LUND UNIVERSITY

SCHOOL OF ECONOMICS AND MANAGEMENT

STAH11: Statistik: Kandidatuppsats HT23

06/01/2024

Comparison of VADER and Pre-Trained RoBERTa

A Sentiment Analysis Application

Authors:

Linda Erwe

Xin Wang

Supervisor:

Jakob Bergman

Abstract

Thesis title: Comparison of VADER and Pre-Trained RoBERTa

Seminardate: 11−12/01/2024

Course: STAH11, Degree Project Undergraduate level, Statistics, Undergraduate level, 15 University Credits Points (UPC).

Authors: Linda Erwe & Xin Wang

Supervisor: Jakob Bergman

Keywords: sentiment analysis, natural language processing, BERT, VADER, sustainability report

Purpose: The purpose of this study is to examine how the overall sentiment results from VADER and a pre-trained RoBERTa model differ. The study investigates potential differences in terms of the median and shape of the two distributions.

Data: The sustainability reports of 50 independent random companies are selected as the sample. The number of non-responses is 6, which means that the reports of 44 companies are included in the study. Furthermore, the total number of paragraphs in the investigated sample is 320. The number of words per paragraph ranges from 16 to 234.

Methods: VADER is a dictionary- and rule-based sentiment analyzer built on a combination of five heuristics and a dictionary of words that connects lexical features to sentiment intensity. The model is accessed through the NLTK library in Python. The algorithm provides four numbers: positive, neutral, negative and a compound score. RoBERTa is a variation of the BERT model, which is based on transformers and a concept called self-attention to be able to associate words with other words in order to understand context. A pre-trained version of the model is utilized in this study. The model provides three values: positive, neutral and negative. A fourth overall sentiment score is computed for comparison to VADER's compound score.

Results: A two-sample Kolmogorov-Smirnov test shows that the two scores are drawn from different distributions. Furthermore, a Wilcoxon signed-rank test shows that the median of the differences between the VADER compound score and the RoBERTa polarity score are not zero. In other words, there is a difference in location. The final general conclusion is that there is a difference between the scores both when considering location and shape combined and when only considering location.

Table of Contents

List of Abbreviations

Abbreviation	Definition
AI	Artificial Intelligence
BERT	Bidirectional Encoder Representations from Transformers
CEO	Chief Executive Officer
ML	Machine Learning
NLP	Natural Language Processing
NLTK	The Natural Language Toolkit
RoBERTa	Robustly Optimized BERT Pre-training Approach
VADER	Valence Aware Dictionary and sEntiment Reasoner

1 Introduction

In this section an introduction to the topic of natural language processing is presented, as well as a problematization that results in the purpose and research question to be investigated in this paper.

1.1 Background

Imagine an artificial intelligence (AI) that is able to thoroughly understand written human language in all its complexity. An AI that can understand humor and emotion in a text. Not that many years ago, this was unthinkable. However, today it is reality (e.g. Chakriswaran et al., 2019). The development in the area of natural language processing (NLP) is rapid. There was a breakthrough after the introduction of an architecture called Transformers by Vaswani et al. in 2017 and since then, the development has been very quick (Queipo et al., 2022; Xiao & Zhu, 2023). For example, the first version of ChatGPT was introduced in 2018 and the latest version, introduced in 2023, is able to interpret prompts and perform complex language-related tasks (Marr, 2023). The evolution of ChatGPT is just one of many examples of the rapid progress in the field of AI and NLP.

The progress in the field enables analyses that were impossible before. For example, it has been difficult to quantitatively analyze text due to its characteristics of high-dimensionality and sparsity in data (i.e. high occurrence of zero elements in the data) (Johnstone & Titterington, 2009; Akritidis et al., 2020; Rahnenführer et al., 2023). However, the development in the field of machine learning (ML) and NLP, and specifically in Transformers, entails new possibilities. Due to the progress, new models have emerged, such as the Bidirectional Encoder Representations from Transformers (BERT) model. BERT was introduced in 2018 by researchers at Google (Devlin et al., 2018). What makes this model different from others is that it does not only look at text from left to right or from right to left, but instead simultaneously interprets text from both directions. This is possible due to Transformers (Devlin et al., 2018). An important part of Transformer models is that they contain special layers. These are called attention layers (Vaswani et al., 2017). They teach the model to concentrate on specific elements within the input data

while ignoring others to enhance problem-solving efficiency (Vaswani et al., 2017). When humans read text, they give greater attention to certain parts. Typically, our focus is drawn to the who, when, and where aspects of a sentence. This is what the Transformer based models try to imitate.

Today there are many different versions of BERT. Some are improved or light versions and some are pre-trained on a specific domain, such as finance (FinBERT). One of the more well-known versions is Robustly Optimized BERT Pre-training Approach (RoBERTa), which is an improved version of BERT that is trained on a much larger data set and that can match or exceed the performance of previous BERT methods (Liu et al., 2019).

Today BERT and its variations are considered the state-of-the-art model in the field of NLP, achieving higher accuracy than other models (Mishev et al. 2020; Chernyavskiy, Ilvovsky & Nakov, 2021; Garrido-Merchán, Gozalo-Brizuela & González-Carvajal, 2023).

1.2 Problematization

An aspect of text that can be analyzed with NLP methods is the sentiment. Sentiment analysis is mainly used to mine the emotions and opinions of text in order to identify to which degree a sentence or paragraph is positive, negative or neutral (Hardeniya & Borikar, 2016). There are primarily two different approaches to sentiment analysis, namely ML and dictionary based approaches (Hardeniya & Borikar, 2016). The first mentioned can be considered a more advanced or complex approach, where the models need access to large data sets in order to be trained. More specifically, the linguistic content of the texts is utilized to train a classifier (Rice & Zorn, 2021). This trained classifier is then applied to assess the sentiment of the remaining cases (Rice & Zorn, 2021). The learning process can be supervised, where pre-tagged texts are used to train the model, or unsupervised, where the models are equipped with intelligence to learn and discover by themselves. One example of an approach that is categorized as an ML approach is BERT (Garrido-Merchán, Gozalo-Brizuela & González-Carvajal, 2023). Dictionary based approaches, or lexicon based approaches, are approaches that use a dictionary containing opinion words in order to match words in the data being investigated (Hardeniya & Borikar, 2016). Since most existing dictionaries list synonyms and antonyms for each word, a simple

technique is to use a rather small number of seed sentiment words to bootstrap based on the synonyms and antonyms (Bhonde et al., 2015). A common procedure is the following: the seed sentiment words with known positive or negative orientations are first collected manually. Then the algorithm increases the seed list by searching for synonyms and antonyms in online dictionaries. This iterative process stops when no more new words can be found and the dictionary is then completed (Bhonde et al., 2015). The main advantages of the dictionary based approach are that it is easy to implement and no training is required (e.g. Iqbal, Karim & Kamiran, 2015). However, the disadvantages are that context is not taken into account, so it may result in lower accuracy, and there could be an issue in some domains which use specific terms that might not be covered in the dictionary (Dhanalakshmi et al., 2023).

Many applications of sentiment analysis so far have been on different types of customer reviews, such as product and restaurant reviews, where the sentiment is reflected in a star rating that can be connected to the written review and that can be used as a comparison to the sentiment predictions from a model (e.g. Yang et al., 2020; Ligthart, Catal & Tekinerdogan, 2021; Kumar et al., 2021; Onan, 2020; Başarslan & Kayaalp, 2021). The model can then, for example, be used in order to predict a rating based only on the review. The star rating is then the dependent variable and the text constitutes the independent variables. Many people, both researchers and others, have compared the performance of different NLP models on different types of reviews. The result has been that the state of the art model, BERT and its variations, in terms of accuracy outperformed other models, both machine learning approaches and dictionary-based models, like the Valence Aware Dictionary for sEntiment Reasoning (VADER) model (Sayeed, Mohan & Muthu, 2023; Başarslan & Kayaalp, 2021; Catelli, Pelosi & Esposito, 2022).

In the above mentioned application setting, the sentiment is usually rather direct (the reviewers say what they think and are often quite clear about it) and there is a predefined label in terms of a rating. However, this is not the case with more complex texts. It can, for example, be assumed to be more difficult to assess text in public documents published by companies. They have an incentive to portray a nice picture of themselves, in order to maintain or obtain legitimacy. Therefore, they might write in a more nuanced way about potential downsides or failures and potentially try to turn something negative into something positive. Moreover, they might write

that they have set a new and ambitious goal. This may not be as positive as it looks at first glance. Since it is ambitious, it means it is challenging, and this in turn means that they still have much to improve. Thus, it is interesting to investigate whether two different models, namely VADER and a pre-trained RoBERTa model, produce the same results in a complex setting, or if they differ from each other. This thesis is therefore aimed at investigating this through an application on the Chief Executive Officer (CEO) letter or introduction in sustainability reports. Here, companies can write about the past, present and future sustainability performance of the company, about their view on sustainability issues related to their business and other things they deem important in relation to sustainability.

The purpose of this thesis is to compare and analyze the differences in the results from two different NLP models, more specifically, VADER and a pre-trained RoBERTa model, for sentiment analysis, when applied on each of the paragraphs in the introduction of sustainability reports.

The question to be addressed in order to fulfill the purpose is:
"How do the overall results from the VADER and RoBERTa model for sentiment analysis differ when applied on the introduction letter of sustainability reports?". This main question is divided into two specific sub questions:

1. "Do the overall results for each of the two models follow different distributions (i.e. shape and location)?"
2. "Do the overall results for each of the two models differ in terms of the central value of their distributions (i.e. location)?"

2 Data

This section starts with a presentation of the sample selection and the data collection procedure. Thereafter, descriptive information about the data is presented.

2.1 Data Collection

In order to sample the sustainability reports to be studied, all companies listed on Nasdaq Stockholm as of September 12 2023 were copied into an excel file in alphabetical order. Duplicates, such as Volvo A and Volvo B, were deleted in order for each company to only be represented one single time. This resulted in a list of 362 companies. Then, simple random sampling was used to choose 50 companies. This method was utilized in order to give all companies an equal chance to be included in the sample. In this way, we obtain a sample that should represent the population of companies listed on Nasdaq Stockholm quite well. In other words, the sample includes a mix of large, medium and small sized companies and companies active in various industries.

The next step in the data collection process was to visit each company's website and gather their sustainability report, or their integrated annual and sustainability report. During this step, it was discovered that there were problems retrieving some of the reports. Some companies only published their reports in Swedish, some did not publish a sustainability report or integrated report and some did not yet publish a sustainability report for the year 2022. The total number of non-response was 6, which meant that the sustainability reports of 44 companies were considered our investigable sample (see appendix A).

