Machine Learning Approaches to Sentiment Analysis

Richard Fremgen

Abstract

Natural Language Processing (NLP) is an interdisciplinary subfield of computer science and computational linguistics that uses machine learning techniques to help computers understand, interpret, and manipulate human language. Perhaps one of the most prominent and widely used NLP applications is sentiment analysis, which aims to classify the polarity (positive, negative, neutral) of a piece of text. In the modern-day digital era, NLP tasks such as sentiment analysis are becoming more commonplace in industry and academia to help individuals digest immense amounts of textual data at scale. In this paper, different machine learning algorithms and NLP methods for text classification were tested and investigated under the setting of sentiment analysis. Such methods can be explored and utilized especially in the financial industry, where much of the work of a financial analyst centers around digesting information from a variety of sources to make a well-informed decision regarding the investment or divestment of a company.

1 Introduction

Anytime one interfaces with modern technology, there is a good chance that the device or system they are using leverages some form of NLP. Advances in NLP techniques over the past decade have given rise to language models such as BERT that is used within the Google Search algorithm, or GPT-3, which forms the underpinnings of ChatGPT. While there are countless use cases for the wide variety of NLP applications, text classification in the form of sentiment analysis is one that can be used to quantitatively extract the emotion or subjectivity of a given piece of text. Machine learning algorithms for classification is a rather well-studied and researched area in academia, as approaches can be easily applied to text data for the task of sentiment analysis. The objective of this project is to investigate the performance of different machine learning techniques and methods for sentiment analysis in the context of classifying financial news articles by their respective polarity. While there are many modeling configurations to test, this paper investigates how different tokenization schemes and machine learning models impact the performance of a text classifier.

1.1 Motivation

Last summer, I interned at Brookfield Public Securities, a leading alternative asset manager that invests in infrastructure, renewable power, and real estate projects across the globe. Investment teams at Brookfield are comprised of financial analysts that are responsible for researching companies in a specific universe and providing investment recommendations to their respective portfolio manager. Much of the analyst's research centers around digesting qualitative textual data from news stories or earnings call transcripts that provide keen insight into the financial health of a company. While software systems such as Bloomberg Terminal help analysts to mainstream this process, they are limited cognitively on how much data they can retain. NLP applications do, however, provide an automated way to digest textual data at scale, which is crucial for financial analysts that track hundreds of companies. As such, much of my internship was spent investigating how NLP and machine learning could be leveraged to build a sentiment news model. Since the data used to build that model was proprietary, this paper expands upon the work started during my internship, and uses only publicly available data.

1.2 Data

When considering the type of data to use to build a sentiment analyzer model, it is imperative to use domain-specific data because due to the complexity of the English language, words can have vastly different meanings and interpretations depending on how and where they are used. Since the financial industry uses such unique jargon, special care was taken to only utilize training and testing data that resembled the common lingo used in financial or economic publications. As such, in an attempt to maximize the size of the training corpus, data was collected and aggregated from the Financial Phrase Bank (FPB) and the 2017 SemEval Workshop on Semantic Evaluation (SemEval) data sets.

The FPB is one of the most cited datasets for sentiment analysis and contains a corpus of English sentences from the financial domain that were categorized by sentiment (positive, negative, or neutral) by 16 financial experts. The FPB provides metrics on the strength of majority sentiment agreement among the labelers, as this corpus was filtered to only include sentences that had at least 75 % majority sentiment agreement. The SemEval corpus comes from a 2017 NLP competition in *Fine-Grained Sentiment Analysis of Financial Microblogs and News* and contains financial news sentences that were given a sentiment score between -1 and 1 by domain experts. Since these scores were used under the logic that a score of 0 represents neutral news, a score greater than 0 represents positive news, and a score less than 0 represents negative news, this data was transformed into the same categorical sentiment labels as FPB. Combining both data sources resulted in an aggregated corpus outlined in Table 1.

