2019-1002 IST 736 Text Mining

Homework Assignment 1 (week 1)

Ryan Timbrook
NetID: RTIMBROO
Course: IST 736 Text Mining
Term: Fall, 2019
Topic: Evaluation of Sentiment Classification Tools

Homework Assignment 1 (week 1)

Table of Contents

1 Introduction

Evaluate and report on which sentiment classification tools available could be used to perform a sentiment classification task for our client. The job is to evaluate the current public sentiment toward AI in social media. Given the complexity of social media data sources, are either of the selected tools right for the task?

1.1 Purpose

Artificial Intelligence (AI) has become a popular topic recently. Our PR firm has been contracted to evaluate the current public sentiment toward AI in social media like Facebook and Twitter. Our effort is to focus on identifying which free tool performs the best for this data source.

1.2 Scope

- Social media data sources limited to Twitter due to time and access restrictions.
- Twitter dataset limited by API rate limits set by Twitter at the account level.
    - Date range search limited to the previous 6-7 days. Historical requires a purchased plan.
- Free sentiment analysis tools evaluated:
    - VADER - Programmatic tool
    - SentiStrength v2.3 - GUI tool

2 Analysis and Models

2.1 About the Data

Twitter text in raw human, unstructured, input format limited to 140 characters. The text data is returned with hashtags, URLs, @ symbols, emoticons as images, and text words. The data could be from a person or a programmed bot.

Due to the developer's account limits used to pull in the data for analysis, the dataset is restricted to the prior weeks' tweets (10-5-2019 to 10-11-2019).

Focusing on AI-related tweet topics, the search criteria used to limit the tweets returned was:

- search_terms = ' Artificial+Intelligence OR machine+learning'
- Additionally adding a filter to remove retweets (those that people forward)
    - ' -filter: retweets'
- number of tweets to return = 10000

This dataset is not a good representation of the overall public sentiment on AI. The sample size is small, it's from only one data source feed, and it needs a more robust filtering mechanism to separate out non-sensible content.

2.1.1 *Dataset Info*

Collection of tweets text in sentence format. The total tweets collected was 2807. The initial file memory size was 311KB, with 5913 lines.

2.1.2 *Data Exploration & Cleaning*

The following cleaning and transformation techniques were performed programmatically in python using a jupyter notebook for code execution and visualization. The python version used was Anaconda 3.6.

The dataset passed through the following steps to remove or separate out (to be used in follow-on analysis) unusable text data for this exercise:

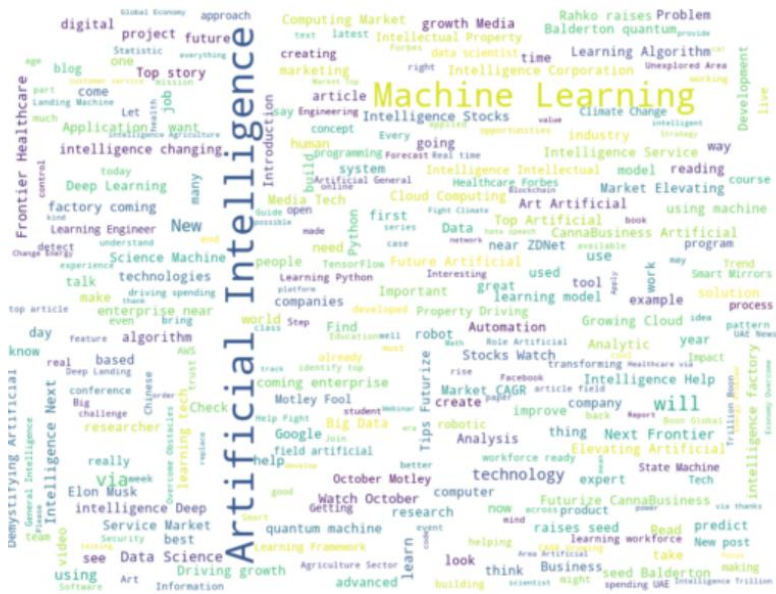
- Texts that contained emoticons were separated from the rest for future evaluation.
    - Without going into more complicated encoding techniques, this was necessary in order to write the data to a txt file throughout the cleaning and transformation steps.
- Using the NLTK TweetTokenizer, each line of text was split into tokens.
    - TweetTokenizer parameters, strip_hanles was set to True and reduce_len set to True

- URLs, Hash Tags, and English words were separated into their own distinct list collections.

