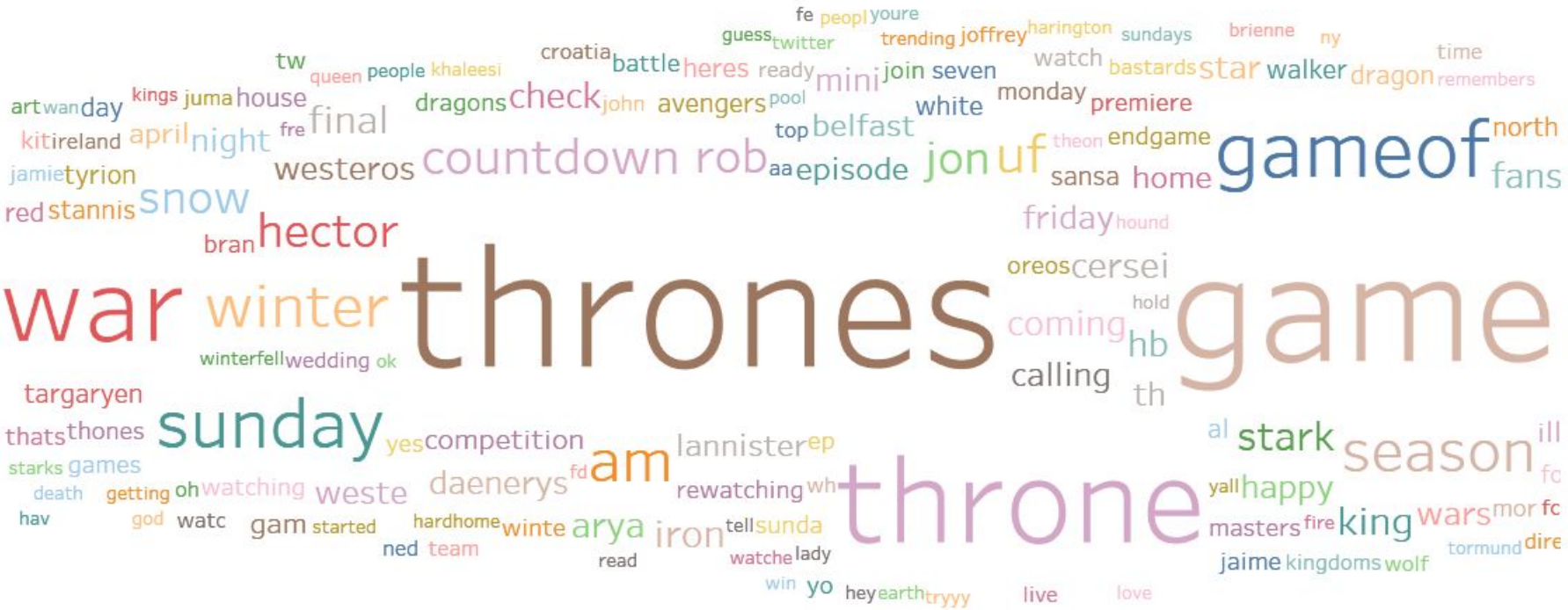
After Season 8 premiere...



Word Cloud for #brexit Tweets



Word Cloud for #GoT Tweets



Future Work

- Improving model performance by using more effective vectorizing techniques of words, such as Word2vec
- Using tools such as Kafka or Spark Streaming for ‘near’ real-time Sentiment Analysis of Twitter feeds