Machine Learning Approaches to Sentiment Analysis

Richard Fremgen

Background

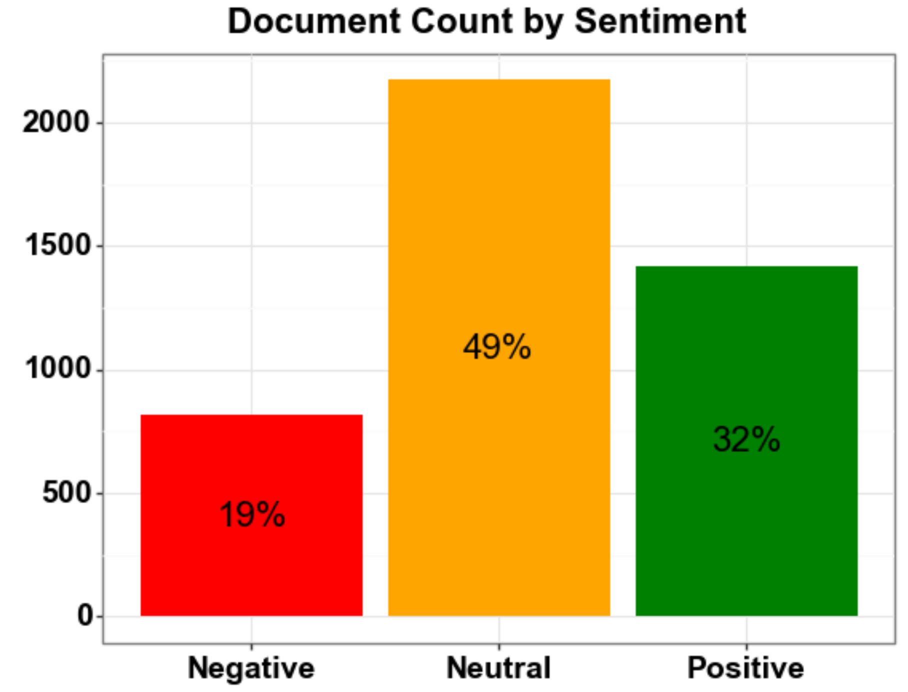
- **Brookfield Public Securities**: financial institution that invests in global alternative assets.
  - Teams are comprised of investment analysts
  - Main goal is to digest news and make buy or sell recommendations regarding a stock
- **Problem:**
  - Difficult to stay on top of news and historical trends of hundreds of companies in a respective universe
  - Analysts are very good at analyzing the current news around a company, but are limited in information retention
- **Objective:**
  - Build a sentiment analysis tool that can classify financial news articles based on polarity (positive, negative, neutral)
  - Enable high-level, macro news digestion at *scale*

Data

Labeled financial news data aggregated from two sources:

- Financial Phrase Bank (FPB)
- 2017 Semantic Workshop on Semantic Evaluation (SemEval)

Document Count by Sentiment



EDA: Word Clouds

Methods

PIPELINE

Input	*Preprocess*	*Feature Extraction*	*Modeling*	Output
Text →	Clean Data Stop Words Stemming →	Tokenize BoW TF-IDF →	Classic ML Ensemble ML CV/Bootstrap →	Positive Neutral Negative

TOKENIZE

Sentence: *the marked crashed*

Unigram: *[the, market, crashed]*
Bigram: *[the market, market crashed]*
Combo: *Unigram + Bigram*

BoW = count(t, d)
TF-IDF = BoW x [log(N+1/df$_t$+1) + 1]

- t = term, d = document
- N = total # of documents
- df$_t$ = # of docs. where term *t* occurs

