Final Year Project

A test of sentiment analysis

Zhihang Zhang

Student ID: 20748939

A thesis submitted in part fulfilment of the degree of

BSc. (Hons.) in Computer Science

Supervisor: Professor Fred Cummins

UCD School of Computer Science
University College Dublin
April 9, 2024

Table of Contents

Abstract

Natural language is complex as it can be ambiguous, context-dependent, and variable depending on the communities, regions, or cultures. Sentiment analysis is one of the critical natural language processing techniques used to categorize the human emotion behind natural language. This paper seeks to provide a comprehensive analysis of current sentiment analysis field, which includes finding state-of-the-art approaches, outlining data considerations, evaluating their performance compared to the ground truth, and discussing their capabilities and limitations.

In this paper, multiple tools have been tested for sentiment analysis on a special dataset that consists of a corpus of news headlines containing the word "ritual". The uniqueness of this dataset is supportive of gaining more insightful results from analysis since "ritual" can be rooted in different cultures, religions, and senses of community belonging. The result from comparative analysis across four types of tools namely: Vader (lexicon-based), DistillBert(Transformer based fine-tuned by movie dataset), Siebert(Transformer based fine-tuned by diverse text sources), and Chatgpt, shows all of the tested tools reach a fairly good accuracy of over 75%, with Chatgpt playing the most well,followed by Siebert, DistillBert and Vader.

This paper offers valuable insights into the current state of sentiment analysis using a more depth dataset in terms of the complexity of human language and inspires the researcher or developer about future directions based on tool's limitations.

Chapter 1: **Project Specification**

Core Goals:

1. Applying sentiment analysis tools to a given data set: The primary objective is to use sentiment analysis tools on a given data-set that consists of a corpus of news headlines containing the word "ritual." This database is unique as it provides a strong polarity that is clear to a human on cursory inspection.
2. Evaluating the performance of sentiment analysis tools : Evaluate the performance of selected sentiment analysis tools to see if they can correctly identify the sentiment behind those unique news headlines. The evaluation result will show how well these tools could distinguish sentiment in different contexts compared to human beings.

Advanced Goals:

1. Comparing across different sentiment tools: Apart from evaluating each sentiment tool, compare different sentiment tools to find out which sentiment analysis tool plays most and least well on this data-set.
2. Understanding the tool mechanics: Understand how these tools work for sentiment analysis like the algorithms or logic underlying the sentiment analysis tools and gain insight from it.
3. Conclusion to the sentiment analysis field: Discuss the capabilities and limitations of current sentiment analysis tools on data-sets with strong emotional polarity based on the test results and knowledge of tool mechanics.

Chapter 2: **Introduction**

Sentiment analysis is a natural language processing technique used to find out whether the human emotional attitude conveyed by the data is positive, negative, or neutral. With the quick exploration of computer science, this technique now involves different fields like deep learning, data mining, artificial intelligence, and computational linguistics. [1]

The earliest motivations behind sentiment analysis can be traced back to the period during and after WWII because of the growing need to measure public opinions. [2] The method used in early sentiment analysis before a large amount of text or opinions were available online was the survey-based method, followed by mid-90s computer-based systems that appeared to be used mainly on expert opinion analysis for industrial safety. Modern sentiment analysis started around the mid-2000's focusing on the reviews available on the web. Since then, the use case of sentiment analysis has gradually expanded to many areas[2] due to various factors, especially the rise of the internet and digital communication that drove the development of sentiment analysis. The internet generates a tremendous amount of text data every day, including social media comments, customer reviews, feedback surveys, and customer service chat transcripts, from which sentiment analysis tools are used to derive many applications. [3]

What are examples of the application of sentiment analysis? Recommender System Through Sentiment Analysis: Reading positive and negative reviews can have different impacts on people's choices. Building a multilingual recommender system combined with sentiment analysis can help users make decisions on products using online reviews. [4] Brand Reputation: By constantly monitoring posts about the brand on social media, the public relations team can be aware of any negative trend through sentiment analysis and evaluate the underlying emotions to deal with complaints properly.[3] Customer Service and Product Improvement: Sentiment analysis provided a way to analyze customer feedback and reviews, which helped to improve their offerings and customer service. [5] Political Analysis and Market Analysis: Apply social media sentiment analysis for political analysis to predict the potential top candidates in an election, which helps marketers to identify trends and make desired decisions based on public opinion. [6]

Sentiment analysis can be so useful in various areas because of its advantage of providing unique and valuable insight. Its capability and accuracy in correctly identifying sentiment become crucial. A question arises: can sentiment analysis tools correctly find out the human emotional attitude conveyed by the text data? Due to the complexity of sentimental expression by the human being, there are possible challenges and limitations that sentiment analysis can face. Sarcasm and irony: the opposite of true opinions, which can lead sentiment analysis tools to produce incorrect results. Ambiguity: The words and phrases can be ambiguous, which can result in misreading by sentiment analysis tools. Understanding Context: The accuracy of the result of sentiment analysis can be affected without consideration of the context of the sentence. Variations Across Languages and Slang: The sentiment analysis tools can struggle with different languages and their slang, as they have distinct cultural contexts that can alter the meaning of words.[7] In order to have a full understanding of how well sentiment tools can distinguish sentiment in different contexts compared to human beings, I will be testing various sentiment analysis tools on a unique data-set that consists of a corpus of news headlines containing the word "ritual." This data-set clearly illustrates two extreme senses of either being very positive or negative, which provides strong polarity in sentiments that are evident to humans at rough inspection.