The paragraphs were copied into an excel file, where each row represented one company and each column represented one paragraph. After each collected paragraph, we checked for and removed hyphens that cut off words because of newlines. Moreover, we removed any extra line breaks that often automatically come with pasting paragraphs consisting of several lines of text. Since both of these procedures had an effect on the results, they were important.

2.2 Descriptives

As mentioned above, the sampled data consists of 44 introductions or CEO letters of sustainability reports for the year 2022 published by different companies listed on Nasdaq Stockholm. The most paragraph-rich report consists of 22 paragraphs and the lowest number of paragraphs of the studied reports is 2. The average number of paragraphs of our studied companies is 6.6. A histogram of the number of paragraphs per sustainability report is shown in figure 1. It can be seen that most of the sustainability reports have an introduction or CEO letter consisting of about 4-5 paragraphs and that there are two outliers that have 21 and 22 paragraphs.

The total number of paragraphs is 321. The number of words per paragraph ranges from 16 to 299 and the average number of words per paragraph is about 69. Figure 2 shows that there is one paragraph, which has more than 250 words, that can be considered an outlier. Since 299 words is very many for the models to analyze in one sequence, especially for VADER which is not suited for long texts, as will be seen later, we choose to remove this outlier paragraph from the dataset. After removal, the average number of words per paragraph is 68 and the number of words per paragraph ranges from 16 to 234. The total number of studied paragraphs is 320.

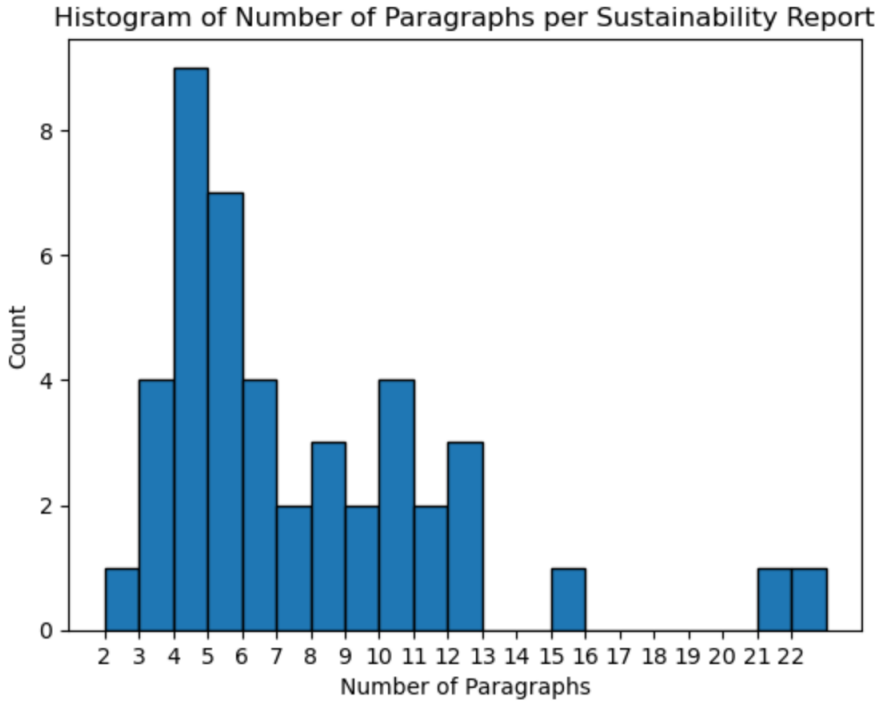

Figure 1: Histogram of the number of paragraphs per sustainability report.

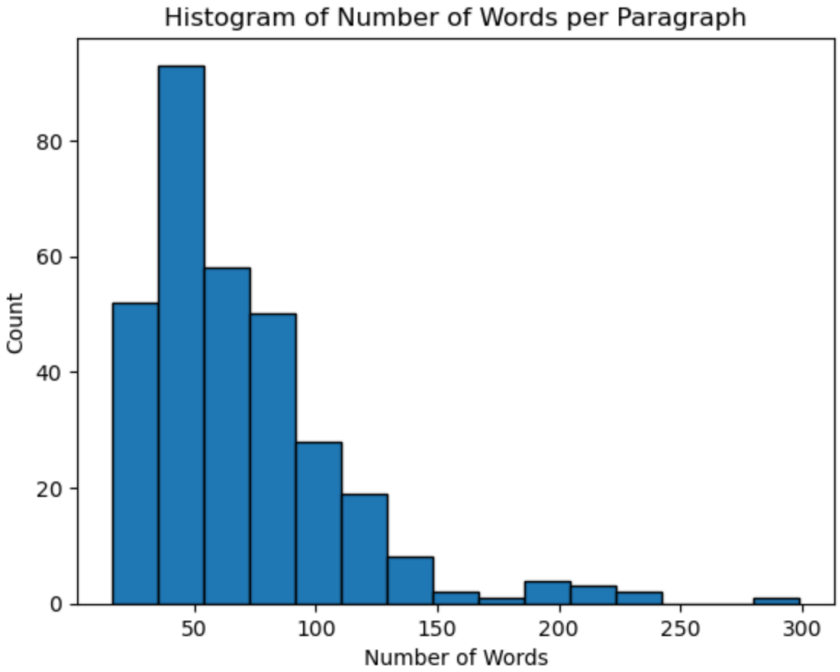

Figure 2: Histogram of the number of words per paragraph.

A word cloud, i.e. a graphical representation of the most frequently used words in a text, is presented in figure 3 for the investigated paragraphs. The most frequently used word in the studied paragraphs is *"sustainability"*. Other commonly used words are *"sustainable"*, *"business"* and *"employee"*. Words clearly connected to sentiment such as *"good"* and *"better"* are also used rather frequently.

Figure 3: Word cloud showing the most frequently used words in the paragraphs.

3 Methods and Results

In this section, a presentation of the methods and results are displayed. The method and results of the VADER method are presented first. Thereafter, the RoBERTa method and its applied results are covered.

3.1 VADER

The greatest advantage of VADER is that it is easy to use since it does not require any training data (Iqbal, Karim & Kamiran, 2015). However, this can also be seen as a disadvantage, as it means that it cannot learn and could therefore, in some cases, lead to less accurate results, since a certain word may be more or less positive based on the context of the text (Dhanalakshmi et al., 2023). One can argue that there is always a balance of user-friendliness versus accuracy, where VADER positions itself more towards the first mentioned. Nevertheless, it has been shown that VADER produces accurate results in domains such as social media posts and movie reviews (Elbagir & Yang, 2019; Bonta, Kumaresh & Janardhan, 2019).

VADER was specifically created to work with text produced in a social media setting, however, it generalizes favorably to other contexts (Hutto & Gilbert, 2014). Furthermore, it was created using a combination of a qualitative and quantitative approach. The quantitative analysis was primarily used for empirical validation and experimental investigations. The qualitative analysis was used to identify characteristics of text that influence the perceived sentiment intensity. The result of the qualitative analysis resulted in the creation of five rules upon which the model is based (Hutto & Gilbert, 2014). Thus, VADER is a dictionary- and rule-based sentiment analyzer, meaning that the system automatically scores input text using a predefined set of rules (Bonta, Kumaresh & Janardhan, 2019). As the creators, Hutto and Gilbert (2014), state, the five rules, or heuristics, of VADER are:

1. Punctuation in the form of exclamation marks increases the magnitude of the intensity without modifying the semantic orientation. For instance, *"This is great!!"* is more intense than *"This is great."*, but both are positive in semantic orientation.

2. Capitalization, specifically using ALL-CAPS to highlight a sentiment-relevant word among other, non-capitalized, words, amplifies the intensity of the sentiment without affecting the semantic orientation. For example, *"This is GREAT!"* is more intense than *"This is great!"*.
3. Degree modifiers (i.e. intensifiers, booster words, or degree adverbs) have an impact on sentiment intensity. They can either increase or decrease the intensity. For example, *"This is exceptionally good"* compared to *"This is good"* increases the intensity, whereas *"This is marginally good"* reduces the intensity.
4. The word *"but"* is of special importance, since it signals a shift in sentiment polarity, where the sentiment of the text after the conjunction is dominant. For instance, the sentence *"The product is great, but the delivery is horrible"* contains mixed sentiment, with the latter half dictating the overall rating.
5. Analyzing the tri-gram (i.e. sequences of three consecutive words) that precedes a sentiment-loaded lexical feature enables the algorithm to identify nearly 90 % of instances in which negation switches the polarity of the text. *"The product isn't really that great"* is an example of a negated sentence.

VADER combines these five heuristics with a dictionary, which maps lexical features (i.e. anything that we use to communicate text, such as words or acronyms) to sentiment intensity. Moreover, VADER relied on the wisdom of the crowd rather than individual opinion when deciding on the sentiment intensity of lexical features. This means that instead of asking a few experts how they would rate a word, 10 independent human raters were asked and their ratings were then averaged for each word in the dictionary. This prevents arbitrary ratings due to personal or individual perceptions. The individual words in the dictionary have a sentiment score, called valence score, between −4 to 4, where −4 is the most negative and 4 is the most positive. 0 represents a neutral sentiment. For instance, the word *"great"* has a valence score of 3.1 (Hutto & Gilbert, 2014).

In order to analyze the sentiment by using VADER by Hutto and Gilbert (2014), we use a library for Python called The Natural Language Toolkit (NLTK) (NLTK Project, 2023a). From this library, we can download the VADER lexicon. In order to use VADER for sentiment analysis, we

also need to import the `SentimentIntensityAnalyzer` class. The `SentimentIntensityAnalyzer` provides a method called `polarity_scores()` that accepts a text input and returns a dictionary that includes sentiment scores corresponding to the given text (NLTK Project, 2023b). The algorithm compares each word in the text to the sentiment dictionary of words with information about their positive or negative semantic meaning. Words such as *"the"*, *"a"*, *"they"*, etc., are disregarded since they are not contributing to any information about the sentiment of the text.

The function `polarity_scores()` produces four different scores (NLTK Project, 2023b). First, it produces three sentiment values, with their own distribution, between 0-1 of how positive, neutral and negative the paragraph is. These numbers are interpreted as percentages of positive, negative and neutral sentiment features (Bird, Klein & Loper, 2009). A paragraph that is only positive would for example have the following scores: `{'neg': 0.0, 'neu': 0.0, 'pos': 1.0}`. Since they represent percentages, the scores can take any value in between 0 and 1 as long as the sum of the three scores equals 1. Last, the function produces a compound score. This score is an overall sentiment score and it ranges between −1 to 1, where 1 is the most positive and −1 the most negative. If the compound score is negative, it means that the paragraph leans more toward a negative sentiment than a positive (Bird, Klein & Loper, 2009).