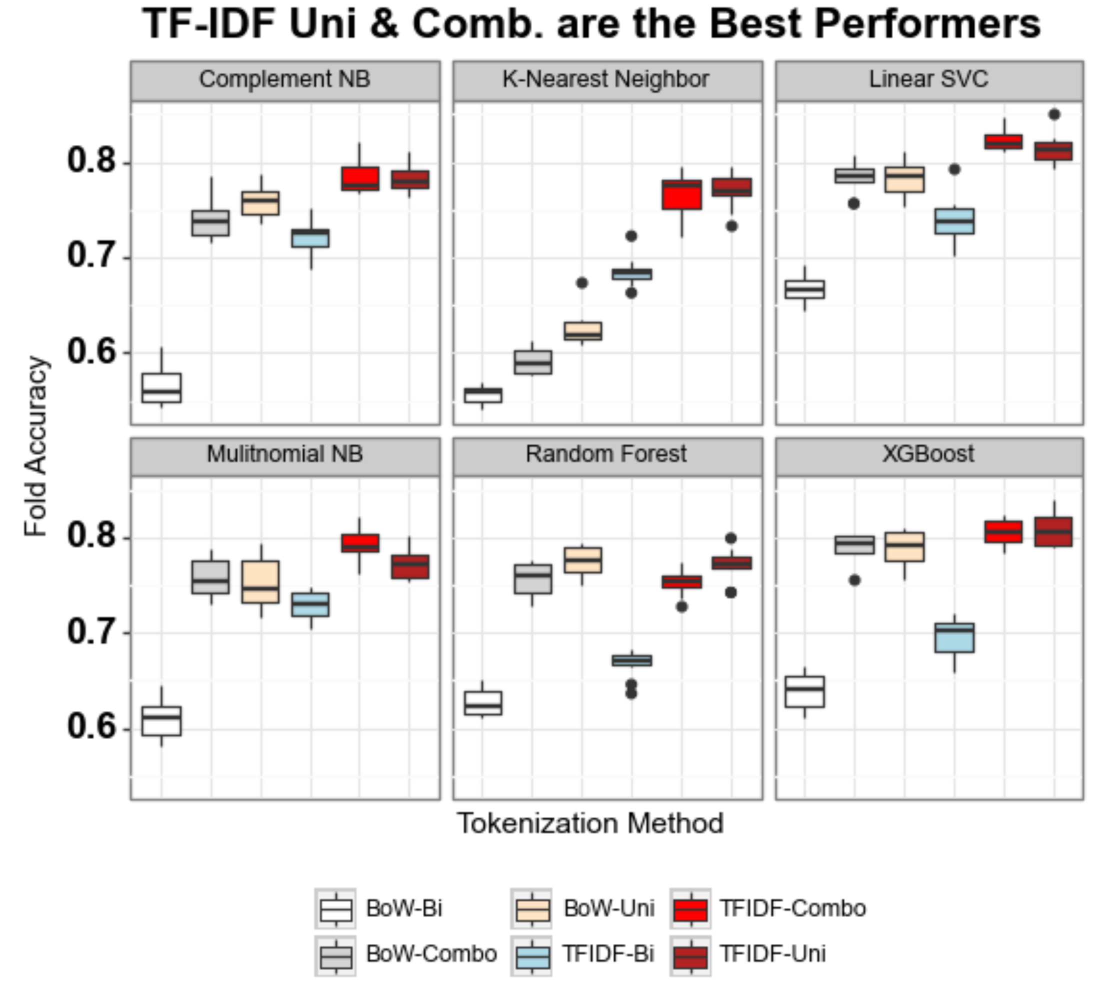
Results

- **Tokenization:**
  - TF-IDF outperformed BoW in every model setup
  - Tokenizing at the Bigram level resulted in the lowest fold accuracy across all models, however, this may be due to the limited training corpus used
- **Classifier:**
  - Linear SVC and XGBoost had highest test accuracy
  - Confusion Matrix displays difficulty in accurately predicting negative sentiment for Linear SVC