Evaluating the capabilities of the current sentiment analysis tool, such as whether the tool can go beyond single word recognition and truly understand the context of a sentence or even a paragraph. The uniqueness of this data set might lead to more insightful results for sentiment tool tests. The word "ritual" can be associated with both positive and negative meanings depending on the context, such as "moon ritual" that normally mean an positive ancient practice helping you to focus inward, but "ritual killing" can mean an negative event involving human or animal killing for unusual purpose. Human activities like ritual are complicated, even as human beings can be confused with defining 'good ritual' and 'bad ritual' because this word can be rooted in different cultures, religions, and senses of community belonging, so how people who have different cultures, societies, or life experiences perceive and describe rituals can be completely different, such as 'ritual slaughter' that normally is sensed as negative for most ordinary people who do not engage in it, while it can mean a positive ritual for food production by some muslim and jewish communities, and 'moon ritual' can be involved in the culture that worships the moon; this is normally recognised as positive to most people by its positive aspect, while it can seem weird for people who grew up without such experience. The current sentiment analysis often conduct on reviews like restaurant reviews and movie for commercial purpose, those reviews can be problematic sample sets as they do not reflect the complexity of human thought. At the end of this research, whether the performance of current sentiment analysis tools reaches human judgemental levels or not, especially in cases where a single word can strongly change the sentiment depending on its context, we can gain valuable insight about the limitations and advantages of current sentiment analysis tools.

Chapter 3: **Related Work and Ideas**

The core of this research is to test sentiment analysis tools and see how they fare on the given data-set, which raises the important question: what sentiment analysis tools should be used in this research? Sentiment analysis has been a constantly developing domain within the natural language process since it first emerged. The article [1] mentioned that this technique now involves different fields like deep learning, data mining, artificial intelligence, and computational linguistics. In order to have a comprehensive understanding of the current sentiment analysis tools, knowing current state of art is crucial for the research, which make sure that the project is based on the most recent and effective methodologies. This section will dig in previous research and various methodological approaches that have been used in sentiment analysis.

Lexicon-Based approach:

In the research done by Soonh Tajet et al, [8] sentiment analysis can be conducted using a lexicon-based approach. This approach uses a predefined dictionary that has each word in its corresponding polarity to assign sentiment scores to words and calculate the overall sentiment score of each article based on those word scores. In simple words, determining the sentiment would be based on the polarity of words within an article without training or using machine learning models.[8] By investigating the mechanics of lexicon-based approaches, this approach could fall short of expectations because of limitations like defining words with an incorrect sentiment score in the dictionary, it is difficult to identify new entities (that are not recorded) and the fixed sentiment score for words can lead to incorrect classification as the sentiment of words can be different depending on context. In the case of this project, a combined word group like "moon ritual" could be incorrectly classified because both words "moon" and "ritual" respectively were assigned by the fixed or incorrect sentiment score without consideration of the context.

Machine Learning-Based Approaches:

A survey research on sentiment analysis done by Wankhade et al,[9] They summarised the idea that machine learning-based approaches can be used in sentiment analysis. This approach primarily consists of supervised and unsupervised learning. The way that unsupervised approaches work for sentiment analysis, such as clustering relies on various resources like knowledge bases, ontologies, databases, and lexicons that contain predetermined information or rules relevant to sentiment analysis. Supervised approaches requires a labelled training data-set to train a machine learning model, including Naive Bayes, Support Vector Machine, Logistic Regression, Decision Tree, and KNN. The trained model can predict or classify the target variable (in this case, the sentiment class label) for new data.[9] As sentiment analysis is dealing with natural human language, this approach requires feature extraction from text data, which involves transforming the text into numerical features that can be understood by the machine so the model can learn from the training data-set.[10]

In the research,[11] the traditional supervised machine learning algorithms, including Support Vector Machine, K-Nearest Neighbour, Decision Trees, and Naive Bayes, are used to conduct sentiment analysis on almost 3700 tweets with the country and the date of tweets about violence against women in Arab countries. Each tweet is manually labelled with either disagree(0) or agree(1) by human beings, in which 0 represents a tweet against women's rights and 1 represents a tweet with women's rights. The data preprocessing techniques were used on the raw data for the training step, and as an experimental result, Knn got **75.86%** accuracy, Naive Bayes got **71.07%** accuracy, the decision tree got **75.25%** accuracy, and SVM got the best accuracy with **78.25%** compared to the others. From the results of sentiment analysis for those tweets, they found that awareness about women's rights has been constantly increasing from 2007 to 2019.

Machine learning-based approaches can be more accurate than lexicon-based approaches because the algorithms learn from a training data-set, which identifies and extracts relevant features from text data and maps those features to sentiment class labels (learn from the variation across features of text). With the proper algorithm used and a large amount of data fed in, this approach can deal better with contextual understanding than a lexicon-based approach.[9] In the case of this project, feeding a sufficient amount of good-quality training data regarding the word "ritual" can possibly make the model more reliable for classifying the sentiment of the news headline about "ritual." However, the pitfall of a machine learning-based approach is quite straightforward: in order to build a robust machine learning model, it requires a sufficient amount of data of good quality; the biased or inaccurate data can lead to a model with terrible performance.

