Department of Digitization (DIGI)

M.Sc. in Business Administration and Data Science

Natural Language Processing and Text Analytics

Final Examination Project

Evaluating Sustainability Impacts based on Company Tweets

Authors:
Henry Stoll, 141636
Alisa Ilina, 141804
Simon Christensen, 111803
Manuel Escotto Kuhn, 146907

Supervisors:
Raghava Rao Mukkamala
....

June 15, 2021

Pages: 15

Abstract

What was the topic? What was the problem formulation? What was the research question? What were the concepts? What was the dataset and what were the main data analytics methods and tools? What were the most important results in terms of meaningful facts, actionable insights, and valuable outcomes What are the conclusions and recommendations?

Keywords: at least five

Contents

1 Introduction

The social and ethical dimensions of businesses play an increasingly vital role in the sustainable growth of organizations. This concept is commonly referred to in literature as corporate sustainability, which can be defined as the impact a firm's policies in terms of environmental, social, and economic governance (ESG) have on society (?, ?). Previous research has shown that a sustainable relationship with the different stakeholders of an organization (such as customers, competitors, investors, supply chain, etc.) is likely to enhance the firm's long-term value creation. This, together with a growing demand for sustainable products on the consumers' side and the increasing interest of portfolio managers in more socially responsible investments (SRI), has incentivized companies to declare themselves sustainable and release sustainability reports together with the usual annual reports (?, ?).

According to the United Nations Sustainable Stock Exchange (2015), by 2030 at the latest, all large companies are expected to report and disclose their ESG practices or provide a valid reason of not doing so. The importance of reporting such information has been backed up by previous research, where authors argue that the disclosure of non-accounting information can serve as an indicator of how a firm manages the business risks, and that better ESG scores were even potentially associated with lower business risks (?, ?). The impact of ESG values on investors' decisions has also been evidenced in previous studies, such as the case of Aureli2020, who concluded that ESG information had an evident impact on the market value of various firms analyzed within the study.

However, while there is compelling evidence that demonstrates the relevance of ESG ratings for companies, this sustainable reporting is still in its infancy and the determinants of a good or bad ESG score have not yet been fully established. On the one hand, as argued by Elzahar2015, since ESG information is a type of non-financial disclosure and thus does not follow a standardized reporting format as financial documents do, ESG reporting tends to be rather heterogeneous. On the other hand, while there exist multiple sustainability indices, such as the Dow Jones Sustainability Emerging Markets Index, the MSCI Emerging Markets Environmental, Social, and Governance (ESG) Index, and the FTSE4Good Emerging Indexes, ESG rating agencies normally use different assessment criteria and research methodologies to evaluate companies (?, ?), which further complicates the task of determining which specific factors might influence ESG ratings.

While there are multiple factors that might influence the ESG scoring of organizations, little attention has been given to the link between these ratings and social media, specifically in the case of Twitter, which is the most frequently used social media platform of the Fortune 500's (?, ?).

With the exponentially increasing volume, velocity, variety and veracity of social media data, as well as sophistication of technologies, Twitter becomes a crucial source of acquiring sustainability impact information. Such dynamic is manifesting the potential for Machine Learning (ML) and Natural Language Processing (NLP) techniques to become the method of critical information extraction from social media. For instance, multiple studies have shown that the Twitter content may help predict macroeconomic market indicators, such as (?, ?), who found out that tweet sentiments are predictive of the Doe Jones Industrial Average closing values. Therefore, based on this premise, the paper will aim to answer the following research question:

1. *Is there a correlation between company tweets and its sustainability rating?*
2. How does communication of companies with low ESG scores differ from those scoring high? maybe??? feature importance?