Bag of Words Count	Bag of Hash Tags Count	Bag of URL Links Count
25630	1891	2777

Homework Assignment 1 (week 1)

This word cloud to the right represents the bag of words filtered from the above steps prior to removing any of the NLTK stopwords and custom words, like the search criteria, that could be applied in giving a more meaningful representation of the publics popular sentiment on AI.



This word cloud to the left represents the bag of words filtered from the above steps, with the addition of removing the NLTK stopwords list as well as a custom list of words that include the search terms. For exploratory visualization purposes, it shows a possibility in narrowing the terms in order to better represent the targetted public sentiment.

NLTK Stopwords list:

```
{'a', "you'd", 'didn', 'be', 'here', 'nor', 'you', 'up', 'they', 'has', 'their', 'him', 'other', 'wasn', 'both', 'doesn', 'h
is', 'after', 'because', 'was', "won't", 'ma', 'to', "isn't", 'shouldn', 'herself', 'again', 'her', 'while', 'how', 'y', "do
esn't", 'some', 'an', 'needn', 'couldn', 'yours', 'between', 's', 'ours', 'am', 'now', 'below', 'that', 'out', 'we', 'who',
'what', "wasn't", "mightn't", "you're", 'weren', "weren't", 'whom', 'at', 'your', 'why', 'and', 'more', 'mustn', "it's", 'yo
urselves', 'than', 'just', 'during', "hadn't", 'ain', 'from', 'had', 'own', 'myself', 'i', 'off', 'where', 'd', 'do', 'wer
e', 'too', 'over', "needn't", 'are', 'against', 'each', 'hadn', 't', 'above', 'with', 'will', 'few', 'it', 'having', 'by',
'can', 'haven', 'further', 'about', 'this', 'all', 'before', 'shan', 'those', "shouldn't", 'itself', 'my', 'once', 'yoursel
f', "wouldn't", 'does', 'our', 'into', 'these', 'doing', 'if', 'most', 'down', 've', 'being', "couldn't", 'such', "shan't",
'been', 'on', 'themselves', 'hers', 'for', 'me', 'only', 'himself', 'the', 'when', 'theirs', 'any', 'same', 'won', 'isn',
'o', "hasn't", 'wouldn', "you'll", 'did', 'but', 'then', "mustn't", 'don', 'as', 'or', 'he', "she's", 'very', 'have', 'm',
"should've", 'should', 'mightn', "you've", "that'll", 'in', 'there', 'not', 'she', 'them', 'so', 're', 'its', 'which', 'unti
l', 'through', "haven't", 'hasn', 'under', 'is', "didn't", 'll', 'of', "don't", 'aren', "aren't", 'ourselves', 'no'}
```

2.2 Models

2.2.1 *VADER - Evaluate Sentiment Classifier*

VADER (Valence Aware Dictionary and sEntiment Reasoner) is a lexicon and rule-based sentiment analysis tool that is *specifically attuned to sentiments expressed in social media*.

DESCRIPTION: Empirically validated by multiple independent human judges, VADER incorporates a "gold-standard" sentiment lexicon that is especially attuned to microblog-like contexts.

The VADER sentiment lexicon is sensitive both the **polarity** and the **intensity** of sentiments expressed in social media contexts, and is also generally applicable to sentiment analysis in other domains.

Python package: vaderSentiment.vaderSentiment **SentimentIntensityAnalyzer**

2.2.2 *VADER Parameters*

The sentiment analyzer uses it's 'polarity_scores' method to analyze and score each sentence of text in a document. For our evaluation purposes, the cleaned tweets document, which had removed all hashtags, urls, emoticons and any other unreadable text, was used as input to the 'ploarity_scores' method.

2.2.3 *VADER Details*

Behind Vader's scoring is its core sentiment analysis engine, vaderSentiment.py.

