Abstract |

Insights gained from sentiment analysis (SA) have allowed businesses and users an opportunity to gain a deeper understanding of how communities perceive products, brands, topics, and/or services. Reddit is an American social news aggregation, web content rating, and discussion website and as of Jan 2020 according to Alexa Internet, it ranks as the fifth most visited website in the Canada and 18th in the world. For users, one advantage to using reddit is that all topics are aggregated in a sub community (known as a subreddit) where individuals may post and comment on topics of specific interest to them. By employing a generalized sentiment analysis script, we aim to mine the text to develop a generalised model that will analyse submissions and comments so that we can gain insights on what is currently trending in any subreddit in real-time. Currently there are 2 popular approaches for analyzing text, the first utilizes a bag of words method and applies TF-IDF while the second employs a shallow 2-layer neural network that are trained to reconstruct the contexts of words known as words2vec. For the scope of this project, our methodology is as follows; 1) Utilise Reddit’s api to scrape subreddit submission titles and comments 2) pre-process (stop words, symbols, apostrophes, lemmatize ) and tokenize text 3) upload the data to a MongoDB database 4) Apply TF-IDF, words2vec, and sentiment analysis models on the pre-processed text 5) Deploy streaming version to get live updates of subreddits.

Introduction |

With the dramatic shift and prevalence of social media in modern society, text analytics can be a powerful tool for companies to take strategic action based on valuable insights on common themes and trends found on the internet. To make text analytics the most efficient, organisations can use text analytics software, leveraging machine learning and natural language processing algorithms to find meaning in enormous amounts of text. Reddit is an American social news aggregation, web content rating, and discussion website and as of Jan 2020 according to Alexa Internet, it ranks as the fifth most visited website in the Canada and 18th in the world. For users, one advantage to using reddit is that all topics are aggregated in a sub community (known as a subreddit) where individuals may post and comment on topics of specific interest to them. Therefore, there is a wealth of commentary, opinions, and knowledge that can be mined from the submissions and comments of reddit that can have the potential to provide valuable insight to any company and/or topic of interest. For the scope of this project, our methodology is as follows; 1) Utilise Reddit’s api to scrape subreddit submission titles and comments 2) pre-process (stop words, symbols, apostrophes, lemmatize ) and tokenize text 3) upload the data to a MongoDB database 4) Apply TF-IDF, words2vec, and sentiment analysis models on the pre-processed text 5) Deploy streaming version to get live updates of subreddits.

Literature Review

1) Řehůřek, R. (2019, November 1). Gensim Topic Modelling for Humans Core Concepts. Retrieved from https://radimrehurek.com/gensim/index.html

This webpage documents the free python package known as gensim and explores key concepts about its usage and required understanding of terms such as documents (text), corpus (collection of documents), vector (mathematical representation of corpus, and models (algorithm employed to transform vectors from one representation to another. It also provides a walkthrough and example for users to get started using genism as well as providing api references.

2) Jurafsky, D. and Martin, J.H., “Vector Semantics and Embeddings,” in Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition, 3rd ed. Stanford University, UK: Online, 2019, pp. 94–122.

The chapter “vector semantics and embeddings” explains 5 main topics which include (1) the various methods to represent words, (2) ways to measure similarity, (3) TF-IDF model, (4) Word2vec, and (5) the properties of embeddings. Additionally, the concept of information retrieval is introduced which is used to find documents in a collection that best matches a query from a term-document matrix. For its application, TF-IDF is a useful model to give weight to terms in a given vector, and it is dependent on the term frequency and its frequency of appearance in documents, such that words that are useful for describing documents from rest of collection are given more weight, while those that occur frequently are less. Therefore, TF-IDF is good for weighting co-occurrence matrices for IR.