Sentiment	**Count**
Positive	1,418 (32 %)
Negative	817 (19 %)
Neutral	2,713 (49%)
Total	4,408

Table 1: Sentiment Document Count

2 Methods

2.1 Preprocessing and Tokenization

Performing adequate and appropriate data preprocessing is an essential first step when building a language model, due to the inherent unorganized structure of text. Since machine learning models require a numerical input, preliminary data cleaning aims to transform textual data into a format that is most conducive for analysis. For consistency, each sentence in the data set was first converted to lowercase, before removing all non-letter characters, since characters such as numbers, punctuation, or special symbols do not determine the polarity of a sentence when looked at independently. Stop words, such as *the* and *is* were then removed, before stemming, the processing of reducing a word to its stem, was applied to all remaining words in order to reduce dimensionality.

Most sentiment analysis language models use either a Bag of Words (BoW) or Term Frequency-Inverse Document Frequency (TF-IDF) approach to convert textual data into numeric vectors. Both approaches start by first forming a Document Term Matrix (DTM), where each column in the matrix represents a unique token found in the corpus and each row represents a different document (e.g. sentence) used in training. Numerical values in the DTM are populated using either the BoW or TF-IDF technique. Under the BoW technique, this equates to using word count frequency. The shortcomings of this approach can be seen in Figure 1, which displays the most frequent (non-stop) words from all negative and positive records in the training corpus.

According to Figure 1, it appears that several words, such as *company* and *profit* are some of the most frequently used words in both positive and negative articles. This can be problematic since extensive overlap in tokens between the two corpora can impact the accuracy of the classifier. A more practical approach to populate the DTM is to compute the TF-IDF score for each document-token combination. TF-IDF is a statistic that is used to reflect how relevant a

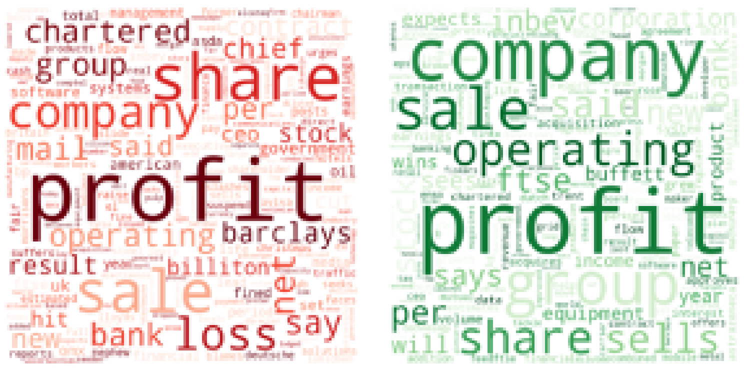


Figure 1: Negative (left) and Positive News (right) Word Clouds

word is to a document in a corpus. Words that appear in most documents, such as *company* will have smaller TF-IDF scores compared to specific phrases such as *loss* that are typically associated with negative news stories. A mathematical comparison of both feature extraction approaches can be seen in Table 2.

Another NLP technique that was considered was tokenization, which is the process of splitting textual data into smaller units or *tokens*. These tokens form the columns of the DTM, as the tokenization method used determines how large each token is. For this project, unigram, bigram, and unigram-bigram combination schemes were used. Unigram corresponds to individual words representing tokens, bigram corresponds to two consecutive words representing tokens, and the combination approach uses both unigram and bigram tokens. For each of the models described in Sections 2.2 and 2.3, six individuals models were trained and tuned based on the preprocessing schemes described above, in order to evaluate how the feature extraction methods (BoW, TF-IDF) and tokenization techniques (unigram, bigram, combination) impact the accuracy of the classifier.

Method	**Formula**
BoW	count(t,d)
TF-IDF	BoW x [log $\left(\frac{N+1}{df_t+1}\right) + 1$]

Table 2: Comparison of DTM Methods
t : term, d : doc., N : number of docs.
df_t : number of docs. where term t occurs

2.2 Conventional ML Approaches

At its core, sentiment analysis is a supervised machine learning (ML) classification problem, where given a sequence of words, a model can be trained to classify the overall sentiment of that text. The three conventional ML classifiers tested were: the Naïve Bayes Classifier, k-nearest neighbors (KNN), and Linear Support Vector Classifiers (Linear SVC). The aggregated labeled sentiment data was split into an 80-20 training-testing split for model comparison and selection. For hyperparameter tuning, 10-fold cross-validation was used, where average fold accuracy was used for model tuning purposes.