2 Related Work

While there is still no previous research on the specific topic of ESG rating prediction based on Twitter sentiment analysis, other studies have analyzed (and successfully proven) the predictive power that tweets have for different organizational indicators. For instance, Sprenger2014 analyzed the sentiment of ca. 250.000 stock-related tweets using Naïve Bayesian text classification and established that they can serve as proxies for belief formation and investment behavior. Similarly, Osatuyi2019 established that the emotional factors (positive or negative) present in company tweets are a strong predictor for the variance of stock prices of a company.

Speaking specifically about corporate social responsibility (CSR),(?, ?) discovered that textual data, particularly social media, is heavily loaded with CSR-related topics, some of the most popular ones being company strategy, community charity (such as crowdfunding initiatives), philanthropy, and green initiatives, among others. This finding was further confirmed by Gomez-Carrasco2021, who even took a step further and found out that the CSR information organizational use of social media is usually for legitimacy purposes. In other words, this study showed that companies tend to use social media more as a stakeholder management tool (such as communication of cultural projects) rather than for communication of topics with societal impact (such as equality and diversity in the workplace).

In the research by (?, ?), election tweet binary classification task was performed. The authors evaluated the impacts of parameters in embedding training process, such as context windows, dimensionality and number of negative samples. The findings showed that the right choice of embedding model used with CNN led to drastic improvements compared to baseline and random classifiers. In general, numerous papers observed word embeddings followed by deep neural networks to be effective in Twitter related classifications, such as, for instance, in sentiment analysis performed by (?, ?) or sentiment-specific word embedding Twitter corpus framework proposed by (?, ?). However, literature reveals rather few multinomial classification tasks based on Twitter were performed.

3 Conceptual Framework

Concepts of relevant to research problem data analytics methods and techniques. Problem statement with problem modelling (if relevant)
Twitter is a microblogging service that allows its users to publish short messages of up to 280 characters, popularly known as ”tweets” (?, ?). As the authors further explain, users have the choice of subscribing (i.e. follow) specific accounts of their interest, or else, to look for tweets with specific keywords.
Tweets differ from regular text due to the fact that very little information gets repeated, thus rendering a string of independent characters. As Tweets are limited to 280 characters, abbreviations, grammar and syntactic errors, and slang are likely to appear which might be unknown to the used pre-trained embeddings, e.g. GloVe. Moreover, hyperlinks, mentions, hashtags and other internet-specific terms are common. Twitter often has noisy data, with texts uncommon to regular corpora. Sosa!!!

ESG risk rating measures the extent to which ESG issues are putting company's enterprise value at risk, particularly the magnitude of the unmanaged ESG absolute risk which is comparable across industries and companies. Exposure to risk, meaning the vulnerability of it, for instance, takes into account the industry as it affects the ESG risks a company may face (e.g. oil companies are more prone to environmental risks, while tech companies are more prone to privacy issues). The metric also includes the management processes in place, such as health and safety systems, employee fatality rate, and others. Negligent, low, medium, high, and severe risk categories are identified