The compound score is calculated by scanning the text for known sentiment features, modifying the intensity and polarity in accordance with the five heuristics, summing together these scores and lastly normalizing the score using the following function:

$$\frac{x}{\sqrt{x^2 + \alpha}} \tag{1}$$

where x is the sum of the valence scores and alpha is a normalization parameter that is set to 15 by default (Bird, Klein & Loper, 2009). A plot of the function is shown in figure 4. It is seen that the maximum value of the function is 1 and the minimum is −1. An important observation is also

that as the sum of the valence scores, x, increases or decreases, the compound score tends to become closer and closer to 1 or −1. This means that if there are many words that are included in the dictionary of VADER in the text to be investigated, the score is more likely to be close to the maximum or minimum value. Therefore, VADER works better on shorter texts compared to large documents.

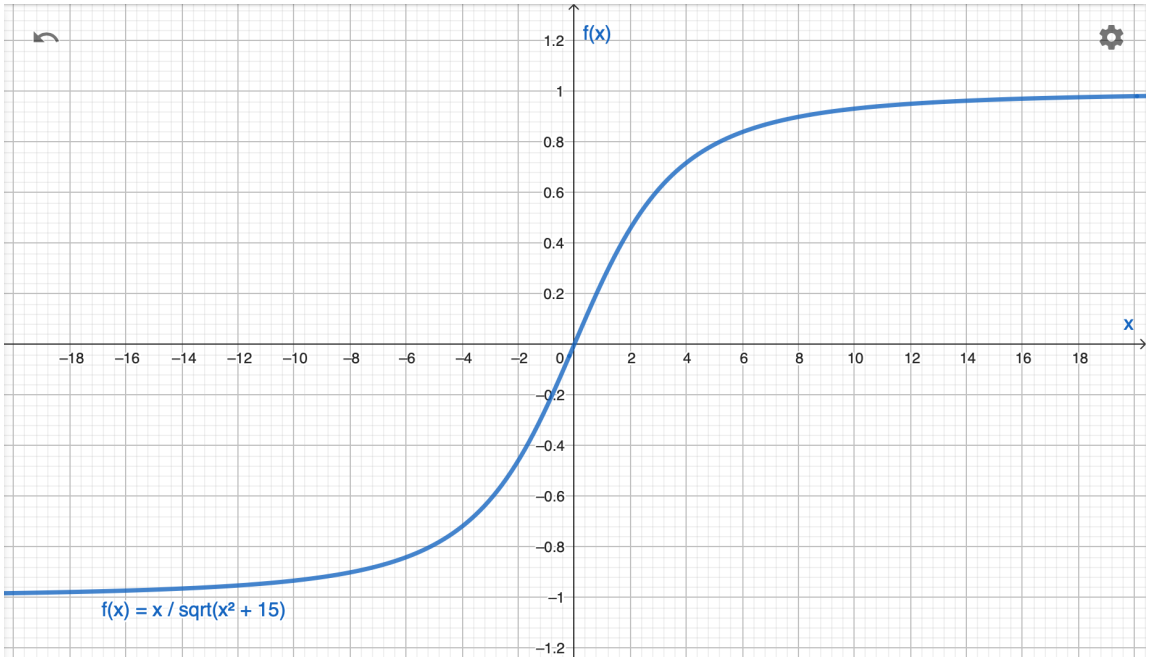

Figure 4: Graph of the normalization function for the compound scores.

Running the function on the collected paragraphs, we obtain results in the following format, for each and every sustainability report:

Table 1: Example of output from VADER method.

Paragraph index	Negative	Neutral	Positive	Compound
0	0.062	0.844	0.094	0.176
1	0.096	0.718	0.186	0.691
2	0.000	0.886	0.114	0.612
3	0.000	0.933	0.067	0.440
4	0.000	0.920	0.080	0.671
5	0.009	0.812	0.179	0.963
6	0.000	0.809	0.191	0.947
7	0.000	0.698	0.302	0.942
8	0.000	0.827	0.173	0.891
9	0.000	0.783	0.217	0.818
10	0.000	0.788	0.212	0.859
11	0.031	0.864	0.105	0.527

The example is typical for the dataset. The index of 0-11 represents the paragraphs and the index number on the last row of every report is the number of paragraphs in the introduction of the sustainability report of the particular company minus 1. In the case of the sustainability report portrayed in table 1, we see that all of the paragraphs are more towards the positive side than the negative, as the compound scores are positive. The first paragraph, however, has a considerably lower compound score than the rest of the paragraphs and is therefore not as positive.

Of all investigated paragraphs, the highest compound score is 0.991 (negative: 0.008, neutral: 0.813, positive: 0.179) and it is connected to the following paragraph:

> We continually build and develop upon our regulatory compliance, information security and platform management efforts to provide our partners and the communities in which they operate in, with the safest, most secure and innovative service, including AML, Player Protection, Safer Gambling, ISO 27001 and ISO 20000 certificates. Our climate

> action includes partnering with Greenly to accurately measure, manage and inturn reduce our Scope 1, 2 and 3 GHG emissions. We have also stopped gifting merchandise, as the emissions produced impact the climate negatively, and are now offering well-being top-ups, a Reward Toolkit for Managers and their teams, and we will shortly provide truly sustainable welcome gifts which offset carbon emissions around the world, including tree planting or donation to biodiverse projects through Switzerland based The Gold Standard's organisation "for a climate secure and sustainable world". We have also offset 167,000kg Co2 (100% of our recorded business travel footprint) through The Gold Standards Verified Emission Reductions scheme. In 2022 our People team collected, compiled and addressed over 63,000 feedback points through annual and monthly Voice of Employee and engagement surveys, to improve the employee experience, and developed a new and improved perks and benefits package. The team also established GiG's first DEI allyship called GiG Allies, with full training and certifications achieved by all members. (Gaming Innovation Group Inc, 2022, p. 3).

This is a quite long paragraph with very few negative words, many neutral words and some positive words, as represented by the negative, neutral and positive scores. Moreover, the most negative compound score is -0.917 (negative: 0.123, neutral: 0.859, positive: 0.017) and it is connected to this paragraph:

> Underground infrastructure in Norva24's current markets are generally in poor condition after decades of delayed renovations resulting in a general investment backlog within the underground infrastructure. Across the current markets, the average age of the underground infrastructure is approximately 40 years and, in some cases, as old as 150 years. This leads to an increase of damage in the sewers and leakage rates, which affects the reliability and quality of the overall underground infrastructure and affects the whole society in a negative way. Additionally, increasing urbanization is putting a strain on the capacity of the underground infrastructure that was not dimensioned for the current population increases in larger cities. The poor state of the underground infrastructure increases the need for maintenance and renovation. (Norva24, 2022, p.42).

There are very few positive words in this paragraph, many neutral words and some negative words. The compound score is very low and rather close to the lowest possible value.

The arithmetic mean of the compound scores for all paragraphs is about 0.61, where 1 is the most positive and −1 is the most negative possible. Furthermore, the variance is 0.15.

3.2 RoBERTa

RoBERTa is an improved version of the BERT model (Liu et al., 2019). Thus, in order to explain the RoBERTa model, we first need to explain BERT. Neither BERT nor RoBERTa are specifically made for sentiment analysis, but since they are pre-trained on large amounts of unlabelled text, they can be fine-tuned to perform the downstream task of sentiment analysis (Devlin et al., 2018; Liu et al., 2019).

There are two established approaches for employing pre-trained language representations in downstream tasks, more specifically, feature-based and fine-tuning (Devlin et al., 2018). The feature-based approach involves utilizing task-specific architectures that incorporate pre-trained representations as additional features, whereas the fine-tuning approach uses minimal task-specific parameters and is trained by fine-tuning all pre-trained parameters. However, both use the same objective function in the pre-training phase, where they base their learning of general language representations by using unidirectional (i.e. operating in a single direction, for example reading text from left to right) models. The developers of BERT, Devlin et al. (2018), saw this as a potential disadvantage as it may restrict the power of the pre-trained representations, especially for fine-tuning approaches. BERT stands for Bidirectional Encoder Representations from Transformers. The first descriptive word for the model is bidirectional, which means that BERT interprets text both from left to right and from right to left at the same time. This is accomplished through the use of a “masked-language-model” (MLM) pre-training objective. This model randomly masks out some of the tokens of the input and then tries to predict the original token of the masked word based on the context both at the left and right side of the masked word at the same time. Word embeddings are representations of words through high-dimensional real-valued vectors. BERT uses WordPiece embeddings with a 30,000 tokens vocabulary (Devlin et al., 2018). WordPiece is a subword tokenization algorithm employed for NLP tasks that divides words into smaller units known as subword tokens (Wu et al., 2016).

There are two general steps in the BERT model. First, the pre-training phase, where the model is trained on unlabeled data over different pre-training tasks (Devlin et al., 2018). Second, the fine-tuning phase, where BERT is first initialized with pre-trained parameters. Then all these parameters undergo fine-tuning using labeled data from downstream tasks. Each downstream task has its own separate fine-tuned model, although they share the same pre-trained parameters during initialization (Devlin et al., 2018).

The pre-training phase uses two unsupervised tasks, namely the MLM, which was briefly described above, and next sentence prediction (NSP) (Devlin et al., 2018). The MLM is used to train a deep bidirectional representation by masking 15 % of the tokens at random with `[MASK]` tokens and then predicting them. The mask tokens constitute hidden vectors. The second unsupervised task, NSP, is needed in order to be able to perform downstream tasks such as question answering and is based on understanding the relationship between two sentences (Devlin et al., 2018).

Fine-tuning is achieved through the self-attention mechanism in the Transformer, introduced by Vaswani et al. (2017) (Devlin et al., 2018). For each downstream task, you can insert the task-specific inputs and outputs into BERT and fine-tune all the parameters end-to-end (Devlin et al., 2018). The preceding sequence transduction models (i.e. models that convert input sequences into output sequences) before Transformer was introduced were based on complex recurrent or convolutional neural networks that included an encoder and a decoder (Vaswani et al., 2017). The best of these models also included an attention mechanism to connect the encoder and decoder. The network architecture Transformer is only based on attention mechanisms and has no recurrence or convolutions (Vaswani et al., 2017). Thus, the attention, or more specifically the self-attention, mechanism can be said to be the core of the Transformer. However, before describing self-attention, presenting an overview of the architecture of the transformer model is necessary (see figure 5).