Deep Learning Approaches:

Subsection 1.4 in chapter 1 of the book written by Neha Gupta and Rashmi Agrawal,[12] It is mentioned that a deep learning approach can be used in sentiment analysis. This approach offers the ability to learn features from data using deep learning models, which extract and transform features to discover the representations needed for classification from the original data. In survey research done by Kian et al,[13] outlined deep learning approach in sentiment that in order to learn complex representations of textual data, common practice involves preprocessing textual data and then vectorization by pre-trained word embeddings like word2vec, followed by training deep learning model such as CNNs, RNNs, LSTM, and GRU with those vectors for representation learning.[13] The role of word embeddings in this approach is converting textual data into vectors. "Each text is transformed into a sequence of numbers, where each number maps to a word in the vocabulary, and semantic mapping is used so that words with similar meanings are mapped closely to positioned vectors[14]". Just like supervised machine learning approaches it is obvious that the traditional deep learning model also requires a larger amount of labeled data to train. In the survey research,[9] transfer learning is another advanced method used in sentiment analysis and other natural language processing jobs. This technique is part of deep learning where a model often based on transformer architecture, which reuses learned knowledge from a pre-trained model that was built for a specific task on a new related job like sentiment analysis. In the famous paper "Attention Is All You Need",[15] the authors introduced a new network architecture called Transformer, which is popular in the field of natural language processing to use as model architecture because of its self-attention mechanism. This architecture follow the encoder-decoder structure that common neural model use but with mechanism (so-called scaled dot-product attention) that maps input-transformed vectors, a query, and a set of key-value pairs to a weighted sum of the values where the weights assigned to each value are calculated by the dot product between the query and keys in terms of the relevance between the query and each key, which allows model to capture how important each different part in the input is based on attention weights. So rather than build a brand new model and train it from scratch, the transfer learning models for natural language process tasks are mostly pre-trained with a large amount of text data and can be adapted to be fine-tuned by learned knowledge with few training data for sentiment analysis.

In the research done by Kaushik Dhola and Mann Saradva,[16] both traditional machine learning approaches and deep learning approaches, including the transfer learning model, are used to do a comparative evaluation across those models on a publicly available dataset of over 1.6 million Twitter posts, where 7,98,988 are positive tweets and 8,01,011 are negative tweets. Some essential data preprocessing techniques were used on raw data for the training step, and the dataset was split into an **80%** training set and a **20%** test set respectively for validation. As a result, SVM got **76.3%** accuracy, LSTM (traditional deep learning) got **80%** accuracy, Multi-Nominal Naive Bayes got **76.9%** accuracy, and BERT (transfer learning approach) got the best accuracy with **85.4%** compared to the others. The result of this comparison suggests that the deep learning model can perform better than traditional machine learning in sentiment analysis due to its complexity with multiple layers, while the transfer learning model can perform better than traditional deep learning like LSTM because of its property of reusing knowledge.

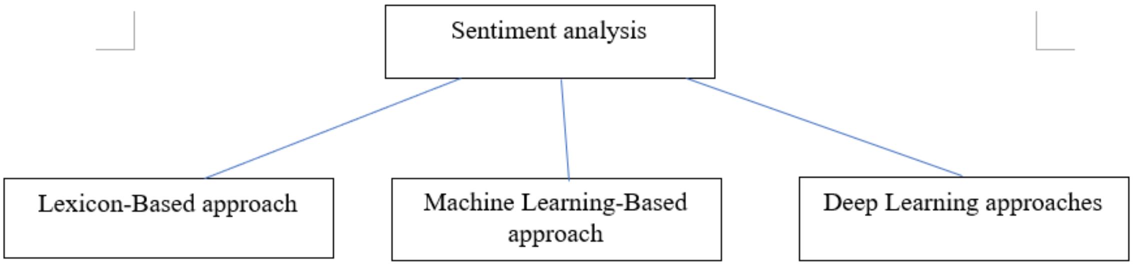


Figure 3.1: Main sentiment analysis approaches

Tool selections for sentiment analysis:

VADER: In the paper written by C. Hutto and Eric Gilbert,[17] they presented a rule-based model called VADER for general sentiment analysis, it combined with both a lexical dictionary(along with associated polarity for words and phrases) and rules: "Five general rules that embody grammatical and syntactical conventions for expressing and emphasizing sentiment intensity according to Hutton[17]". The rules are important to help correct the sentiment score based on context by identifying generalizable heuristics that can affect sentiment intensity, like punctuation, capitalization, degree modifiers, negation and conjunctions. In terms of social media text, the authors found that Vader performs extremely well compared to other sentiment analysis tools by comparison and even individual human raters.[17]

BERT(Bidirectional Encoder Representations from Transformers): BERT built on the transformer architecture can be used for many NLP tasks like text classification, question-answering, and sentiment analysis. In the paper written by Jacob Devlin et al,[18] they pointed out the limitation of standard models's unidirectional architecture in the pre-training phase (left to right) that only considering context in one direction can cause missing out of full context. To address this limitation, they introduced a new language representation model called BERT based on the transformer model. It consists of two steps: in the pre-training phase, it uses Masked Language Model(randomly mask words in the input and predicting masked words) and next sentence prediction (predict words by their context) to pretrain the model on unlabeled data over different pre-training tasks to get a deep bidirectional transformer that achieves better contextual understanding; in the fine-tuning phase, "BERT model is first initialized with the pre-trained parameters, and all of the parameters are fine-tuned using labeled data from the downstream tasks accroding to Devlin"[18]. Their models are pre-trained by BooksCorpus and Wikipedia in a total of over 3 billion words, which does not require a large labelled data-set and a self-attention mechanism in the transformer can help the model to discover the relationship between different parts in the text (contextual understanding). After pre-training phase, with the support of self-attention mechanism the model can be fine-tuned for my task by training on labelled data.[18]