2.2.1 Naive Bayes

Naive Bayes is a simple, but effective probabilistic classifier that uses Bayes' Theorem to solve text classification problems. A key advantage of Naive Bayes classifiers is that they are computationally fast and can be trained on small amounts of data, but do make the naive assumption of conditional independence of tokens given a particular sentiment class, which is not always valid. The two Naive Bayes classifiers tested were Multinomial Naïve Bayes (MNB) and Complement Naïve Bayes (CNB). MNB assumes that the conditional probability of a token given a sentiment class comes from a multinomial distribution. CNB is an adaption of MNB, but instead of computing the likelihood of a token occurring in a class, the likelihood of a token occurring in other (complement) classes is used. Due to this feature, CNB works particularly well with imbalanced data sets when there are large differences in class proportions. Both approaches utilized Laplacian Smoothing to handle zero-probability instances, where the smoothing parameter `alpha` was tuned during cross-validation.

2.2.2 KNN

KNN is a non-parametric, supervised algorithm that can be used for both classification and regression problems. The KNN classifier uses instance-based learning to classify new data, meaning that the algorithm does not construct a general internal model, but rather stores training data instances for use in predicting test data. New records are classified via a simple majority voting mechanism based on the class labels of the k nearest neighbors. While KNN has the advantage of being easy to implement and requires no training, this classifier does not perform well with large data sets or with high dimensional data, which is usually the case with text classification problems.

2.2.3 Linear SVC

Support Vector Machines (SVM) are a class of supervised learning algorithms that work to find an optimal boundary or hyperplane in n-dimensional space that can distinctly separate classes in a data set. Linear SVC is a type of SVM that uses a linear kernel and is commonly used for data sets with many features, since increasing the dimensionality of such data sets does not improve separability, and therefore does not require more complex kernels. Text classification tasks, such as sentiment analysis, inherently train on data sets with many features, based on the Document-Term Matrix that is formed, which provided motivation to test the performance of a linear kernel. Using a Linear SVC also has the advantage of only having to hyptertune the regularization parameter, `C`, whereas other kernels such as the Radial Basis Function (RBF) required additional parameter tuning.

2.3 Ensemble Approaches

In addition to the three conventional ML approaches discussed, ensemble approaches such as Random Forest and eXtreme Gradient Boosting (XGBoost) were trained and tested on the financial training corpus using the same 80-20 train-test split and 10-fold cross-validation setup described in Section 2.2

2.3.1 Random Forest

The Random Forest classifier is comprised of a large number of relatively uncorrelated individual decision trees that work together in an ensemble for classification tasks. Random Forests use a concept known as *bagging*, where individual decision trees are formed based on bootstrapped samples from the training corpus, which then through a majority vote determine the classification of a new record. Random Forests are known to behave well in high-dimensional settings and can handle noisy data, such as that used in sentiment analysis tasks.

2.3.2 XGBoost

A major drawback of the Random Forest algorithm is that it utilizes parallel learning when building an ensemble of decision trees, meaning that such an approach cannot *learn* from mistakes made in previous decision trees. XGBoost is a type of gradient-boosted decision tree algorithm that provides a solution to this problem by using *boosting*, where new trees are formed by considering the errors of trees from previous rounds. This approach is commonly referred to as sequential learning since boosting transforms weak learners into strong learners. Due to the iterative nature of boosting, the amount of time it takes to train gradient-boosted trees is a cause of concern. As such, XGBoost was used over scikit-learn's Gradient Boosting Classifier (GBC), primarily because XGBoost is a more regularized form and delivers high performance when compared to GBC because of the ability to be parallelized and distributed across clusters.