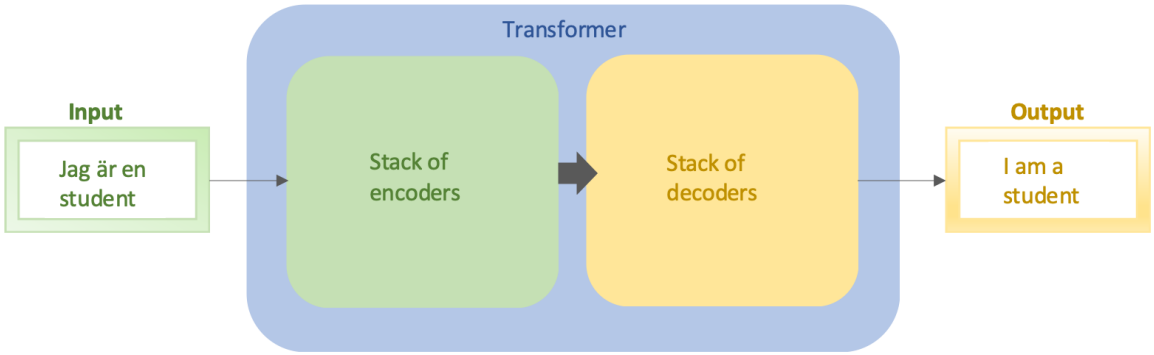

Figure 5: Illustration of the Transformer architecture.

The Transformer consists of a stack of originally 6 layers of encoders and 6 layers of decoders (Vaswani et al., 2017). The input sentence is sent to the encoders that process it and generate an encoded representation of it. The final result from the encoders is passed further to each of the decoders that are responsible for generating the output. Figure 5 illustrates the simple example of translation. In the case of sentiment analysis, the outputs are instead sentiment scores. Each encoder layer has two sublayers and each decoder layer has three sublayers as illustrated in figure 6. The encoders have a self-attention layer and a feed forward neural network layer. The self-attention layer facilitates the encoder to consider other words in the input sentence while encoding a specific word. The results of the self-attention layers are passed onto the feed forward layer. The identical feed forward layer is employed at each position. These feed forward layers serve the purpose of processing the output from the self-attention sublayers to be more suitable as input for the self-attention in the next encoder or decoder layer. The only difference between the encoder and decoder layers is that the decoder layers have an extra sublayer, called the encoder-decoder attention layer, which assists in the decoder's focus on relevant parts of the input sentence (Vaswani et al., 2017).

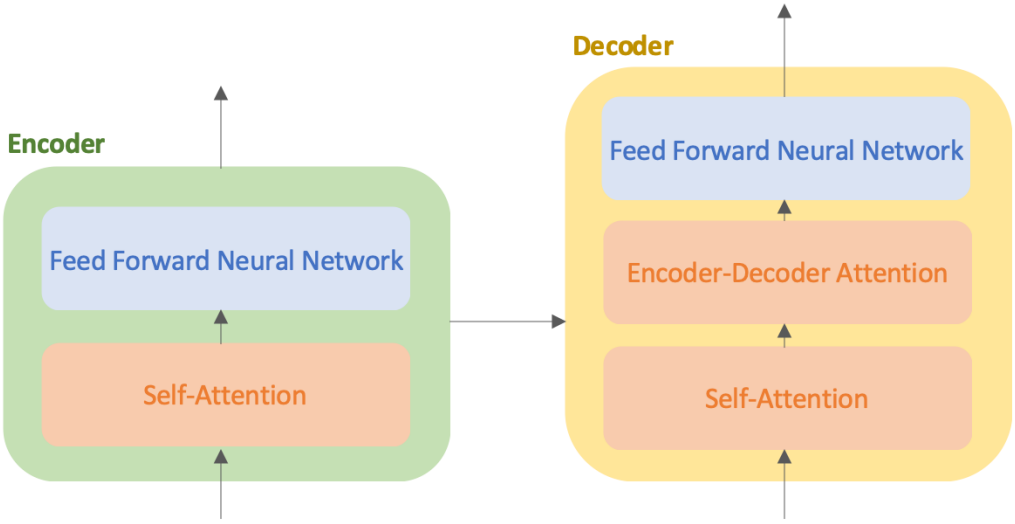

Figure 6: Illustration of the encoder and decoder sublayers in the Transformer architecture.

Each input word is turned into a vector by the application of an embedding algorithm (Vaswani et al., 2017). This takes place only at the most bottom encoder in the stack. All encoders receive a list of vectors, each of them with size 512. For the most bottom encoder, this is the word embeddings and for the other encoders this is the output of the previous encoder below. The size of the list is a hyperparameter that can be defined, typically corresponding to the length of the longest sentence in the training dataset (Vaswani et al., 2017).

Self-attention is connected to the ability to associate certain words with another word when making predictions (Vaswani et al., 2017). For instance, *"it"* in the sentence *"The food was not eaten for lunch because it had gone bad"* refers to *"the food"*. For humans, this is obvious, but for a computer it might as well have referred to *"lunch"*. With the use of self-attention, as the model processes each word, it analyzes other words in the input sequence that can lead to a better encoding of the word. Self-attention is therefore a way to include the whole context of the input sequence in order to improve predictions. Figure 7 illustrates the previously mentioned example. The darker the color, the higher the attention weight is. In this case, it can be seen that *"it"* relates the most to *"food"*. The visualization only shows one attention mechanism in the model. In BERT, multiple of these mechanisms work in parallel in order to encompass a more extensive spectrum of relationships between words.

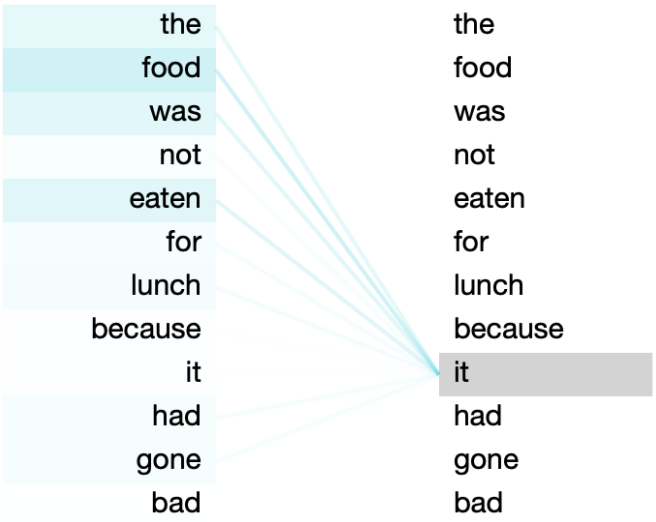

Figure 7: An illustrative example of the self-attention mechanism (visualization using BertViz).

The decoder layers operate in the same general manner as the encoder layers (Vaswani et al., 2017). The difference between the two separate attention sublayers in the decoder layers is that in the "encoder-decoder attention" sublayers, the queries are passed from the previous decoder layer, and the memory keys and values come from the output of the encoder. The output of the decoder is not a word, but a real-valued vector. Thus, a final linear layer and a softmax layer are added. The linear layer projects the results vector from the stack of decoders into a larger vector where each cell reflects the score of a unique word and the softmax turns the scores into probabilities. The cell connected to the highest probability is then chosen and the word connected to it is produced as the output (Vaswani et al., 2017).

Around each sublayer of the encoder and decoder, a residual connection is employed (Vaswani et al., 2017). Moreover, there is a layer normalization step. Additionally, due to the fact that the model does not include recurrence or convolution, it takes the order of the sequence into account by adding positional encodings to the input embeddings at the very bottom of both the encoder and decoder stacks (Vaswani et al., 2017).

Now we return to the RoBERTa model, introduced by Liu et al. (2019). The main differences between BERT and RoBERTa are that in RoBERTa:

1. the model has been trained longer, with larger batches, with more data,

2. the next sentence prediction (NSP) objective has been removed,
3. training is done on longer sequences and
4. the masking pattern applied to the training data has been changed to become more dynamic (Liu et al., 2019).

First, even the longest trained model in the experiments by Liu et al. (2019) does not seem to overfit the data. Additional training was therefore thought to be beneficial for the model. Second, the NSP was thought to be an important part in training the BERT model for downstream tasks. However, the developers of the RoBERTa model discovered that removing the NSP matched or even slightly improved the downstream task performance. Third, Liu et al. (2019) find through their experiments that using individual sentences for training harms performance on downstream tasks. Therefore, RoBERTa employs training on longer sequences. Last, the original BERT implementation applied masking only once during data preprocessing, leading to a single fixed mask. RoBERTa instead uses a dynamic mask, where the masking pattern is produced every time a sequence is given to the model (Liu et al., 2019).

One of the most used pre-trained RoBERTa models is trained on about 58 million tweets and fine tuned for sentiment analysis with the TweetEval benchmark (Barbieri et al., 2020). TweetEval is a standardized test bed for seven tweet classification tasks, where sentiment analysis is one of them (Barbieri et al., 2020). In 2022, a new version of RoBERTa, that is trained on a larger corpus of tweets was released (Camacho-Collados et al., 2022). The new model is trained on about 124 million tweets from January 2018 to December 2021 and is, similarly to the previous model, fine tuned for sentiment analysis with the TweetEval benchmark (Camacho-Collados et al., 2022). In this thesis, we use the new version, since it is trained on a larger amount of data.

The new pre-trained RoBERTa model, based on Twitter posts, only provides three scores as output. It provides positive, neutral and negative sentiment values. In other words, the model does not automatically provide a compound score that indicates the overall sentiment of a paragraph, like the VADER compound. Therefore, we have to calculate a sentiment score that indicates the overall sentiment of a paragraph for the RoBERTa model. We name it “RoBERTa polarity score” and it is calculated as follows. Each probability (i.e. `pos, neu, neg`) is

multiplied by its polarity weight (−1 negative, 0 neutral, 1 positive) and then summed. The sum is then passed through the hyperbolic tangent function, which scales the values from −1 to 1, similar to the normalization in the VADER compound. It is important to note that the compound and the RoBERTa polarity score are based on different considerations made by the models. Nevertheless, they are both sentiment values that indicate the overall sentiment of a paragraph, with negative values meaning a negative sentiment and positive values representing a positive sentiment.

Table 2 shows a typical example of the results from the pre-trained RoBERTa model for one of the studied sustainability reports, the same one as shown in table 1 for the VADER model. We see that all paragraphs are assessed as positive, since the RoBERTa polarity scores are positive. The third paragraph is, however, very close to neutral, with a compound score of about 0.05.