4 Methodology

4.1 Data Analysis Process/Workflow

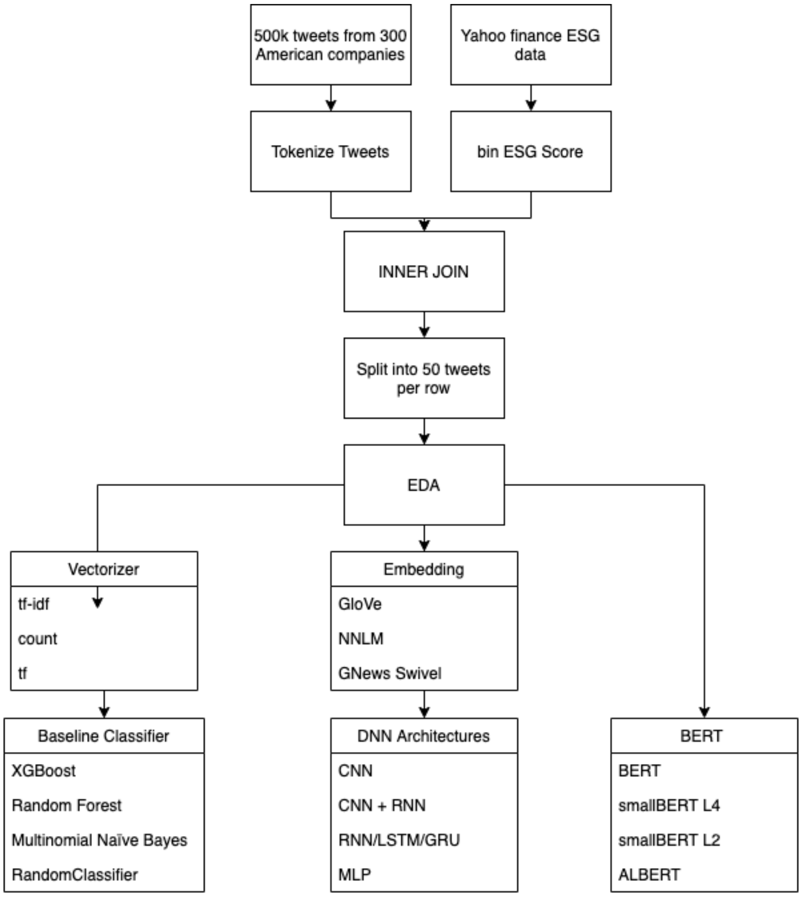

Figure 1: Process Diagram

The workflow is presented in the diagram above in figure 1, with the models sorted in the descending performance order. The tweets of 300 American companies are merged with ESG scores from Yahoo Finance and the ESG Score binned in risk categories. After cleaning and tokenization, the tweets were used to perform multinomial classification per ESG risk categories. Grid Search was employed to find the best combination of vectorizers and classifiers to serve as a baseline. Learnings from the traditional models were used in selecting Embeddings and then used for comparing different DNN Architectures. Lastly different pretrained BERT Models were used to compare against the results of DNNs.

4.2 Data Acquisition

Twitter handles and usernames were manually obtained for 300 American companies. The `Twitter API v2` was first used to get the user ID based on the Twitter handle. Then, tweets were crawled for each companies username starting from May 2021 up to 2019, resulting in at max 3250 tweets per company (due to API limitations) with some negligent imbalance, and amounting to 500 000 tweets in total. The ESG Risk scores were obtained through web-crawling it from Yahoo Finance. The ESG scores categorical bins were used as target labels, thus identifying the task as multinomial classification.
MENTION HOW DATA WAS SPLIT FOR TRAINTEST

4.3 Data Pre-Processing and Normalisation of Tweets

The data was joined with the tweets and split into 50 tweets per row. Any company that had under 50 tweets was discarded. Despite Sustanalytics providing five categories for ESG risk scores, only four bins (low, medium, high, severe) were used, omitting the first category (negligent) since no companies in the dataset depicted negligent ESG risk.

Pre-processing techniques reduce noise and render a consistent transformed dataset further to be used in modelling. Prior to natural language processing of tweets, the text was normalised and pre-processed with the use of tokenisation, lemmatisation and stop words removal. Tokenisation segmented the tweets into separate tokens, thus extracting relevant features, further tokens were lower-cased. It is common practice to use deterministic techniques based on RegEx since carefully crafted algorithm can help avoid any ambiguities in text (?, ?). Thus, RegEx was utilized for hashtag and username symbols removal. Then applying the processing of TextVectorizer of the Keras library, it splits each sample into substrings (usually words), then recombines the substrings into tokens (usually n-grams). Afterwards, it indexes tokens (associates a unique `integer` value with each token) and finally transforms each sample using this index, into a vector of `integers`. After the initial data preprocessing, the TextVectorizer in the Keras library analyzes the dataset, whereafter it determines the frequency of individual string values, and create a 'vocabulary' from them. The output is a vocabulary that contains either an unlimited or capped size, depending on the configuration options. If there exist more unique values in the input than the maximum parsed vocabulary size, the most frequent terms will be used to create the vocabulary. The values of 20.000 most frequent terms were used as an input along with a maximum output sequence length of 400.