3 Results and Discussion

Each model was fit and tuned six separate times using the different tokenization schemes outlined in Section 2.1. Table 3 displays the test accuracy *(Test Acc.)* of the top-performing tokenization schemes for each model using the initial 80-20 train-test split. Aside from computing traditional test accuracy, classifier uncertainty was quantified by using bootstrapping in order to compute average Out-of-Bag accuracy and 95% confidence interval intervals *(OOB CI)* from 200 bootstrapped samples. From an accuracy perspective, Linear SVC and XGBoost were the top-performing models, classifying over 80% of the test set correctly and having the highest average OOB accuracy. CNB slightly outperformed MNB, which was expected considering the slight imbalance of the data. In spite of this, there is no statistical evidence to support the claim that one classifier significantly outperformed the remaining classifiers, as evidenced by the extensive overlapping *OOB CI* and marginally close *Test Acc.*

Model	**Test Acc.**	**OOB CI**
Linear SVC	82.4 %	81.5 ± 1.8 %
XGBoost	81.3 %	80.0 ± 1.7 %
CNB	79.7 %	78.5 ± 1.9 %
MNB	78.8 %	77.8 ± 2.0 %
RF	77.4 %	76.4 ± 2.1 %
KNN	77.4 %	75.1 ± 2.8 %

Table 3: Model Results by Test and OOB Accuracy

Considering the performance of tokenization schemes, Figure 2 (Appendix) illustrates the distribution of fold (validation) accuracy by model type and tokenization method for the top-performing hyperparameters in each model-tokenization combination. For each class of models, the TF-IDF Unigram and Combination tokenization methods (red box plots) had the highest median fold accuracy and were the tokenization technique used in every one of the final models in Table 3. Generally speaking, Bigram methods (BoW-Bi, TFIDF-Bi) had the lowest classification accuracy, which could potentially be due to the limited text size of the training corpus, since only individual sentences were used.

Ultimately, Linear SVC TF-IDF Combo was selected as the final model to build a sentiment analyzer around since the overall accuracy of the classifier was valued by the investment team over specific class accuracy. Figure 3 (Appendix) displays a Confusion Matrix for this model, which aside from its test accuracy, posted a precision of 0.818, recall of 0.780, and F1 score of 0.794. It is apparent, however, in Figure 3 that this classifier does encounter difficulties when classifying negative news, as this was the minority class in the training corpus. Preliminary attempts to reconcile this class imbalance were attempted by using the Synthetic Minority Over-Sapling Technique (SMOTE) on the training corpus, however, this led to severe overfitting on the training corpus.

From an application standpoint, a news aggregation API called *Newscatcher* was used to programmatically extract news articles in Python about companies of interest from reputable news sources. The trained Linear SVC model was then used to classify the overall sentiment of each article's extract, as results were then fed into an interactive Tableau dashboard that can be seen in Figure 4 (Appendix).

4 Conclusion

In this project, machine learning classifiers and NLP techniques were tested under the context of sentiment analysis. Modeling results showed that using TF-IDF scores and a unigram or combination tokenization scheme generally outperformed using a BoW approach. Linear SVC and XGBoost models were found to have the highest test accuracy, however, no statistical evidence was found to conclude that one model significantly outperformed the others. Future work will focus on testing deep learning approaches to sentiment analysis, along with using other data preprocessing schemes such as word2vec and word embeddings. In addition, aspect-based sentiment analysis will be explored to replace the brute-force approach taken to classify the new's sentiment around a company.

5 References

1. Jurafsky, Daniel and Martin, James H.. Speech and language processing. 2. ed., [Pearson International Edition] London [u.a.]: Prentice Hall, Pearson Education International, 2009.

2. Hastie, Trevor, et al. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. 2nd ed. New York, Springer, 2009.

3. Malo, P., Sinha, A., Takala, P., Korhonen, P. and Wallenius, J. (2013): "Good debt or bad debt: Detecting semantic orientations in economic texts." Journal of the American Society for Information Science and Technology. (in Press)

4. K. Mishev, A. Gjorgjevikj, I. Vodenska, L. T. Chitkushev and D. Trajanov, "Evaluation of Sentiment Analysis in Finance: From Lexicons to Transformers," in IEEE Access, vol. 8, pp. 131662-131682, 2020, doi: 10.1109/ACCESS.2020.3009626.