Table 2: Example of output from pre-trained RoBERTa model and computed RoBERTa polarity score.

Paragraph index	**Negative**	**Neutral**	**Positive**	**RoBERTa polarity score**
0	0.013	0.307	0.680	0.583
1	0.011	0.262	0.728	0.615
2	0.067	0.262	0.119	0.052
3	0.028	0.840	0.132	0.104
4	0.021	0.638	0.341	0.310
5	0.018	0.587	0.395	0.360
6	0.003	0.090	0.907	0.718
7	0.003	0.043	0.954	0.740
8	0.006	0.186	0.808	0.665
9	0.005	0.079	0.916	0.721
10	0.039	0.342	0.618	0.522
11	0.008	0.686	0.306	0.290

The highest RoBERTa polarity score of all paragraphs is about 0.753 and it is connected to the following paragraph:

> We love winning. We are entrepreneurial, innovative, and always on our toes, hungry for more. Our customer's success is our success. (Evolution, 2022, p.31).

The context of the paragraph is that it is describing the company's values. The short paragraph is describing the value that they name "ALIVE". The paragraph is connected to a negative score of 0.003, neutral score of 0.013, and positive score of 0.983. The most negative RoBERTa polarity score is −0.653 (negative: 0.797, neutral: 0.187, positive: 0.016) and it is connected to this paragraph:

> Many investments regarding sustainability and ESG have unfortunately failed and created paper products. On the positive side, for some, it has generated work for a large consulting industry, but unfortunately much of this consulting work has not made much difference to long-term sustainability. (NP3 Fastigheter, 2022, p.44).

This result is interesting since although the text mentions "on the positive side", the score for positive according to the pre-trained RoBERTa model is very low. This is an illustration of the fact that the model takes the context into account, because later it says "but unfortunately…" which shifts the sentiment.

The arithmetic mean of the RoBERTa polarity score for all paragraphs is about 0.43, which means that the average paragraph is positive. The variance is 0.08.

3.3 Comparison

A boxplot of the VADER compound and RoBERTa polarity score is presented in figure 8. It can be seen that the polarity scores are in general lower than the compound values. Additionally, there is less variation in the RoBERTa polarity scores than in the VADER compounds since the box is more compact and the bottom whisker is shorter. The compound values have a median of about 0.77, a minimum of −0.92, a first quantile of 0.41, a third quantile of 0.90 and a maximum

of 0.99. The RoBERTa polarity scores have a median of about 0.52, a minimum of −0.65, a first quantile of 0.27, a third quantile of 0.66 and a maximum of 0.75.

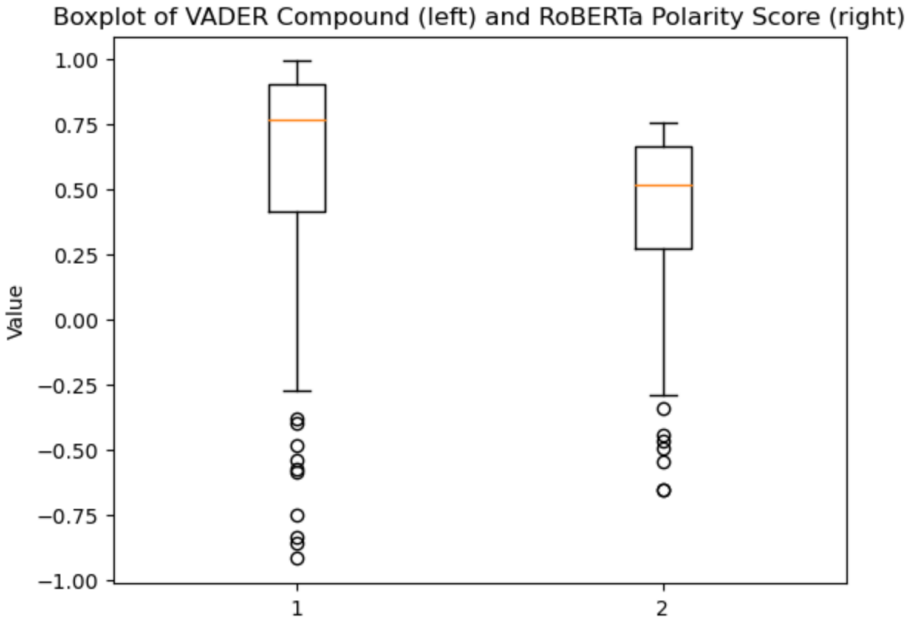

Figure 8: Boxplot of VADER compound and RoBERTa polarity score.

Having computed all compound and RoBERTa polarity scores, we can also plot them directly in relation to each other. A scatter plot of the scores is shown in figure 9. If the overall sentiment results from the two models would be the same or very similar, we would see a linear relationship in this plot. However, we do not, instead the points look almost randomly scattered with an inclination towards higher values. Moreover, an interesting observation is that the points along the upper and right edges show that for a high value of either the compound or the polarity score, the results of the other method range from a little below zero to high. This means that in some instances, one of the methods evaluates the sentiment as strongly positive and the other as neutral or even slightly negative. This is interesting since they are meant to measure the same thing; the overall sentiment of the text.

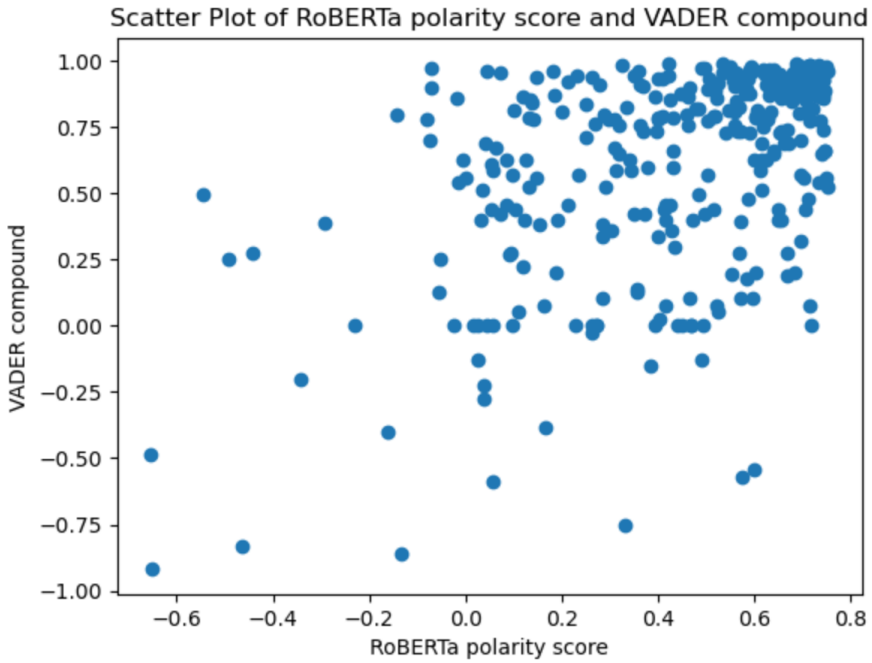

Figure 9: Scatter plot of RoBERTa polarity score versus VADER compound on paragraph level.

Moreover, the two scores can be illustrated in a density plot, which is shown in figure 10. The values on the x-axis exceed the known possible range of −1 to 1 because the kernel density estimate plot provides a smooth visualization of the shapes of the distributions. VADER tends to produce higher numbers, but also lower. Moreover, there is an interesting behavior around 0 for both distributions. It can also be seen that there is a difference in the distributions since they are not completely overlapping. However, is this difference significant?

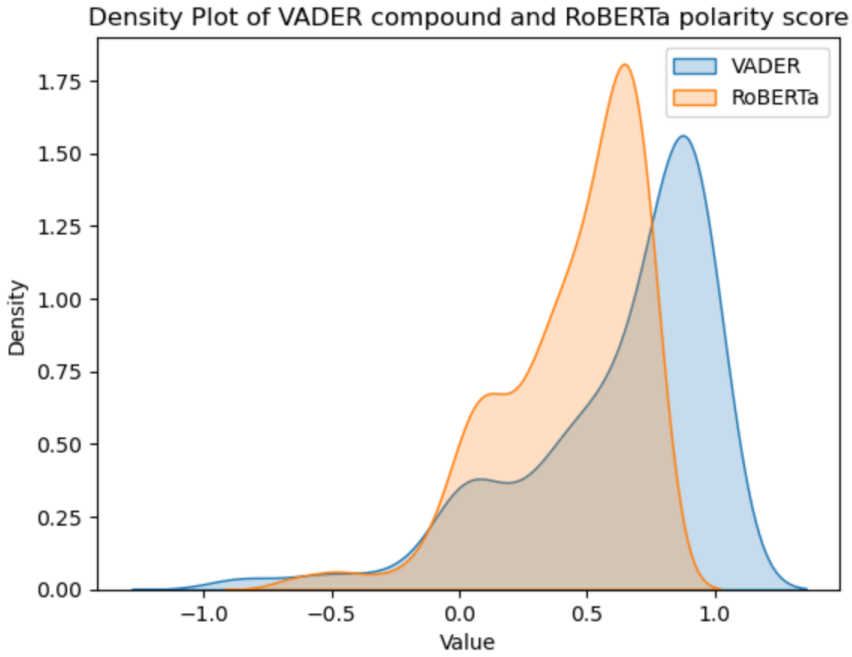

Figure 10: Density plot of VADER compound (blue) and RoBERTa polarity score (orange).

This is investigated using the two-sample Kolmogorov-Smirnov test with the hypotheses:

H_0: *The two values are from the same distribution.*

H_1: *The two values are from different distributions.*

The Kolmogorov-Smirnov test is a non-parametric test and can be used to determine whether two distributions differ (Lopes, 2014). The test relies on the cumulative distribution function (CDF) to make determinations regarding the particular distribution of the data (Massey, 1951). The test is exact for continuous distributions and the samples need to be independent. The first assumption is fulfilled in our case since the scores are continuous between −1 to 1. In order to choose the sample, the companies were randomly drawn with equal probability. Next, all paragraphs of that company's introduction in the sustainability report were selected. One can argue that the paragraphs are independent of each other as the author writes each paragraph about a different key point or subject, with a clean mind, that is not affected by the rest of the text. For example, the author may write the first paragraph with a positive sentiment and then write about something else negatively in the second paragraph. Moreover, in sustainability reports, it is not necessarily the same person who writes all the paragraphs. There may for example be one person

who writes one paragraph, another who writes a second paragraph and a third writing a third paragraph. Therefore, we argue that this assumption is fulfilled. This test examines differences in both shape and location of distributions. Thus, the test provides a general picture of potential differences. The obtained p-value is less than 0.001. Therefore, the null hypothesis is rejected and the conclusion is that the two values were drawn from different distributions.