RoBERTa(A Robustly Optimized BERT Pretraining Approach): In the research done by Yinhan Liu et al, they introduced a Robustly Optimized BERT Pretraining Approach based on BERT with better performance. This approach focuses on optimising the pre-training phase and can be deemed an improved version of BERT by removing the next sentence prediction and using dynamic masking on training data in the pre-training phase, "training on longer sequences, and training the model longer with bigger batches over more data" accronding to Yinhan[19]. As well as it uses a much larger corpus for pre-training compared to BERT.

Chatgpt(Chat Generative Pre-trained Transformer): Chatgpt developed by OpenAi is currently one of the most well-known large language models. This model is built based on the Generative Pre-trained Transformer architecture. As an AI model it can interact with humans in a conversational way, which includes answering human questions in detail, questioning incorrect human prompts, and refusing inappropriate human requests. [20].

NTLK: NLTK is popular platform in python dealing with textual data for NLP tasks, it provides text processing libraries including pre-built supervised machine learning models,tokenization, stemming, tagging, parsing, semantic reasoning etc.[21]

Keras: Keras is a multi-framework deep learning API offering deep learning library in Python. It provides many pre-built deep learning models that can be used for sentiment analysis.[22]

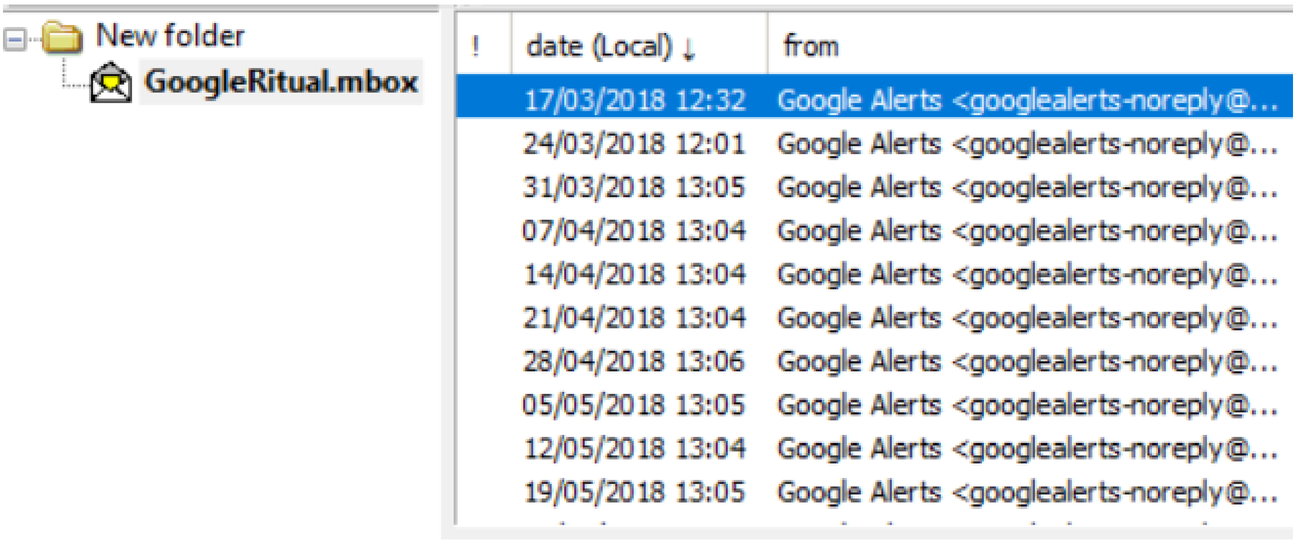
XLNet: XLNet is another transfer learning model built on the transformer architecture that can be used for many NLP tasks. In the paper written by Zhilin Yang et al,[23] they pointed out the issue of BERT in the pre-training phase that Masked Language Model masks the elements of input without considering the dependency between the masked positions, but in the real world, words usually depend on each other. Based on the pro and con of BERT they introduced XLNet, which abandoned Masked Language Model and used an improved autoregressive method in the pre-training phase. Unlike the normal autoregressive method with a unidirectional architecture, XLNet's improved autoregressive method considers all possible permutations of the factorization order in the pre-training phase. According to the experimental result in a fair setting, the authors also mentioned that XLNet outperforms BERT on 20 tasks including sentiment analysis.[23]

Chapter 4: Data Considerations

For this project, I will use the given raw database: (i) GoogleRitual.mbox that contain a list of news headlines generated by weekly reports by Google over a year; each headline contains the word 'ritual' along with corresponding tag text, author, date, and tag text. In this section, I will describe data collection including data properties and summarising essential data preparation work prior to the analysis.

4.1 Data Collection

The data i will use in this project for the sentiment analysis tool test is given in mbox format, which is a common format used to store messages. In this case, it contains all the details about Gmail messages, and each dated Gmail message contains information about headlines, tag text, authors, and urls. Therefore, there was no data collection process required for this project.