5. Sudhir, P., amp; Suresh, V. D. (2021). Comparative study of various approaches, applications, and classifiers for sentiment analysis. Global Transitions Proceedings.

6. Moralwar, S. B., amp; Deshmukh, S. N. (2015). Different Approaches to Sentiment Analysis. International Journal of Computer Science and Engineering, 3(3).

7. Jain, A. (2023, February 9). XGBOOST parameters: XGBoost parameter tuning. Analytics Vidhya. Retrieved February 14, 2023, from https://www.analyticsvidhya.com/blog/2016/03/complete-guide-parameter-tuning-xgboost-with-codes-python/

8. Fazlija, B., amp; Harder, P. (2022). Using Financial News Sentiment for Stock Price Direction Prediction. Mathematics, 10(2156).

9. Sharma, M. (2020, May 11). Sentiment Analysis (introduction to naive Bayes algorithm). Medium. Retrieved February 14, 2023, from https://medium.com/towards-data-science/sentiment-analysis-introduction-to-naive-bayes-algorithm-96831d77ac91

10. T., B. (2022, December 9). Comprehensive guide to multiclass classification with Sklearn. Medium. Retrieved February 14, 2023, from https://medium.com/towards-data-science/comprehensive-guide-to-multiclass-classification-with-sklearn-127cc500f362

11. Li, S. (2018, February 20). Multi-class text classification with Scikit-Learn. Medium. Retrieved February 14, 2023, from https://medium.com/towards-data-science/multi-class-text-classification-with-scikit-learn-12f1e60e0a9f

12. Pietro, M. D. (2022, March 22). Text classification with NLP: TF-IDF vs word2vec vs Bert. Medium. Retrieved February 14, 2023, from https://medium.com/towards-data-science/text-classification-with-nlp-tf-idf-vs-word2vec-vs-bert-41ff868d1794

13. Pramoditha, R. (2021, November 6). Introduction to boosted trees. Medium. Retrieved February 14, 2023, from https://towardsdatascience.com/introduction-to-boosted-trees-2692b6653b53

14. KOWALCZYK, A., amp; Alexandre KOWALCZYKI am passionate about machine learning and Support Vector Machine. I like to explain things simply to share my knowledge with people from around the world. (2020, July 26). Linear Kernel: Why is it recommended for text classification ? SVM Tutorial. Retrieved February 14, 2023, from https://www.svm-tutorial.com/2014/10/svm-linear-kernel-good-text-classification/