4.4 Dataset Description

Exploratory data analysis reveals the dataset to be quite vast, yet the distribution of ESG Risk scores depicts the majority of companies scoring low and medium ESG Risk, making high and severe categories relatively under-representative. The dataset only contains recent ESG rankings and does not account for historical ESG labels. The average tweet length is 170 characters. The industry scoring highest in ESG Risk is the Energy, possibly due to its wide exposure to environmental risks, as it is one of the three categories comprising ESG Risk score. Sentiment analysis using VADER was performed to get an overview of emotional context of Tweets. Identifying companies with least positive tweets (lowest compound score), NewsCorp displayed lowest sentiment score, possibly since the tweets are associated with news events content likely to be rather impassive. Sentiment analysis shows that among most common negative words used are “apologise, sorry, issue, help, inconvenience, frustration”, while positive words include ”thanks, learn, experience, information, team”.

The most frequently used words (unigrams) for each of the four labels as seen in figure XX often revolt around direct support through twitter or directions towards contact information

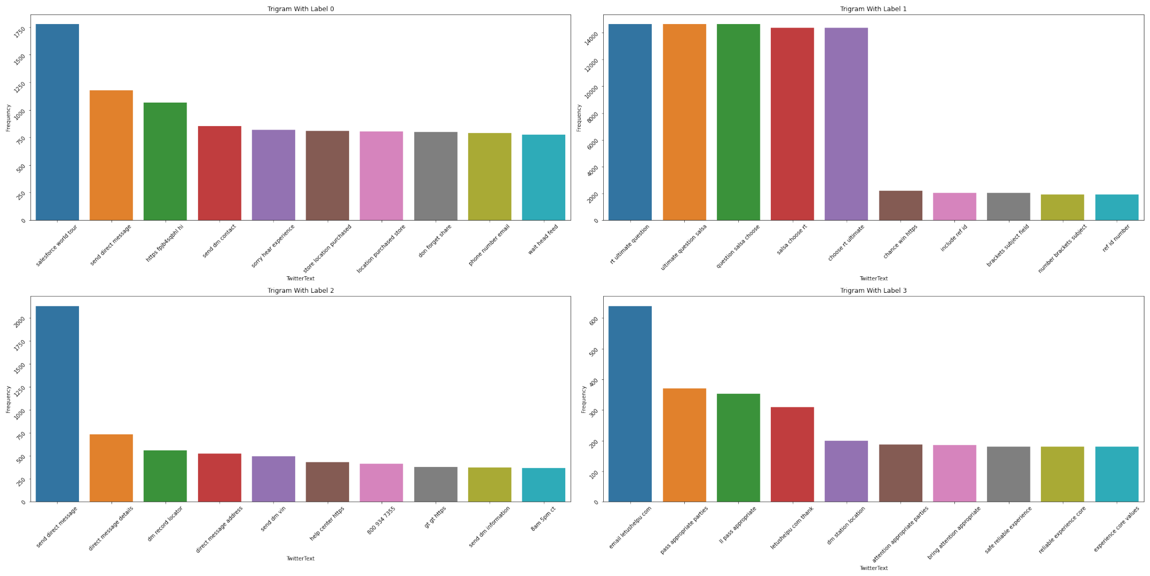

Figure 2: Most frequently used words per label

I THINK WE SHOULD SKIP THIS. I cannot see any tendency between label 0 and 3.

4.5 Data Filtering, Transformation and Combination (if relevant)

4.6 Data Analytics: Modeling, Methods and Tools

4.6.1 Traditional Classifiers

Prior to application of classifiers, several NLP vectorizing techniques were used with attention being paid to max_features, to keep the training time low, as well as n_gram ranges. After GridSearch selection of best performing vectoriser, it was further piped with machine learning models, which were evaluated to select the best algorithm.