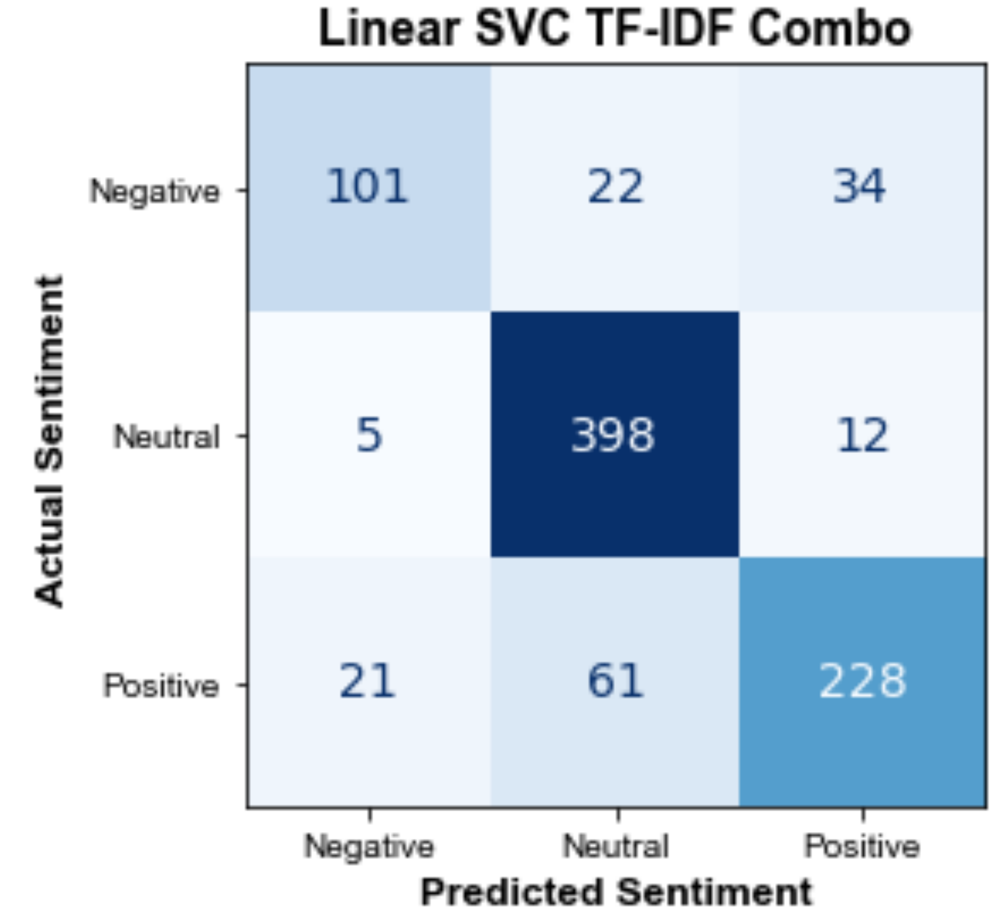
Model	Tokenization	Test Accuracy[1]	Bootstrap Test Accuracy[2]
Linear SVC	TF-IDF, Combo	82.4 %	81.5 ± 1.8 %
XGBoost	TF-IDF, Uni	81.3 %	80.0 ± 1.7 %
Comp. NB	TF-IDF, Combo	79.7 %	78.5 ± 1.9 %
Mult. NB	TF-IDF, Combo	78.8 %	77.8 ± 2.0 %
Random Forest	TF-IDF, Uni	77.4 %	76.4 ± 2.1 %
k-NN	TF-IDF, Uni	77.4 %	75.1 ± 2.8 %

Notes: 1.) Test Accuracy is computed from 80-20 Train-Test Split
2.) Bootstrap Test Accuracy is the 95% CI for Out-of-Bag Accuracy