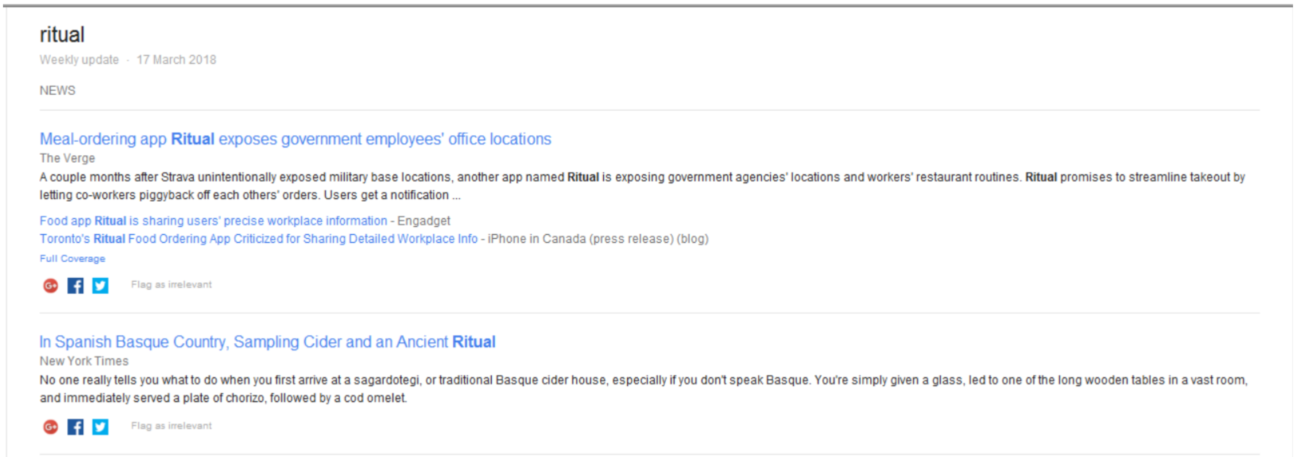


!	date (Local) ↓	from
	17/03/2018 12:32	Google Alerts <googlealerts-noreply@...
	24/03/2018 12:01	Google Alerts <googlealerts-noreply@...
	31/03/2018 13:05	Google Alerts <googlealerts-noreply@...
	07/04/2018 13:04	Google Alerts <googlealerts-noreply@...
	14/04/2018 13:04	Google Alerts <googlealerts-noreply@...
	21/04/2018 13:04	Google Alerts <googlealerts-noreply@...
	28/04/2018 13:06	Google Alerts <googlealerts-noreply@...
	05/05/2018 13:05	Google Alerts <googlealerts-noreply@...
	12/05/2018 13:04	Google Alerts <googlealerts-noreply@...
	19/05/2018 13:05	Google Alerts <googlealerts-noreply@...

Figure 4.1: View of mbox file by mbox viewer

In order to give a better understanding of this mbox file to the reader, I used the mbox viewer to present what the mbox file contains in a user-friendly way, From figure 4.1, the file contains many Gmail messages with its collected date.

And for each message in figure 4.2, there are many headlines that accompany its tag text, author, and url.



ritual

Weekly update · 17 March 2018

NEWS

Meal-ordering app **Ritual** exposes government employees' office locations

The Verge

A couple months after Strava unintentionally exposed military base locations, another app named **Ritual** is exposing government agencies' locations and workers' restaurant routines. **Ritual** promises to streamline takeout by letting co-workers piggyback off each others' orders. Users get a notification ...

Food app **Ritual** is sharing users' precise workplace information - Engadget

Toronto's **Ritual** Food Ordering App Criticized for Sharing Detailed Workplace Info - iPhone in Canada (press release) (blog)

Full Coverage

In Spanish Basque Country, Sampling Cider and an Ancient **Ritual**

New York Times

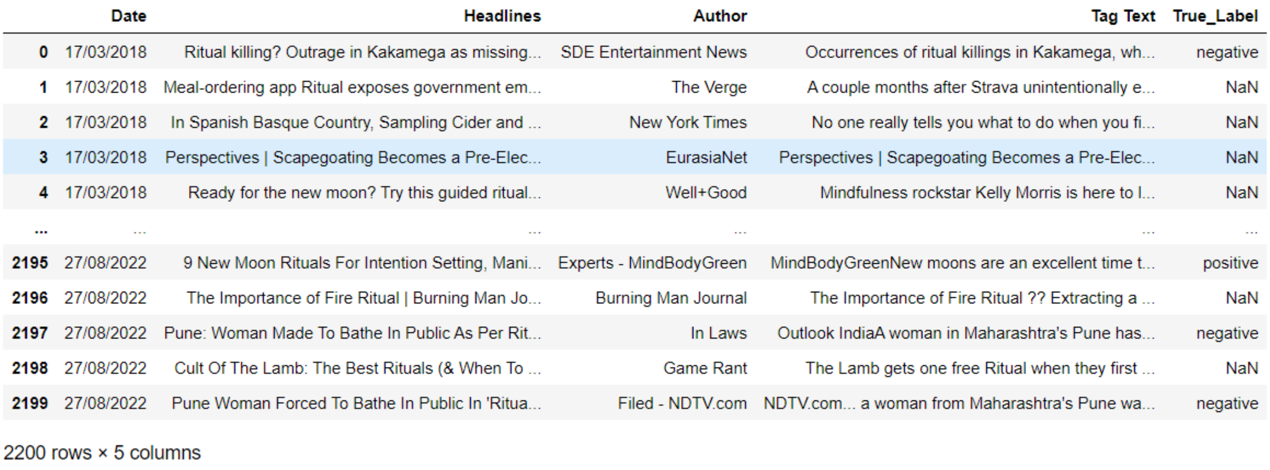
No one really tells you what to do when you first arrive at a sagardotegi, or traditional Basque cider house, especially if you don't speak Basque. You're simply given a glass, led to one of the long wooden tables in a vast room, and immediately served a plate of chorizo, followed by a cod omelet.