Since we now know that the distributions are different, we investigate further to see whether their central metrics differ. As mentioned before, the arithmetic means of the two values are 0.61 for VADER and 0.43 for RoBERTa. Furthermore, the medians are 0.77 for VADER and 0.52 for RoBERTa. The two sample t-test is one option. It tests whether the means are different and assumes that the values are normally distributed. As can be seen in figure 10, the distributions for the VADER compound and RoBERTa polarity score do not look like they fulfill this assumption since they are both skewed and heavy-left tailed. However, since the sample size is big, the t-test and z-test are essentially the same as a consequence of the central limit theorem. Nevertheless, an alternative test is the non-parametric Wilcoxon signed-rank test, which tests another central metric, namely the median. Here, an assumption is that the differences between the paired observations have a symmetric distribution (Rey & Neuhäuser, 2014). Figure 11 shows the density plot of the differences between the scores. It can be seen that the distribution is approximately symmetric, with a little larger area to the left. Furthermore, the mean is about 0.18 and the median is 0.22, so they are not completely identical but rather similar.

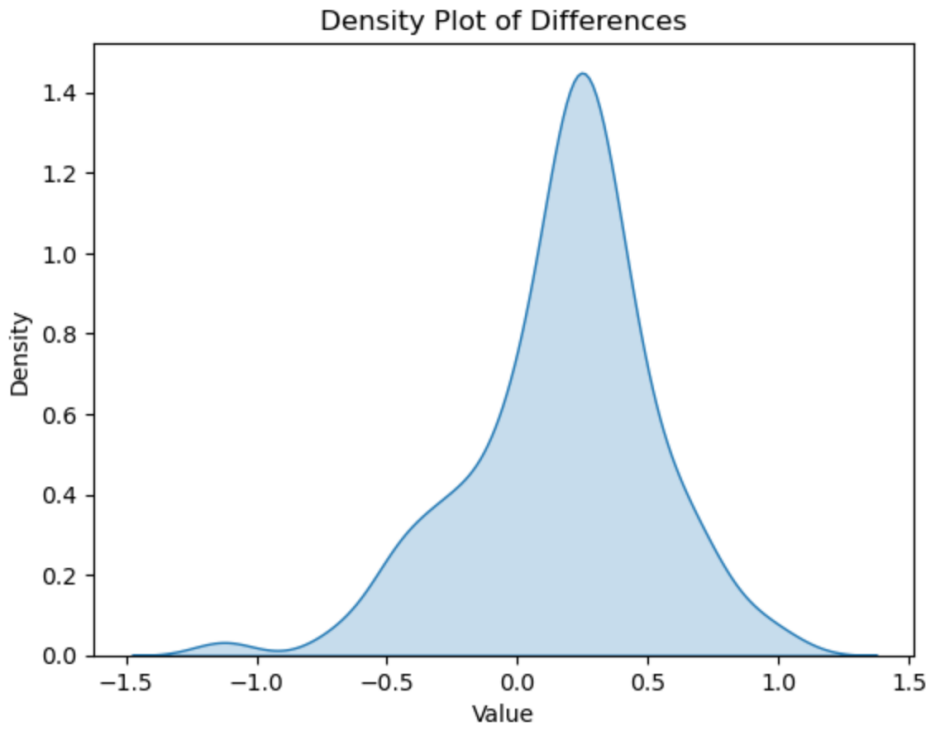

Figure 11: Density plot of the differences between the scores (VADER compound − RoBERTa polarity score).

Other assumptions for the Wilcoxon signed-rank test are that the measurements are continuous and that the differences are independent (Rey & Neuhäuser, 2014). As argued above, these assumptions are fulfilled. In this case, the last mentioned assumption means that there can be no ties in the differences. Also, the differences have to be non-zero (Rey & Neuhäuser, 2014), which follows from our continuous scores. The test uses the hypotheses:

H_0: *The medians are the same.*

H_1: *The medians are different.*

The observed p-value is less than 0.001, which is significant on the three star level. Therefore, the null hypothesis is rejected.

4 Conclusions and Discussion

In this final section of the thesis, the conclusions are first summarized, followed by a discussion of the conclusions, future research and possible limitations of the study.

4.1 Conclusions

The first sub question of this thesis is whether the VADER compound and RoBERTa polarity score follow different distributions. This question covers the general shape and location of the distributions. The compound and the polarity score represent the overall sentiment of a paragraph and the analysis is done based on the paragraph level. The test used for answering this question is a two-sample Kolmogorov-Smirnov test. It is concluded that the distributions of the scores are different.

The second sub question is whether the values are different in terms of the location of their distributions. A Wilcoxon signed-rank test is used to test this question. The conclusion is that the results of the two models differ in terms of their medians.

The main question of this study is how the overall results from the two models differ when applied on the introduction of sustainability reports. This question is answered by the two sub questions and the conclusion is that the distributions are different, both when looking only at location and when looking at location and shape combined.

4.2 Discussion, Future Research and Limitations

The result that the distributions of the VADER compound and RoBERTa polarity score differ in location and shape could be because RoBERTa considers more information since it is better at taking context into account and, as mentioned, has in previous studies achieved higher accuracy than VADER. Therefore, this could be a potential reason for the significantly different results from the models. To know this, future research could study the accuracy of the two models on businesses' sustainability texts. In order to do this, however, one needs to classify the paragraphs manually since there is no evaluation score connected to this type of text. Manually classified

paragraphs could also be used for pre-training a new BERT sentiment model that specializes in sustainability texts. In general, future research could investigate the differences in the results in more detail in order to increase the knowledge about why the methods, in some cases, generate remarkably different results.

A note of caution regarding the results of this study is that there was an overrepresentation of positive paragraphs. However, this could be assumed to be more a rule than an exception when studying sustainability texts published by companies, since they could be inclined to describe their business and their surroundings in a positive way. Therefore, the results are not generalizable to other types of complex texts than companies' sustainability texts.

A limitation of this thesis is that some of the paragraphs may contain too many words for the VADER model to work at its best. An option, instead of working with full paragraphs, would have been to cut the paragraphs into smaller sections and run the models on these sections instead. However, the disadvantage with that approach would be that one would lose some contextual information. Moreover, the assumption of independent samples for conducting the statistical tests would probably not hold. In this thesis, both used models are based or trained on Twitter posts, so they both come from an analysis of rather short paragraphs. Nevertheless, the VADER compound does get closer and closer to the end values, −1 and 1, as the number of words increases and this could influence the results of the longer paragraphs. It was considered that a paragraph of more than 250 words would be too long for VADER to operate well on and therefore an outlier of 299 words was removed. However, it is difficult to assess how many words per paragraph that is too many for VADER since it also depends on how many of these words are included in the VADER dictionary.

References

Akritidis, L., Alamaniotis, M., Fevgas, A. & Bozanis, P. (2020). Confronting Sparseness and High Dimensionality in Short Text Clustering via Feature Vector Projections, 2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI), Tools with Artificial Intelligence (ICTAI), 2020 IEEE 32nd International Conference on, ICTAI, pp. 813–820.

Barbieri, F., Camacho-Collados, J., Neves, L. & Espinosa-Anke, L. (2020). TweetEval: Unified Benchmark and Comparative Evaluation for Tweet Classification, Available online: https://arxiv.org/pdf/2010.12421.pdf [Accessed 18-12-23].

Başarslan, M.S. & Kayaalp, F. (2021). Sentiment Analysis on Social Media Reviews Datasets with Deep Learning Approach, *Sakarya University Journal of Computer and Information Sciences*, Vol. 4, No. 1, pp. 35–49.

Bhonde, R., Bhagwat, B., Ingulkar, S. & Pande, A. (2015). Sentiment Analysis Based on Dictionary Approach, *International Journal of Emerging Engineering Research and Technology*, Vol. 3, No. 1, pp. 51-55.

Bird, S., Klein, E., & Loper, E. (2009). Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit. O'Reilly Media, Inc., Available online: https://www.nltk.org/book/ [Accessed 27-12-23].

Bonta, V., Kumaresh, N. & Janardhan, N. (2019). A Comprehensive Study on Lexicon Based Approaches for Sentiment Analysis, *Asian Journal of Computer Science and Technology*, Vol. 8, No. S2, pp. 1–6.

Camacho-Collados, J., Rezaee, K., Riahi, T., Ushio, A., Loureiro, D., Antypas, D., Boisson, J., Espinosa-Anke, L.,, Liu, F., Martínez-Cámara, E., Medina, G., Buhrmann, T., Neves, L. & Barbieri, F. (2022). TweetNLP: Cutting-Edge Natural Language Processing for Social Media, Available online: https://arxiv.org/pdf/2206.14774.pdf [Accessed 18-12-2023].

Catelli, R., Pelosi, S. & Esposito, M. (2022). Lexicon-Based vs. Bert-Based Sentiment Analysis: A Comparative Study in Italian, *Electronics*, Vol. 11, No. 3.

Chakriswaran, P., Vincent, D.R., Srinivasan, K., Sharma, V., Chang, C.-Y. & Reina, D.G. (2019). Emotion AI-Driven Sentiment Analysis: A Survey, Future Research Directions, and Open Issues, *Applied Sciences*, Vol. 9, No. 24.

Chernyavskiy, A., Ilvovsky, D. & Nakov, P. (2021). Transformers: "The End of History" for Natural Language Processing?, Available online: https://arxiv.org/pdf/2105.00813.pdf [Accessed 18-12-2023].

Devlin, J., Chang, M-W., Lee, K. & Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, Google AI Language, Available online: https://arxiv.org/pdf/1810.04805.pdf [Accessed 28-12-23].

Dhanalakshmi, P., Kumar, G.A., Satwik, B.S., Sreeranga, K., Sai, A.T. & Jashwanth, G. (2023). Sentiment Analysis Using VADER and Logistic Regression Techniques, 2023 International Conference on Intelligent Systems for Communication, IoT and Security (ICISCoIS), Coimbatore, India, 2023, pp. 139–144.

Elbagir, S. & Yang, J. (2019). Twitter Sentiment Analysis Using Natural Language Toolkit and VADER Sentiment, Proceedings of the International MultiConference of Engineers and Computer Scientists 2019 IMECS 2019, March 13–15, 2019, Hong Kong.

Garrido-Merchán, E.C., Gozalo-Brizuela, R. & González-Carvajal, S. (2023). Comparing BERT Against TraditionalMachine Learning Models in TextClassification, *Journal of Computational and Cognitive Engineering*, Vol. 2, No. 4, pp. 352–356.