TF-IDF Uni & Comb. are the Best Performers



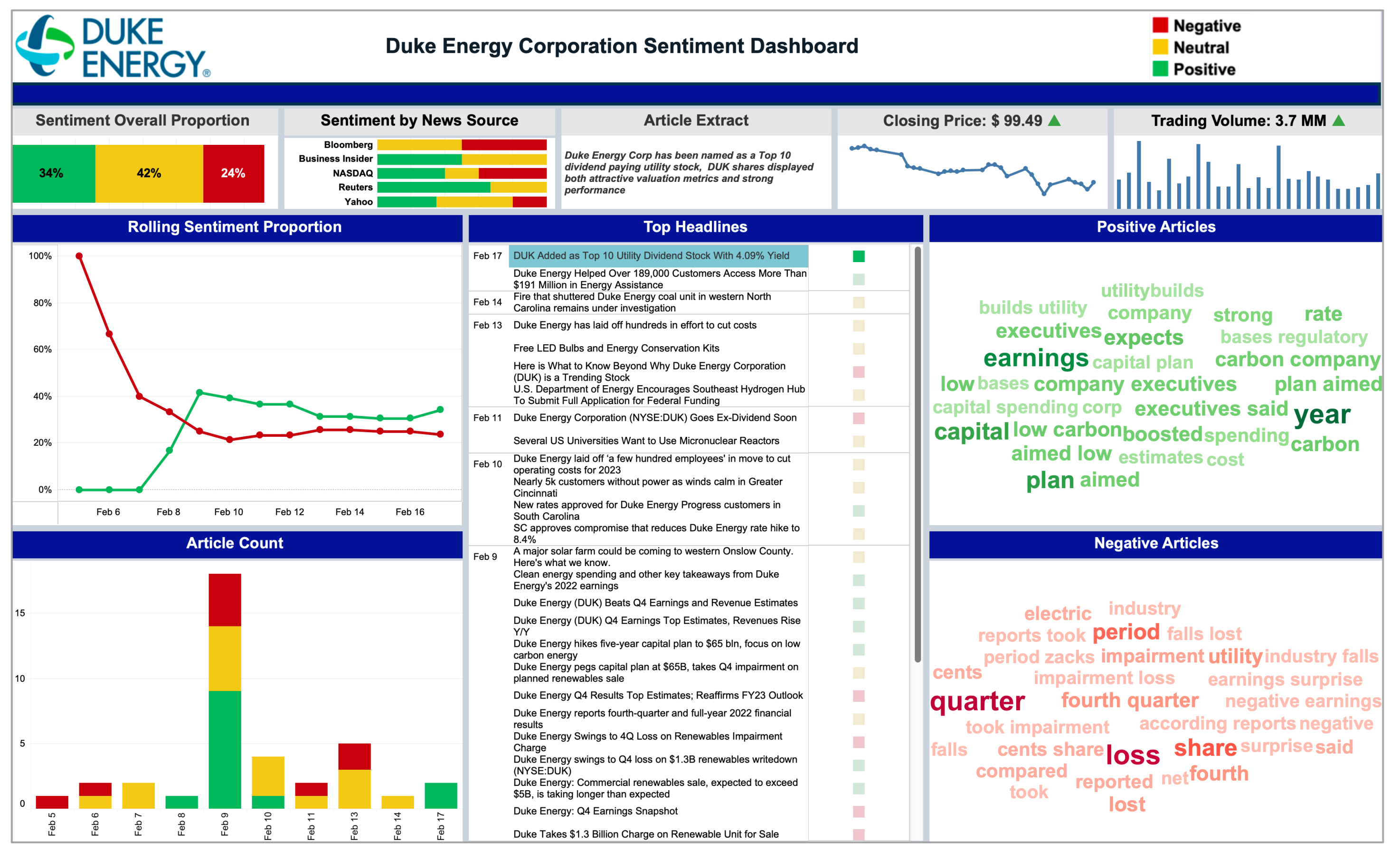
Linear SVC TF-IDF Combo



Application

- **Overview:** once the optimal sentiment analysis model was selected, measures were taken to integrate this model into a tool that can be used by the investment team.
- **Workflow:**
  - News articles are queried from Newcatcher API
  - Data is cleaned, processed, and transformed in Python
  - ML sentiment model outputs a predicted sentiment class for each article
  - Sentiment labels and article metadata are stored in a MySQL database
  - Results are displayed in an interactive Tableau dashboard

- **Dashboard Example:** *Duke Energy*



Conclusion

- TF-IDF appears to be the optimal method to transform text data; no statistical evidence to conclude that one classifier outperformed the others.
- **Next Steps:**
  - Apply deep learning methods such as LSTM and BERT, and compare results with using word embedding and word2vec
  - Expand analysis for aspect-based sentiment analysis for long text
  - Analyze how stock prices fluctuate with changes in sentiment