Figure 4.2: View of message in mbox file by mbox viewer

4.2 Data Preparation

Given a database in mbox format, data preparation is essentially required before jumping into any kind of analysis. First of all, extracting the data and parsing the dataset in an mbox file into a format that will allow me to use sentiment analysis tools on it. After proper consideration, I have chosen to extract 'Date' from each message's subject, and all the headlines with their corresponding author and tag text from the body content of the message, as those data might be useful in the analysis. In those chosen data, the headline is obviously the core data for analysis because the words used for the headline are commonly more powerful, clear, and emotional to make the headline concise and point-clear. The headline can draw the reader's attention and bring up their emotional connection to the headline, while the tag text here is a piece of text cut from the start of body content (see figure 4.2). Unlike the headline, the wording style of this tag text is more detail-oriented and informative.

As the core of this project is testing the sentiment analysis, the evaluation becomes crucial, so I have manually annotated ground truth labels for 270 headlines with clear polarity out of a total of 2200 headlines. Figure 4.3 shows the Panda data frame that I created using the extracted data with its corresponding ground truth sentiment label.



	Date	Headlines	Author	Tag Text	True_Label
0	17/03/2018	Ritual killing? Outrage in Kakamega as missing...	SDE Entertainment News	Occurrences of ritual killings in Kakamega, wh...	negative
1	17/03/2018	Meal-ordering app Ritual exposes government em...	The Verge	A couple months after Strava unintentionally e...	NaN
2	17/03/2018	In Spanish Basque Country, Sampling Cider and ...	New York Times	No one really tells you what to do when you fi...	NaN
3	17/03/2018	Perspectives | Scapegoating Becomes a Pre-Elec...	EurasiaNet	Perspectives | Scapegoating Becomes a Pre-Elec...	NaN
4	17/03/2018	Ready for the new moon? Try this guided ritual...	Well+Good	Mindfulness rockstar Kelly Morris is here to l...	NaN
...	...	...	...	...	...
2195	27/08/2022	9 New Moon Rituals For Intention Setting, Mani...	Experts - MindBodyGreen	MindBodyGreenNew moons are an excellent time t...	positive
2196	27/08/2022	The Importance of Fire Ritual | Burning Man Jo...	Burning Man Journal	The Importance of Fire Ritual ?? Extracting a ...	NaN
2197	27/08/2022	Pune: Woman Made To Bathe In Public As Per Rit...	In Laws	Outlook IndiaA woman in Maharashtra's Pune has...	negative
2198	27/08/2022	Cult Of The Lamb: The Best Rituals (& When To ...	Game Rant	The Lamb gets one free Ritual when they first ...	NaN
2199	27/08/2022	Pune Woman Forced To Bathe In Public In 'Ritua...	Filed - NDTV.com	NDTV.com... a woman from Maharashtra's Pune wa...	negative

2200 rows × 5 columns

Figure 4.3: Dataframe with ground truth labels

Chapter 5: **Outline of Approach**

Just a quick recap of my project specification in Chapter 1, my core goal is to bring the tools of sentiment analysis to the given unique data-set to see how they play out. In this section, I will outline my considered approaches and their advantages and potential issues because of the existing data limitations in this project.

Consideration of the approaches being used in this project:

Lexicon-based approach: As discussed in the related work about lexicon-based approach, the potential limitations about ambiguity and contextual understanding can appear for a lexicon-based approach, so the minimum expectation would be kept for this approach, but it has the advantage of requiring no labelled training data to use. The tool called "VADER" combined both lexicon-based and rule-based techniques, has shown considerable accuracy in terms of social media text as mentioned in the authors's paper [17]. So it can be worth using it as a baseline compared to other advanced approaches.

Machine learning-based approach: The limitation of traditional supervised machine learning-based approaches using Naive Bayes, Support Vector Machine, Logistic Regression, Decision Tree, and KNN is quite clear as they need a massive amount of good-quality data to build a reliable model. The given unique data-set has around 2200 entries without label class, even though I can label the entries that have extreme polarity for final testing, but those manually labelled entries can be insufficient for the model to train and test, so this can be a serious challenge for machine learning-based approaches. The possible way of addressing this challenge can be using the public labelled sentiment data-set provided by various sources like Hugging Face, NLTK or even kaggle to train the model. Another method can possibly be helpful in this project due to scarcity of labelled data is using semi-supervised way. In the paper,[24] the semi-supervised learning can be used in sentiment analysis against the scarcity of labelled data. This method separates the labelled data into training (some labelled and remaining unlabelled data) and testing (only labelled data), then starts with training the deep learning model with a small amount of labelled training data, uses the trained model to predict label class for the remaining unlabelled data, and finally combines the majority of predicted label data with a minority of true label data together to train the model again, which saves the effort of human labelling. From the experimental results of this paper, the method can help with the training model, but need to be careful to deal with the model-labelled data, as bad-quality data that fed into the model can hugely affect the performance. The tool NTLK platfrom provides the libraries for directing using pre-built supervised learning models.