6 Appendix

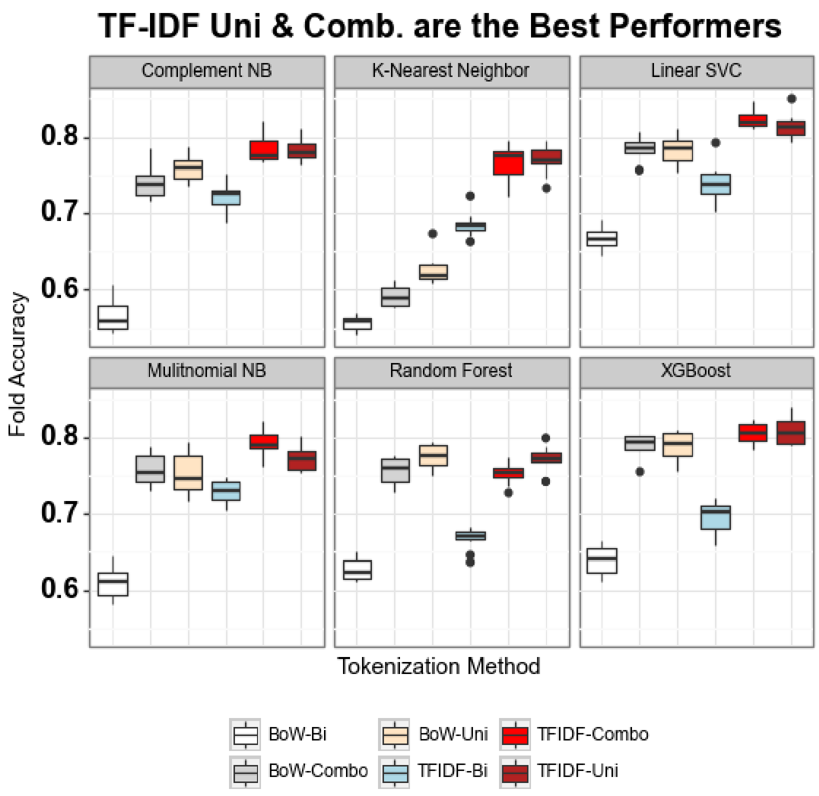
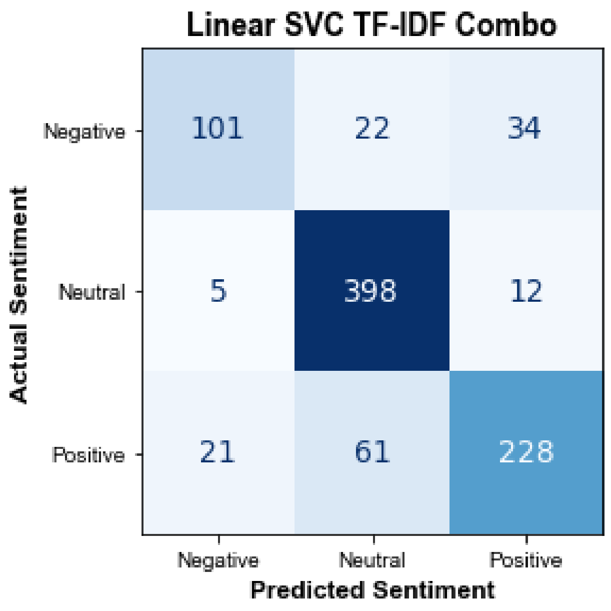
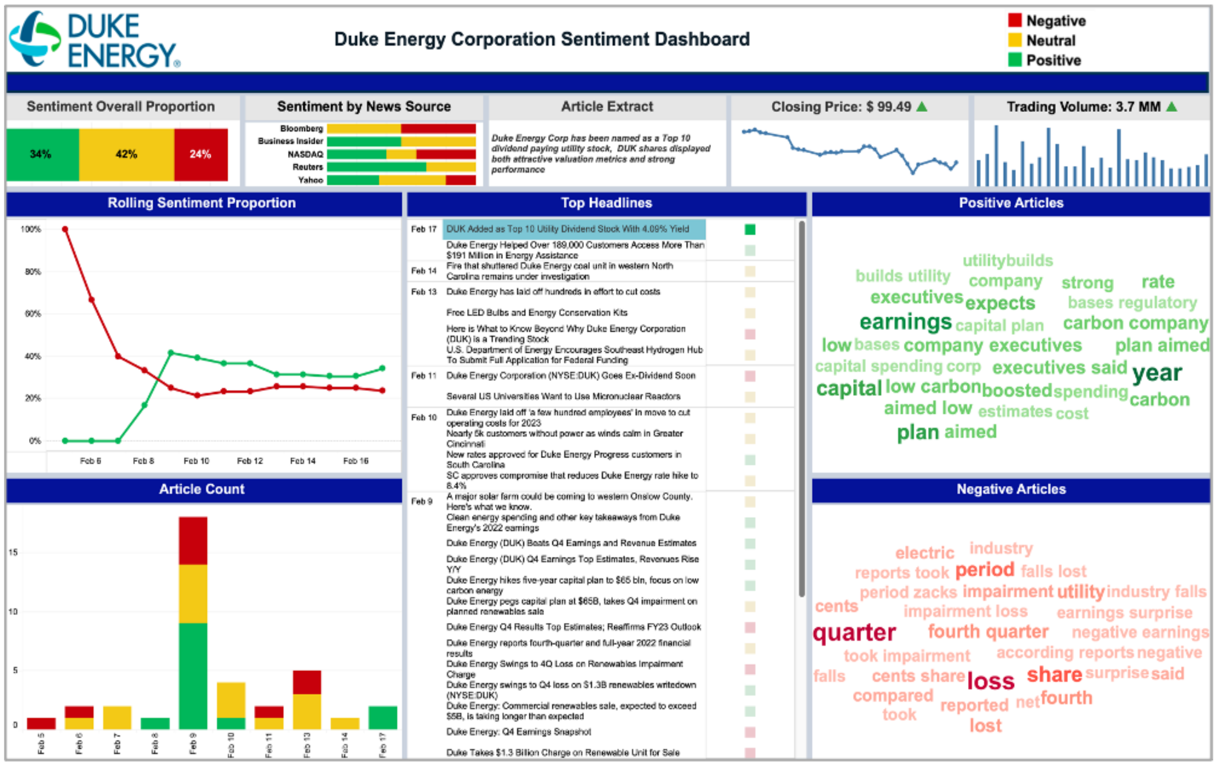