"The Python code for the rule-based sentiment analysis engine. Implements the grammatical and syntactical rules described in the paper, incorporating empirically derived quantifications for the impact of each rule on the perceived intensity of sentiment in sentence-level text. Importantly, these heuristics go beyond what would normally be captured in a typical bag-of-words model. They incorporate **word-order sensitive relationships** between terms. For example, degree modifiers (also called intensifiers, booster words, or degree adverbs) impact sentiment intensity by either increasing or decreasing the intensity. " - https://github.com/cjhutto/vaderSentiment

Vader Scoring:

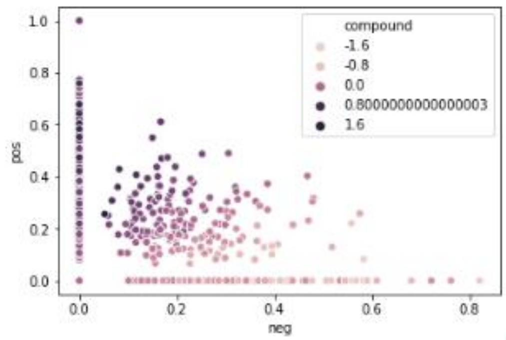
"The compound score is computed by summing the valence scores of each word in the lexicon, adjusted according to the rules, and then normalized to be between -1 (most extreme negative) and +1 (most extreme positive). This is the most useful metric if you want a single unidimensional measure of sentiment for a given sentence. " - https://github.com/cjhutto/vaderSentiment

Homework Assignment 1 (week 1)

2.2.4 *VADER Results*

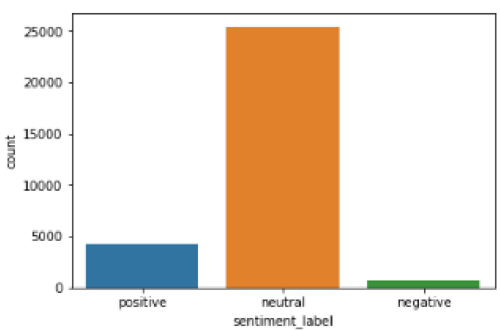
Vader polarity scoring on the cleaned AI twitter text shows and major majority of tweets in this dataset as being classified as neutral, with positive classifications of 4299 to negative classification of 744. Positive classifications 6x of negative.

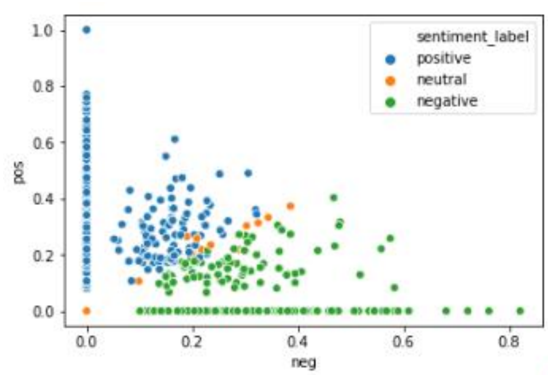
	compound	neg	neu	pos
count	30471.000000	30471.000000	30471.000000	30471.000000
mean	0.062315	0.009987	0.846487	0.050029
std	0.212454	0.056426	0.308688	0.130458
min	-0.918600	0.000000	0.000000	0.000000
25%	0.000000	0.000000	0.838000	0.000000
50%	0.000000	0.000000	1.000000	0.000000
75%	0.000000	0.000000	1.000000	0.000000
max	1.000000	0.821000	1.000000	1.000000



To standardize thresholds for classifying these sentences as either positive, neutral, or negative the recommended compound score values from the github vaderSentiment site were used.
The were as follows:

1. positive sentiment: compund score >= 0.05
2. neutral sentiment: (compund score > -0.05) and (compund score < 0.05)
3. negative sentiment: compund score <= -0.05





2.2.5 *SentiStrength v2.3 GUI - Evaluate Sentiment Classifier*

SentiStrenght is an open-source, free sentiment analysis GUI tool that can processes up to 16,000 social web texts per second "with up to human-level accuracy for English" - http://sentistrength.wlv.ac.uk/

"SentiStrength estimates the strength of positive and negative sentiment in short texts, even inform language. SentiStrength reports two sentiment strengths:

- -1 (not negative) to -5 (extremely negative)
- 1 (not positive) to 5 (extremely positive) " - http://sentistrength.wlv.ac.uk/

2.2.6 *SentiStrength Parameters*

All default values selected other than the below items.