Deep learning approaches: Traditional deep learning approaches for sentiment analysis could be better than supervised learning approaches but also require a sufficient amount of good-quality data to train. The Transformer Based Models using pre-training and transfer learning approaches seems to offer a way to deal with this challenge because of its properties of pre-training, reusing knowledge, and adapting to be fine-tuned with a fair amount of labelled data for the task. Although the transfer learning models for natural language processing are pre-trained with a large corpus of text, they still need to be fine-tuned by feeding textual data for specific natural language processing tasks, like sentiment analysis in this case. The possible ways that I listed in the discussion about machine learning-based approach can also be used here, which are feeding the public-labelled sentiment data-set or using semi-supervised learning. The tool Keras is available on TensorFlow, in which i can directly import to use pre-built deep learning models. The transfer learning models for natural language processing like RoBERTa, BERT, XLnet are available to use by importing Hugging Face Transformers libraries on python.

Final confirmation of the tools being used in this project:

1. Vader: A sentiment analysis tool specialized in social media text and combined with both lexicon-based and rule-based approaches.
2. Transformer Based Models: Models based on transformer architecture that pre-trained with large text corpus and fine-tuned by textual data for sentiment analysis task.
3. Chatgpt4: Chat Generative Pre-trained Transformer 4 is one of the most famous and latest large language models. The old version was initially created for generating human-like text based on the received input, but the current version now comes with more capabilities beyond text generation; it can be used for many different natural language processing tasks, including sentiment analysis.

Text pre-processing for sentiment analysis tools(if required by the tools):

1. Vader: Vader provides a package that can be used to create an instance of the sentiment analyzer. The built-in function of this sentiment analyzer instance 'polarity scores()' can directly take the text input and return a sentiment score.
2. Transformer Based Models: Transfomers package provides various pre-trained and fine-tuned models for sentiment analysis, those models all require tokenizing the input text to feed in so they can output the desired score for sentiment classification. Transfomers package provides a tokenizer for each of those models that automatically handles all the required pre-processing, so i just need to feed the raw text into their tokenizer, and then the output of the tokenizers can be fed into the model for sentiment analysis.
3. Chatgpt4: OpenAi provides API access to Chatgpt4, so there is no need to do any text pre-processing as it is going to interact with user in conversational way.

Hyper-parameters setting for sentiment analysis tools (if required by the tools):

1. Vader: Vader sentiment package provides a way to directly access vader sentiment analyzer using an API, so hyper-parameter settings is not necessary required here.
2. Transformer Based Models: The Transformers package also provides a way to directly access those fine-tuned models with proper parameter settings using an API, so there is no need to do any hyper-parameter settings for the models. Some models might be trained and fine-tuned for multi-class sentiment analysis, but because my ground truth label is binary, it can require some adjustments to outputs.
3. Chatgpt4: It does not need any particular hyper-parameter setting, as introduced, it interacts with the user in a conversational way, but the message that consists of prompt + text data (input) is required in order to get the sentiment label (output) back from chatgpt4.

Chapter 6: Project Workplan

Apart from the approaches that deal with the core plan, a well-structured work plan throughout the entire project is important. In this section i will present a detailed work plan for the project including dependencies, and evaluation strategies.

The list of tasks required to complete the project, along with the time allocation, for each task the dependency is in sequential order from top to bottom.

1. Data extraction: This task involves obtaining raw data from a source for sentiment analysis. A unique raw database of news headlines collected by Google based on the occurrence of the word "ritual" in the headline is given for sentiment analysis. As discussed the potential issue of the scarcity of labelled data for the sentiment analysis tools, importing the public data-set for sentiment analysis from libraries could be considered in the analysis phase.

2. Data preprocessing: The given raw database in mbox format first needs to be converted into pd dataframe containing dates, headlines, authors, and tag text for sentiment analysis, followed by necessary data cleaning steps like handling missing values and duplicated values.

3. Ground truth label: The data-set has no ground truth label. In order to achieve the ultimate goals in this project, the ground truth label is mandatory to compare with the results of the sentiment analysis tools classified to know whether the current sentiment analysis tools can reach human judgemental levels in terms of sentiment analysis. The total number of headlines in the data-set is around 2200. Manual labelling is time-consuming, so I will label those headlines with a very clear polarity that can be easily determined as positive or negative by human inspection.

4. Identifying the sentiment analysis tools: As discussed in chapter 5.

5. Applying sentiment analysis tools: In the analysis phase, apply each of the confirmed sentiment tools to the data-set for sentiment classification. Use the imported public sentiment analysis data-set and text pre-processing like tokenization and transformation(such as vectorization) to help build robust tools if required by the sentiment analysis tools.

6. Evaluating the sentiment analysis tools: As sentiment analysis is a classification task, the evaluation metrics that can be used in the analysis are F-score, Precision, Recall, and Accuracy for evaluating the results of the sentiment analysis tools against the ground truth label, as well as comparative analysis across different tools. Finally, evaluating used sentiment analysis tools based on the results helps judge their capabilities, limitations, or biases, which can contribute to the sentiment analysis field.