Hardeniya, T. & Borikar, D.A. (2016). Dictionary Based Approach to Sentiment Analysis - A Review, *International Journal of Advanced Engineering, Management and Science (IJAEMS)*, Vol. 2, No. 5, pp. 317–322.

Hutto, C.J. & Gilbert, E.E. (2014). VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text. Eighth International Conference on Weblogs and Social Media (ICWSM-14). Ann Arbor, MI, June 2014.

Iqbal, M., Karim, A. & Kamiran, F. (2015). Bias-Aware Lexicon-Based Sentiment Analysis, SAC '15: Proceedings of the 30th Annual ACM Symposium on Applied Computing, April 2015, pp. 845–850.

Johnstone, I.M. & Titterington, D.M. (2009). Introduction: Statistical Challenges of High-Dimensional Data, *Philosophical Transactions: Mathematical, Physical and Engineering Sciences*, Vol. 367, No. 1906, pp. 4237–4253.

Kumar Jain, P., Pamula, R. & Srivastava, G. (2021). A systematic literature review on machine learning applications for consumer sentiment analysis using online reviews, *Computer Science Review*, Vol. 41, No. 100413.

Ligthart, A., Catal, C. & Tekinerdogan, B. (2021). Systematic reviews in sentiment analysis: a tertiary study, *Artif Intell*, Vol. 54, pp. 4997–5053.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L. & Stoyanov, V. (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach, Available online: https://arxiv.org/pdf/1907.11692.pdf [Accessed 18-12-2023].

Marr, B. (2023). A Short History of ChatGPT: How We Got To Where We Are Today, Available online: https://www.forbes.com/sites/bernardmarr/2023/05/19/a-short-history-of-chatgpt-how-we-got-to-where-we-are-today/?sh=f42b7c674f14 [Accessed 08-10-23].

Massey, F. J. (1951). The Kolmogorov-Smirnov Test for Goodness of Fit, *Journal of the American Statistical Association*, Vol. 46, No. 253, pp. 68–78.

Mishev, K., Gjorgjevikj, A., Vodenska, I., Chitkushev, L.T. & Trajanov, D. (2020). Evaluation of Sentiment Analysis in Finance: From Lexicons to Transformers, *IEEE Access*, Vol. 8, pp. 131662–131682.

NLTK Project. (2023a). Documentation: Natural Language Toolkit, Available online: https://www.nltk.org/ [Accessed 03-12-23].

NLTK Project. (2023b). Documentation: Sample Usage for Sentiment, Available online: https://www.nltk.org/howto/sentiment.html [Accessed 03-12-23].

Onan, A. (2020). Sentiment analysis on product reviews based on weighted word embeddings and deep neural networks, *Concurrency and Computation Practice and Experience*, Vol. 33, No. 5.

Queipo, S., Garcia-Cabot, A., Garcia-Lopez, E. & de-Fitero Domínguez, D. (2022). Introduction to Transformers: A Breakthrough in NLP, ATICA2022 - Aplicación de Tecnologías de la Información y Comunicaciones Avanzadas y Accesibilidad, pp. 142–186.

Rahnenführer, J., De Bin, R., Benner, A., Ambrogi, F., Lusa, L., Boulesteix, A-L., Migliavacca, E., Binder, H., Michiels, S., Sauerbrei, W. & McShane, L. (2023). Statistical analysis of high-dimensional biomedical data: a gentle introduction to analytical goals, common approaches and challenges, *BMC Medicine*, Vol. 21, No. 1, pp. 1–54.

Raul H. C. Lopes, R.H.C. (2014). Kolmogorov-Smirnov Test, in Lovric, M. (ed.), International Encyclopedia of Statistical Science. Berlin, Heidelberg: Springer. pp. 718–720.

Rey, D. & Neuhäuser, M. (2014). Wilcoxon-Signed-Rank Test, in Lovric, M. (ed.), International Encyclopedia of Statistical Science. Berlin, Heidelberg: Springer. pp. 1658–1659.

Rice, D.R & Zorn, C. (2021). Corpus-based dictionaries for sentiment analysis of specialized vocabularies, *Political Science Research and Methods*, Vol. 9, No. 1, pp. 20–35.

Sayeed, M.S., Mohan, V. & Muthu, K.S. (2023). BERT: A Review of Applications in Sentiment Analysis, *HighTech and Innovation Journal*, Vol. 4, No. 2, pp. 453–462.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł. & Polosukhin, I. (2017). Attention Is All You Need, 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

Wu, Y., Schuster, M., Chen, Z., Le, Q.V. & Norouzi, M. (2016). Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation, Available online: https://arxiv.org/pdf/1609.08144.pdf [Accessed 28-12-23].

Xiao, T. & Zhu, J. (2023). Introduction to Transformers: an NLP Perspective, Available online: https://arxiv.org/pdf/2311.17633.pdf [Accessed 01-01-24].

Yang, L., Li, Y., Wang, J. & Sherratt, R.S. (2020). Sentiment Analysis for E-Commerce Product Reviews in Chinese Based on Sentiment Lexicon and Deep Learning, *IEEE Access*, Vol. 8, pp. 23522–23530.

Appendix A - Included companies in the study and non-responses

Table A1: Table of the companies whose sustainability report was included in the study.

Studied Reports	
AAK	Intrum
Alfa Laval	INVISIO
Astra Zeneca	ITAB Shop Concept
Axfood	Lagercrantz Group
Bergs Timber	Lundbergföretagen
Byggfakta Group	Midsona
Coor Service Management (1 outlier paragraph removed)	Momentum Group
CTEK	Nanologica
Duroc	Net Insight
Electrolux	Norva24 Group
Ependion	NP3 Fastigheter
Epiroc	Peab
EQT	Ratos
Evolution	Sandvik
Fabege	SkiStar
Fagerhult Group	SSAB
Ferronordic	Swedbank
Fortnox	Tietoevry
G5 Entertainment	Trianon
Gaming Innovation Group	VEF
HANZA	Volvo Cars
HEXPOL	XSpray Pharma

Table A2: Non-responses.

Non-responses	
Company	**Reason**
Abliva	Does not publish a sustainability report
Active Biotech	Does not publish a sustainability report
Eniro	Only in Swedish
Rizzo Group	Does not publish a sustainability report
Starbreeze	Did not publish their report for 2022 yet
Wise Group	Only in Swedish