In addition to representing words in a term-document matrix, word-word matrices enable the identification of n-grams around a target word and by using cosine similarity it is possible to determine if a word is similar to another pair based on its frequency. Furthermore, to identify collocations it is practical to use pointwise mutual information to calculate the probability of 2 words (x and y) occurring compared to what we would expect if they were to appear independently.

Alternatively, rather than asking how often each word occurs near a target word as with TF-IDF. It is possible to ask using the Word2vec model, how likely a word is to appear a target word. This model is a binary classification trained on a logistic regression classifier which uses and algorithm known as skip-gram. Skip-gram is a shallow neural network that treats the target word and each neighboring word as a positive example while randomly sampling other words to get negative samples. This method uses a logistic regression model to train classifier to distinguish the 2 cases and uses the regression weights as embeddings.

3) Hutto, C.J. & Gilbert, Eric. (2015). VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text. Proceedings of the 8th International Conference on Weblogs and Social Media, ICWSM 2014.

Highlights the development, evaluation, and validation of VADER (for Valence Aware Dictionary for sEntiment Reasoning). Sentiment analysis was previously heavily reliant on pre-existing manually constructed lexicons where each lexical feature was labelled manually according to their semantic orientation. The article identifies 3 sentiment oriented (polarity-based) lexicons, which include LIWC, General Inquirer (GI), and Hu & Liu and notes that the common disadvantage for using them is that they are unable to discern sentiment intensity of words and an inability to parse sentiment-bearing lexical items such as slang or emoticons from social text. The authors construct and empirically validate a gold standard list of lexical features (along with their associated sentiment intensity measures) which are specifically attuned to sentiment in microblog-like contexts.

TextMiner (2014, January 17). Text Mining Online: NLP. Retrieved from https://textminingonline.com/

TextMiner provides an in depth walk through using python’s famous natural language processing package known as NLTK. The website contains a series of articles that introduce key concepts in the preprocessing of text such as word tokenization, parts-of-speech-tagging, and stemming and lemmatization all within the python environment.

1) Corpus is a collection of document objects, the corpus in this project will include a collection of comments.

2) Tokenize the document objects

3) Keep words that appear more than once

4) Assign IDs to unique words in the corpus

5) Represent each document as a vector of features (known as a dense vector) and remove any elements within the vector that consist of 0.0, which would indicate a word has zero occurrences. This new vector is known as a sparse vector or bag-of-words vector. Dimension of the dense vector is equal to the number of unique words, the dimension of a bag-of-words vector is the number of unique words found in document (comment).

6)Convert entire corpus as a bag-of-words vector

7) Apply TRANSAFORMATION, the tf-idf model transforms vectors from the bag-of-words representation to a vector space where the frequency counts are weighted according to the relative rarity of each word in the corpus

8) We then index the transformed corpus and can then process our query by tokenizing and vectorizing it. The model will then return which documents are most similar (similarity score)

Dataset

The dataset used is an on-demand web scrape of a subreddit of particular interest to the user. Using PRAW: The Python Reddit API Wrapper, we are able to pull submission titles and all comments of “n” hottest submissions from the front page and store the data in a mongoDB database (schema-less NoSQL document database). Within the database, we can have multiple collections each representing a submission within the subreddit and all the comments stored as documents specific to that submission. Furthermore, we have an additional collection called “[subreddit name]\_overview” which includes information about all submissions we scraped which includes their title, score, id, url, number of comments, and content. Lastly, we have 3 collections called dictionary, dictionary\_raw, and dictionary\_token that are similar in that they contain an aggregation of comments from all submissions and includes information such as the message, submission title, and timestamp. However they differ in that dictionary\_raw contains text that has not been preprocessed, while dictionary has text that has gone through the preprocessing step (refer to approaches section for details on preprocessing) but is not tokenized, and dictionary\_token have both been tokenized and preprocessed. For the scope of this project, we scrape 2 subreddits as an example known as r/leagueoflegends and r/politics and take the top 20 (due to computing and time constraints) hottest submissions and its comments.