Figure 2: Comparing Tokenization Techniques



	Negative	Neutral	Positive
Negative	101	22	34
Neutral	5	398	12
Positive	21	61	228

Figure 3: Linear SVC TF-IDF, Combo Confusion Matrix



Duke Energy Corp has been named as a Top 10 dividend paying utility stock, DUK shares displayed both attractive valuation metrics and strong performance

Feb 17	DUK Added as Top 10 Utility Dividend Stock With 4.09% Yield	🟩
	Duke Energy Helped Over 189,000 Customers Access More Than $191 Million in Energy Assistance	🟩
Feb 14	Fire that shuttered Duke Energy coal unit in western North Carolina remains under investigation	🟨
Feb 13	Duke Energy has laid off hundreds in effort to cut costs	🟨
	Free LED Bulbs and Energy Conservation Kits	🟨
	Here is What to Know Beyond Why Duke Energy Corporation (DUK) is a Trending Stock	🟥
	U.S. Department of Energy Encourages Southeast Hydrogen Hub To Submit Full Application for Federal Funding	🟨
Feb 11	Duke Energy Corporation (NYSE:DUK) Goes Ex-Dividend Soon	🟥
	Several US Universities Want to Use Micronuclear Reactors	🟨
Feb 10	Duke Energy laid off 'a few hundred employees' in move to cut operating costs for 2023	🟨
	Nearly 5k customers without power as winds calm in Greater Cincinnati	🟨
	New rates approved for Duke Energy Progress customers in South Carolina	🟩
	SC approves compromise that reduces Duke Energy rate hike to 8.4%	🟨
Feb 9	A major solar farm could be coming to western Onslow County. Here's what we know.	🟨
	Clean energy spending and other key takeaways from Duke Energy's 2022 earnings	🟩
	Duke Energy (DUK) Beats Q4 Earnings and Revenue Estimates	🟩
	Duke Energy (DUK) Q4 Earnings Top Estimates, Revenues Rise Y/Y	🟩
	Duke Energy hikes five-year capital plan to $65 bln, focus on low carbon energy	🟩
	Duke Energy pegs capital plan at $65B, takes Q4 impairment on planned renewables sale	🟨
	Duke Energy Q4 Results Top Estimates; Reaffirms FY23 Outlook	🟥
	Duke Energy reports fourth-quarter and full-year 2022 financial results	🟨
	Duke Energy Swings to 4Q Loss on Renewables Impairment Charge	🟥
	Duke Energy swings to Q4 loss on $1.3B renewables writedown (NYSE:DUK)	🟩
	Duke Energy: Commercial renewables sale, expected to exceed $5B, is taking longer than expected	🟩
	Duke Energy: Q4 Earnings Snapshot	🟥
	Duke Takes $1.3 Billion Charge on Renewable Unit for Sale	🟥

Figure 4: Duke Energy Sentiment Dashboard Example