Appendix B - Python code

```
# Import libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from nltk.tokenize import word_tokenize
from string import punctuation
from wordcloud import WordCloud
import re
import nltk
nltk.download('vader_lexicon')
from nltk.sentiment import SentimentIntensityAnalyzer
from tqdm.notebook import tqdm
import seaborn as sns
import transformers
transformers.__version__
from transformers import AutoTokenizer, AutoModelForSequenceClassification,
pipeline
import torch
import requests
from bs4 import BeautifulSoup
from scipy.special import softmax
import torch.nn as nn

df = pd.read_excel('ParagraphData.xlsx')

#Histogram number of paragraphs per report
plt.hist(df['Number of paragraphs'], bins=range(int(min(df['Number of
paragraphs'])), int(max(df['Number of paragraphs'])) + 2), edgecolor='black')
plt.title('Histogram of Number of Paragraphs per Sustainability Report')
plt.xlabel('Number of Paragraphs')
plt.ylabel('Count')
plt.xticks(range(int(min(df['Number of paragraphs'])), int(max(df['Number of
paragraphs'])) + 1))
plt.show()

#make data frame with only paragraph data
df_only_paragraph = df.drop(columns=['Company', 'Sector', 'Number of
paragraphs'])

#make one data frame for each row (company)
dataframes_list = [pd.DataFrame(df_only_paragraph.iloc[i]).transpose() for i in
range(len(df_only_paragraph))]

## Delete outlier list index 23, Paragraph_2
new_dataframes_list = []
for i, df in enumerate(dataframes_list):
    # Check index is 23 and "Paragraph_2" in the df
```

```
    if i == 23 and "Paragraph_2" in df.columns:
        # Drop "Paragraph_2"
        new_dataframes_list.append(df.drop(columns=["Paragraph_2"]))
    else:
        # Append unmodified df to new list
        new_dataframes_list.append(df)

df2 = df_only_paragraph
df2.replace('Coor’s framework for sustainability work consists of the Group’s
sustainability policy, Code of Conduct and values/guiding stars. The Board of
Directors has ultimate responsibility for Coor’s organisation and operations,
and continuously assesses the company’s performance from a triple bottom line
perspective. The Board addresses strategic matters, financial performance and
matters relating to customers, employees, sustainability and risk management.
They also monitor progress towards the company’s sustainability goals and Coor
2025 – the company’s ambition to become a truly sustainable company. The Board
of Directors has delegated operational responsibility for the company and its
management to the company’s President and CEO, AnnaCarin Grandin, who leads the
activities within the limits and guidelines established by the Board. This
responsibility includes setting goals for the company’s operational activities,
allocating resources and monitoring sustainability issues. Members of the
executive management team have been assigned responsibility for strategic
development of the various sustainability dimensions. The business dimension is
led by the CFO, the social dimension, which includes diversity and inclusion,
by the HR Director, and the environmental dimension by the Head of
Sustainability, who is invited to attend meetings of the executive management
team when necessary. The CFO monitors the issues addressed at all meetings. A
separate management team, the Sustainability Management Team (SuMT), which
reports directly to the EMT, is responsible for governance, decisions on focus
areas for Coor 2025 – a truly sustainable company, and prioritisation of major
strategic sustainability initiatives as well as monitoring. The SuMT acts as a
sponsor for the strategy and has a mandate to make decisions at the executive
level and ensure compliance with policy decisions related to sustainability
management. As part of its remit, the SuMT also prepares draft triple bottom
line decisions for the consideration of the executive management team. ',
pd.NA, inplace=True)

#Number of words per paragraph
#Create words count function
def count_words(text):
    words = word_tokenize(text)
    words = [word for word in words if word.isalnum()] #excludes punctuation as
a word
    return len(words)

#Original dataframes_list
def process_dataframe(df):
    new_column_data = []
    df.fillna(0, inplace=True)
```

```
    for index, row in df.iterrows():
        for column in df.columns:
            if row[column] == 0:
                break
            words = count_words(str(row[column]))
            new_column_data.append({
                'Words': words
            })
return pd.DataFrame(new_column_data)
# Process all dataframes in dataframes_list
words_dfs = [process_dataframe(df) for df in dataframes_list]
#All dataframes into one
ALL_word_df = pd.concat(words_dfs, ignore_index=True)

#Mean, max, min
ALL_word_df.mean()
ALL_word_df.max()
ALL_word_df.min()
ALL_word_df.idxmax()

#Histogram number of words per paragraph
plt.hist(ALL_word_df['Words'], bins=15, edgecolor='black')
plt.title('Histogram of Number of Words per Paragraph')
plt.xlabel('Number of Words')
plt.ylabel('Count')
plt.show()

#Removed outlier
def process_dataframe(df):
    new_column_data = []
    df.fillna(0, inplace=True)
        for index, row in df.iterrows():
        for column in df.columns:
            if row[column] == 0:
                break
            words = count_words(str(row[column]))
            new_column_data.append({
                'Words': words
            })
        return pd.DataFrame(new_column_data)
# Process all dataframes in dataframes_list
words_dfs = [process_dataframe(df) for df in new_dataframes_list]
#All dataframes into one
ALL_word_df = pd.concat(words_dfs, ignore_index=True)

#New mean and max
ALL_word_df.mean()
ALL_word_df.max()
```

```
## Word cloud
# Create a new df with a single column with non-NaN values
non_nan_values = pd.DataFrame(df2.stack().dropna().values,
columns=['Non_NaN_Values'])

#Word list
words_list = [word_tokenize(word) for word in non_nan_values['Non_NaN_Values']]
words = ' '.join(' '.join(sublist) for sublist in words_list)
words = ' '.join(word for word in words.split() if word.lower() != 's')
wordcloud=WordCloud(width=1000, height=600, random_state=10,
max_font_size=110).generate(words)
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.show()

## VADER
sia = SentimentIntensityAnalyzer()

#VADER first company AAK
new_column_data=[]
AAK=dataframes_list[0]
AAK.fillna(0, inplace=True)
for column in AAK.columns:
    if AAK.at[0, column] == 0:
        break
    res=sia.polarity_scores(str(AAK.at[0, column]))
    print(res)
    new_column_data.append({
        'Neg': res['neg'],
        'Neu': res['neu'],
        'Pos': res['pos'],
        'Compound': res['compound']
    })
AAK_VADER_result_df = pd.DataFrame(new_column_data)
.
.
.
#VADER last company XsprayPharma
new_column_data=[]
XsprayPharma=dataframes_list[43]
XsprayPharma.fillna(0, inplace=True)
for column in XsprayPharma.columns:
    if XsprayPharma.at[43, column] == 0:
        break
    res=sia.polarity_scores(str(XsprayPharma.at[43, column]))
    print(res)
    new_column_data.append({
        'Neg': res['neg'],
        'Neu': res['neu'],
```

```
        'Pos': res['pos'],
        'Compound': res['compound']
    })
XsprayPharma_VADER_result_df = pd.DataFrame(new_column_data)

# List of ALL VADER DataFrames
VADER_dfs = [AAK_VADER_result_df, AlfaLaval_VADER_result_df, …,
XsprayPharma_VADER_result_df]

#Data frame of ALL VADER values
ALL_VADER_df = pd.concat(VADER_dfs, ignore_index=True)

#VADER Average Compound (average per paragraph)and variance
print(ALL_VADER_df['Compound'].mean())
print(ALL_VADER_df['Compound'].var())

# Find maximum
max_value = ALL_VADER_df['Compound'].max()
# Find minimum
min_value = ALL_VADER_df['Compound'].min()
#Print
print(f"Maximum value: {max_value}")
print(f"Minimum value: {min_value}")
# Find the index of minimum
min_index = ALL_VADER_df['Compound'].idxmin()
# Find the index of maximum
max_index = ALL_VADER_df['Compound'].idxmax()
#Print
print(f"Index of minimum value: {min_index}")
print(f"Index of maximum value: {max_index}")

## RoBERTa
MODEL = f"cardiffnlp/twitter-roberta-base-sentiment-latest"
tokenizer = AutoTokenizer.from_pretrained(MODEL)
model = AutoModelForSequenceClassification.from_pretrained(MODEL)

#RoBERTa first company - AAK
new_column_data=[]
AAK=dataframes_list[0]
AAK.fillna(0, inplace=True)
for column in AAK.columns:
    if AAK.at[0, column] == 0:
        break
    encoded_text = tokenizer(str(AAK.at[0, column]), return_tensors='pt')
    output = model(**encoded_text)
    scores = output[0][0].detach().numpy()
    scores = softmax(scores)
    scores_dict = {
    'roberta_neg' : scores[0],
```

```
    'roberta_neu' : scores[1],
    'roberta_pos' : scores[2]
}
    new_column_data.append({
        'Neg': scores_dict['roberta_neg'],
        'Neu': scores_dict['roberta_neu'],
        'Pos': scores_dict['roberta_pos']
    })

AAK_ROBERTA_result_df = pd.DataFrame(new_column_data)
#make polarity score comparable to compound score for VADER
polarity_weights = torch.tensor([-1, 0, 1])
probs = torch.tensor(AAK_ROBERTA_result_df[["Neg", "Neu", "Pos"]].values)
polarity = polarity_weights * probs
polarity = polarity.sum(dim=-1)
polarity_scaled = nn.Tanh()(polarity)
AAK_ROBERTA_result_df["RoBERTa Polarity Score"] = polarity_scaled.numpy()
.
.
.
#RoBERTa last company - XsprayPharma
new_column_data=[]
XsprayPharma=dataframes_list[43]
XsprayPharma.fillna(0, inplace=True)
for column in XsprayPharma.columns:
    if XsprayPharma.at[43, column] == 0:
        break
    encoded_text = tokenizer(str(XsprayPharma.at[43, column]),
return_tensors='pt')
    output = model(**encoded_text)
    scores = output[0][0].detach().numpy()
    scores = softmax(scores)
    scores_dict = {
    'roberta_neg' : scores[0],
    'roberta_neu' : scores[1],
    'roberta_pos' : scores[2]
}
    new_column_data.append({
        'Neg': scores_dict['roberta_neg'],
        'Neu': scores_dict['roberta_neu'],
        'Pos': scores_dict['roberta_pos']
    })

XsprayPharma_ROBERTA_result_df = pd.DataFrame(new_column_data)
#make polarity score comparable to compound score for VADER
polarity_weights = torch.tensor([-1, 0, 1])
probs = torch.tensor(XsprayPharma_ROBERTA_result_df[["Neg", "Neu",
"Pos"]].values)
polarity = polarity_weights * probs
```

```
polarity = polarity.sum(dim=-1)
polarity_scaled = nn.Tanh()(polarity)
XsprayPharma_ROBERTA_result_df["RoBERTa Polarity Score"] =
polarity_scaled.numpy()

# List of ALL ROBERTA DataFrames
ROBERTA_dfs = [AAK_ROBERTA_result_df, AlfaLaval_ROBERTA_result_df, …,
XsprayPharma_ROBERTA_result_df]
#Data frame of ALL RoBERTa values
ALL_ROBERTA_df = pd.concat(ROBERTA_dfs, ignore_index=True)

#RoBERTa Average Compound (average per paragraph) and variance
print(ALL_ROBERTA_df['RoBERTa Polarity Score'].mean())
print(ALL_ROBERTA_df['RoBERTa Polarity Score'].var())

# Find maximum
max_value = ALL_ROBERTA_df['RoBERTa Polarity Score'].max()
# Find minimum
min_value = ALL_ROBERTA_df['RoBERTa Polarity Score'].min()
#Print
print(f"Maximum value: {max_value}")
print(f"Minimum value: {min_value}")
# Find the index of minimum
min_index = ALL_ROBERTA_df['RoBERTa Polarity Score'].idxmin()
# Find the index of maximum
max_index = ALL_ROBERTA_df['RoBERTa Polarity Score'].idxmax()
#Print
print(f"Index of minimum value: {min_index}")
print(f"Index of maximum value: {max_index}")

#Density plot of compound/polarity score
# Create a kernel density estimate
sns.kdeplot(data=ALL_VADER_df['Compound'], fill=True, label='VADER',
cmap="Blues", thresh=0, levels=30)
sns.kdeplot(data=ALL_ROBERTA_df['RoBERTa Polarity Score'], fill=True,
label='RoBERTa', cmap="Reds", thresh=0, levels=30)
# Set plot labels, title, legend
plt.xlabel('Value')
plt.ylabel('Density')
plt.title('Density Plot of VADER compound and RoBERTa polarity score')
plt.legend()
# Show plot
plt.show()

#Scatter plot
plt.scatter(ALL_ROBERTA_df['RoBERTa Polarity Score'], ALL_VADER_df['Compound'])
plt.title('Scatter Plot of RoBERTa polarity score and VADER compound')
plt.xlabel('RoBERTa polarity score')
plt.ylabel('VADER compound')
```

```
plt.show()

#Boxplot
#Box plots of compound and polarity score
new_df = pd.concat([ALL_VADER_df['Compound'], ALL_ROBERTA_df['RoBERTa Polarity
Score']], axis=1)
plt.boxplot(new_df, notch=None, sym=None, vert=None, whis=None, positions=None,
widths=None, patch_artist=None, bootstrap=None, usermedians=None,
conf_intervals=None, meanline=None, showmeans=None, showcaps=None,
showbox=None, showfliers=None, boxprops=None, labels=None, flierprops=None,
medianprops=None, meanprops=None, capprops=None, whiskerprops=None,
manage_ticks=True, autorange=False, zorder=None, capwidths=None)
plt.title('Boxplot of VADER Compound (left) and RoBERTa Polarity Score
(right)')
plt.ylabel('Value')

new_df.median()
new_df.min()
new_df.max()
new_df.quantile(q=[0.25, 0.75], interpolation='midpoint')

##Tests
#Wilcoxon
#Difference
diff=pd.DataFrame(ALL_VADER_df['Compound']-ALL_ROBERTA_df['RoBERTa Polarity
Score'])
#Density plot of diff
sns.kdeplot(data=diff[0], fill=True, label='Diff', cmap="Blues", thresh=0,
levels=30)
# Set plot labels and title
plt.xlabel('Value')
plt.ylabel('Density')
plt.title('Density Plot of Differences')
plt.show()

mean = np.mean(diff)
median = np.median(diff)
print("Mean:", mean)
print("Median:", median)


import scipy.stats as stats
from scipy.stats import wilcoxon
#the test
wilcoxon(ALL_VADER_df['Compound'], ALL_ROBERTA_df['RoBERTa Polarity Score'],
alternative='two-sided')

#K-S
```

```
stats.ks_2samp(ALL_VADER_df['Compound'], ALL_ROBERTA_df['RoBERTa Polarity
Score'], alternative='two-sided', method='auto')
```