Data Mining Options:

- +Include emoticons
- +Include all non-standard punctuation and ! and?
- +Include incorrect spelling
- Export as trinary[-1,0,1]

Sentiment Analysis Options:

- Use the Strongest Emotion of All Sentences in Comment
- Use Strongest of All Sentiment Words in a Sentence
- Allow multiple +ve words to increase+ve emotion
- Allow multiple -ve words to increase -ve emotion
- Booster words increase emotion or decrease
- Count neutral emotions as positive for emphases
- Repeated letters boost emotion
- Correct spellings due to repeated letters
- Use idium Lookup Table to Override Matching Word Strengths
- Negative words (e.g, not) flip emotion of the following word
- Never count booster words when counting intervening words after negative
- Correct repeated letter spelling differences

Sentiment Strength Analysis:

- [Speed algorithm by pre-loading whole data set]

2.2.7 *SentiStrength Details*

The same dataset and cleaned tweets text file used in the Vader analysis was used in the SentiStrength GUI tool analysis.

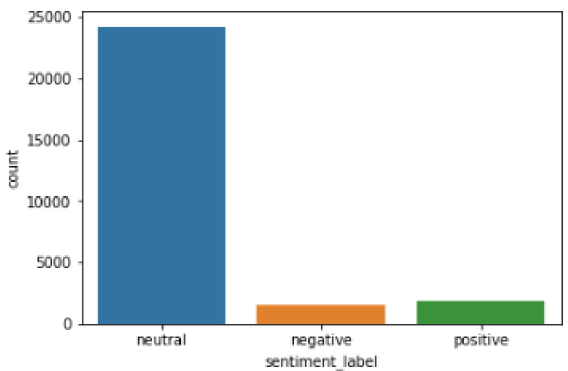
"Positive sentiment strength ranges from 1 (not positive) to 5 (extremely positive) and negative sentiment strength from -1 (not negative) to -5 (extremely negative). The sentiment strength detection results are not always accurate - they are guesses using a set of rules to identify words and language patterns usually associated with sentiment." - http://sentistrength.wlv.ac.uk

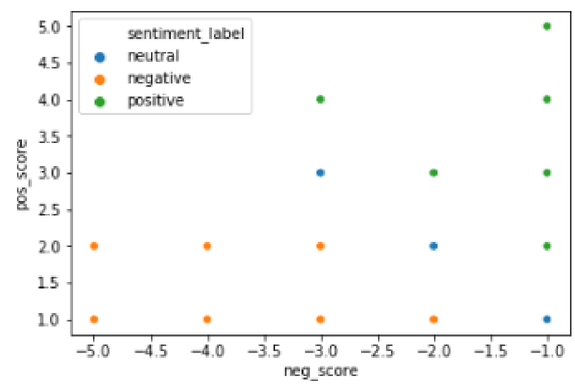
2.2.8 *SentiStrength Results*

To standardize thresholds for classifying these sentences as either positive, neutral, or negative the following criteria was applied:

The were as follows:

1. positive sentiment: (pos_score+neg_score) >= 1
2. neutral sentiment: (pos_score+neg_score) == 0
3. negative sentiment: (pos_score+neg_score) < 0





2.2.9 *p*
arison of Tools Classification Output

Comparison of classification results:

Tool	Positive Count	Negative Count	Neutral Count
Vader	4299	744	25428
SentiStrength	1907	1519	24195

The varian output could be a number of things relating to parameter settings as well as how Vader uses the compound score to determine final classification through a threshold condition. Further research would be needed to determine true reasons for these classification differences.

A few example comparisons:

Tweet Text	SentiStrength Scoring	Vader Scoring	Human Classification
Artificial intelligence helps rangers protect endangered wildlife Artificial intelligence	classification: negative pos: 1 neg: -2	classification: positive neg: 0.0 neu: 0.42 pos: 0.58 compound: 0.8074	classification: positive
Fixing The Human Problem Artificial Intelligence How	classification: neutral pos: 2	classification: positive neg: 0.117	classification: neutral

can strive for ethical data and intelligent systems	neg: -2	neu: 0.476 post: 0.407 compound: 0.7717	
EACH OTHER Artificial intelligence behind life episde starting our responsibility buffering face and his dining	classification: neutral neg: -1 pos: 1	classification: positive neg: 0.0 neu: 0.819 pos: 0.181 compound: 0.476	classification: neutral

3 Conclusions

Given the source data structure, alone these tools are not suitable for the task of reporting on the public sentiment toward AI. However, combining either of these tools with other programmatic text mining techniques that could clean and transform the input data into a more usable format would provide more meaningful results. Taking these results and applying human review of the text classifications for correctness, focused on the topic of AI, a baseline training dataset could be created for supervised learning classification algarathms to use on new AI sentiment datasets.