Count Vectorization leverages frequencies in which a token appears in a tweet. CountVectorizer generates a document term matrix with each cell representing the count of terms in a document given a vocabulary. To demonstrate, given a vocabulary `['exams','are','stressful']` and a sentence

```
'Exam is life, but life is somewhat stressful'
```

Would result in the following vector:

```
[1, 0, 2]
```

Sklearn's implementation does allow to define an n-gram range for their implementation. Without supplying the class with an n-gram range (m,n), it follows a Bag of Words model.

TFIDFVectorization computes a sparse model which defines word meanings by computing term frequencies (tf) and inverse document frequency (idf). Tf assists in identifying the frequency of word t in the document d, as the CountVectorizer described above. A common technique is to lower the term frequency to avoid skewness, by taking $\log_{10}$ which results in the following calculation:

$$tf_{t,d} = \begin{cases} 1 + \log_{10}, & \text{if count(t,d)} > 0 \\ 0 & \text{otherwise} \end{cases}$$

The idf is computed by $idf_t = \log_{10}(\frac{N}{df_t})$, where N is the total number of documents and df_t is the number of documents with contain the term t. It is used to allocate higher weight to words in only a few documents. The difference in performance of tf-idf vectorization compared to count vectorization will analysed in section XX (results) and XX. Max_features will be considered in order to optimise the train time. In order to obtain a high TF-IDF weight for a given term, the term needs to contain a high term frequency(tf) in a given document (local parameter) and a low document frequency of the term across all documents (global parameter) (?, ?).

n_grams also generate a document term matrix, yet the count here represents the combinations of adjacent tokens in length n. A smaller n might not explain enough information, while bigger n will overload the information on features.

The best performing vectorizer then was further evaluated and piped into the following classifiers:

Random Forest addresses the risk of overfitting that arises from making the classifiers learn from too many features and from potentially noisy data (as is the case of tweets). The individual decision trees that comprise the random forest may perform well on prediction, but they could eventually overfit a portion of the dataset. An averaging of the results is an effective alternative to overcome this issue (?, ?). In the case of Random Forest (as well as for all other classifiers that will be mentioned in the following sub-sections), grid search was applied to find the model that best suits the data.

Multinomial Naïve Bayes Classifier is a frequency-based model designed to determine the number of times a specific word or term occurs in a document. This in order to establish whether a term is useful for the analysis or not, which is a pivotal metric in determining the sentiment of a text (?, ?). For instance, a term may occur multiple times in a document, which would increase its frequency, but it may also be a stopword that agruably adds no meaning to the text, in which case it should be removed to achieve a better model accuracy.

XGBoost is a scalable machine learning system for tree boosting. It achieves optimal results by training each new instance with the aim to emphasize the training instances previously mis-modeled for better classification results (?, ?). The algorithm can be best described as a combination of classification and regression trees, but with a re-defined objetive function in terms of complexity of the trees and training loss in order to mitigate the risk of overfitting the data, which allows it to be a highly robust and accurate model.

4.6.2 Neural Language Model Architectures

Generally, neural networks have been widely adopted for NLP tasks (lizard book). Neural language models use neural networks as a classifier computing probabilities. They do not require smoothing and are outstanding in generalisation of context and handling histories, often outperforming n_gram models, thus are often expected to reach high performance scores. However, the training time, costs and complexity need to be considered. The prior context in neural models is represented with the use of embeddings, which are advantageous compared to n_grams because they reduce the dimensonality of the data. Pretrained or 'learned from scratch' embeddings may be used, yet due to time constrains, only pretrained embeddings were used and fine-tuned during model training.

Vector Semantics and Embeddings

Vector semantics instantiate distributional hypothesis on meaning similarity of words by learning embeddings from their distributions in tweets. Both static and more dynamic contextualised (e.g. BERT) embeddings were used. This constitutes to representation learning which has been getting much spotlight in the recent NLP research. Vector semantics in general aid in representation of a word in multidimensional semantic space crafted by distributions of neighbouring words. Word and document similarities are commonly calculated by some function of the dot product with cosine, normalised dot product, being the most widely used one. Word embeddings map the word to 1V (vocabulary size) representation, use a neural network hidden layer as a feature extractor to learn semantics from vectors, and further finalise the architecture with a classifier. Word embeddings based on neural networks have recently been widely adopted as an alternative to traditional vectorizers, such as `tf-idf` (?, ?).

Embeddings are short dense vectors, which empirically outperform sparse vectors possibly due to smaller parameter space aiding in overfitting avoidance and generalisation. **GloVe**, short for Global Vectors due to its focus on global corpus statistics, which uses probability ratios of a co-occurence matrix. It is an unsupervised algorithm which acquires vector representations by mapping the words inance being associated with semantic similarity (?, ?). GloVe was implemented in a set of used models, such as CNN, RNN and a combination of them both by transforming each token in a tweet to its vectorized counterpart.
- NNLM 50d, 128d, 128d Normalized (d-dimension) google news - GNews Swivel 20d google news - GloVe: 50, 100, 300d wikipedia ADD WHAT THEY ARE TRAINED ON - DNN

architectures. To do that, embeddings were tried: one was glove (vectorize and set it to a certain output length and then u say how many tokes u want to have) text gets tokeniized up to 2400 tokens (max any tweet had) every token gets a vector with the dimensions of the embedding (50, 100, 300d) tried out different embeddings

you vectorise and set to a certain output length, and specify number of tokens up to 2400 tokens (max of what tweets had), normally 200-400 evrey token in fixed output sequence gets a vector with dimensions of the embedding 600 parameters that are pre-trained, but are fine tuned

Deep Neural Network Architectures

"Language is a sequence that unfolds in time" (p.173). Approaches which do not account for a time dimension assume simultaneous access to inputs, which makes it troublesome to address temporal nature of text and varying sequences. Many language tasks require access to arbitrarily distance information. Recurrent architectures and transformer architectures address these issues and allow to capture temporarity of text, which will be further analysed through the prism of probabilistic models which assign conditional probability to the next token. The traditional n_gram or feed-forward networks, are facing constraints of the Markov assumption (insert), yet the recurrent and transformer architectures manage to use greater contexts weakening this equation.

Recurrent Neural Network (RNN) is a sequential algorithm with cycled connections and dependence of values upon previous inputs. It introduces recurrent neurons which maintain memory state from the past. Thus, the outputs of units are based both on current input as well as previous valued of the hidden layer. RNNs are trained with backpropogation through time and commonly used for sequence classification. However, they face risks of unstable gradients and are challenging to train. Thus, more complex architectures such as Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU) were implemented as those contain gates to control information flows, and are expected to perform better. **LSTM** which is a more complex variant of RNN is able to extract long-term dependencies in word sequences with the use of gates to control information flow within the recurrent cell. **GRU** is simpler than LSTM yet still handles longer sequences than RNN. It combines forget and input gates and returns a state vectot at each input (?, ?).

Convolutional Neural Networks have revolutionised various areas of machine learning, carving their way to becoming a building block for NLP (?, ?). Convolution, the mapping of filter to an input to extract features, results in a feature map, which indicates patterns of detected features in an input. Word embeddings together with CNNs particularly have received wide attention in text classification tasks for Twitter, becoming an obvious choice for this research (?, ?). Specifically when used with pre-trained word embeddings, CNNs show outstanding results, outperforming classical classifiers. According to Goldberg, the easy integration of pre-trained embeddings, ability to pick crucial tokens or sequences of tokens in a manner which is invariant to the inputs, makes CNNs particularly strong in their performance. CNNs are also especially useful in classification of problems where strong local signals of class, which may occur in various input parts, are expected. Such observation is highly relevant for the given task of this research, since the goal is to identify whether certain sequences of words, despite their position, may indicate class membership. CNNs peculiarity is the ability to learn such local indicators in varying locations. Literature suggests the following architecture, which has further been implemented in the workflow: (1) word embedding, (2) convolutional model which learnt to extract important features from word embedded tweets, (3) fully connected model which interprets the features and performs a classification. In a study by (Conneau), the authors prove that deeper architectural approach for NLP tasks is often beneficial. Max_pooling was estimated to outperform other types of pooling, and deeper architectures decreased classification errors, thus being implemented in this work (?, ?).

To perform ESG classification the authors used CNN architecture proposed by Kim2014. Pre-processed tweets were the inputs to the CNN. Pre-trained GloVe embeddings, described above, were used with CNN. Semantic relations in word embeddings aided in identification of semantic similarities among the tweets. ReLU activation function was used due to its empirically superior performance in CNN (?, ?). In the fully connected layer, softmax was used to perform classification. INSERT SOFTMAX DESCRIBE LAYERS OPTIMIZERS REGULARIZATION USED ACTIVATION FUNCTIONS ETC

4.6.3 Transformer Architecture and Bert

Despite the fact that sequential architectures assist in remaining previous information, the problem of relevant information and efficient training still remains significant. Transformers were developed to tackle such shortcomings. In addition to standard network components, transformers use self-attention layers, which allow to capture information from vast context,

mitigating the necessity to transfer such information through recurrent connections. Self attention layer map sequences of input vectors $(x_1, \ldots, x_n)$ to outputs $(y_1, \ldots, y_n)$ of the same length, each item's computation is independent of others. This architecture can compare the item to others in terms of relevance in the given context. The comparison in its simplest essence is computed with score $(x_i, x_j) = x_i \cdot x_j$.
The larger the output, the more similarity vectors share. Then, scores are normalised with softmax to output a weights vector alpha ij. Given proportional scores in alpha, output value is obtained by summing the inputs already seen by the architecture, weighted with corresponding alpha values. $y_i = \sum_{j<=i} \alpha_{ij} x_j$ Thus, the attention-based method involves the following steps: comparison of the relevant items, normalisation to generate probability distribution, and computation of weighted sum of the obtained distribution. mainbook Transformer, according to Vaswani2017 is the first model to rely fully on self-attention while generating input and out representations avoiding RNN or convolution.

Bidirectional Encoder Representation from Transformers (BERT) is a deeply bidirectional pre-trained model architecture, meaning it interprets the text from both sides, as opposed to traditional sequential methods. Transformer, the underlying mechanism of BERT, interprets contextual relationship with the use of encoders and decoders. Encoders capture the input — the sequence of tokens, while decoders build predictions (?, ?). WHAT EXACTLY WAS DONE

4.7 Model complexity analysis

(such as running time) compared to baseline model.
TODO: BERT Types DNN Architectures ACTUAL IMPLEMENTATION 2 other embeddings

5 Results

5.1 Meaningful Facts

(Describe the result with explanation of the reason(s))

5.2 Actionable Insights

(do some Interpretations/predictions from the result)

5.3 Valuable Outcomes

(Applicability of the result) (Hints – How this information can be used to make some recommendations)

Add feature importance, what are the most common words (word cloud), XGBoost feature importance

6 Discussion

Answers to the Research Question(s)

6.1 Implications for Research / Learning Reflections

6.2 Limitations of the dataset/work

(if any)

Due to noisy nature of Tweets, training the embeddings would possibly deem better results, yet it was infeasible due to time constraints.

ESG scores do not stay the same over time, even though foundational ESG performance historically does not alter much. However, the tweets should still represent the underlying patterns.....

7 Conclusion Future Work

Limitations / Future work: sample only from the US, in the future we should work with small and medium-sized firms.

8 Appendix