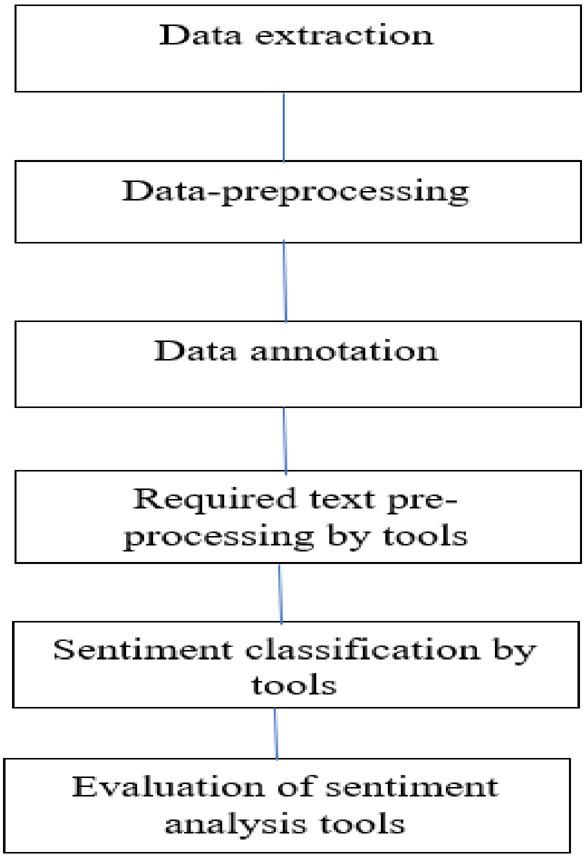

Data extraction
Data-preprocessing
Data annotation
Required text pre-processing by tools
Sentiment classification by tools
Evaluation of sentiment analysis tools

Figure 6.1: Work plan

Chapter 7: Summary and Conclusions

In this section, I will present the results of the analysis for each of the sentiment analysis tool tests and draw insights from comparative analysis across all of the tools used. In each sentiment analysis tool test, I describe the datasets and methods used, along with an interpretation of the results.

7.1 Data & Evaluation metrics

The dataset used for all of the tools: I used a cleaned dataset with ground truth labels (see figure 4.3 in the section on data consideration): As the number of ground truth label is limited, so this cleaned dataset is filtered to contain entries with ground truth label only for performing analysis and evaluation.

Evaluation metrics: As it is classification task, the following common evaluation metrics[25] would be used for all of the tool evaluations:

1. Accuracy: The percentage of correctly classified sentiment made by the tool.

$$Accuracy \;= \frac{Total\ correctly\ classified\ examples}{Total\ number\ of\ examples} \tag{7.1}$$

2. Precision: It measures the accuracy of positive predictions.

$$Precision \;= \frac{True\ positive}{True\ positive + False\ positive} \tag{7.2}$$

3. Recall: It measures the ability of the model to identify all the true positives.

$$Recall \;= \frac{True\ positive}{True\ positive + False\ Negative} \tag{7.3}$$

4. F1 Score: The balanced score between precision and recall. As my ground truth labels are imbalanced with 177 positive samples and 93 negative samples, the F1 score can be appropriate for an imbalanced class in this case.

$$F1 \;= 2 * \frac{precision\ * recall}{precision\ + recall} \tag{7.4}$$

7.2 Tool1: Vader sentiment analysis tool

As discussed in the related work, Vader is a sentiment analysis tool that uses both lexicon (all the words or phrases come with an associated polarity score) and rules to classify the sentiment of the input text. The intensity of sentiment would be considered and determined by following five rules:

Punctuation: In the case of the exclamation point, it can increase the sentiment intensity without changing sentiment orientation and capture more precisely the sentiment, such as "I am angry!" would be more intense than "I am angry.".

Capitalization: The word capitalized with all letters would be more intense than the words that is not capitalized, such as "I am ANGRY." would be more intense than "I am angry.".

Degree modifiers: The intensity can be affected up down by degree adverbs, such as "I am fucking angry." would be more intense than "I am angry.", "I am slightly angry." would be less intense than "I am angry.".

Negation: The intensity can be flipped by negation, so the sentiment polarity of a word would be flipped after following the negation, such as 'not happy' can flip sentiment into positive polarity from negative negative polarity.

Conjunctions: It considers the conjunction word like 'but', which can shift the sentiment polarity of the text following the conjunction word, such as "Money is good, but it can be used for killing". so the latter half dominates the overall score.

7.2.1 The methods used for test of Vader:

1. Threshold setting for score mapping: Refer to some code from the Vader official documentation [26] for API access. For each headline and tag text, I called its library built-in function 'polarity scores', which directly outputs sentiment polarity scores by passing in input text.

For each input text, the 'polarity scores' function returns a dictionary that contains four different polarity scores[26]:

1. neg: how likely is this text classified as negative
2. neu: how likely is this text classified as neutral
3. pos: how likely is this text classified as positive
4. compound: The calculated overall sentiment score is calculated by summing the polarity scores of each word in the lexicon and adjusted according to the rules, which take values from -1 (most negative) to 1 (most positive).
5. The neg, neu, and pos scores are probabilities that the text is classified as each sentiment, those three values are added up to 1.

The output sentiment probability scores needed to be mapped to sentiment labels, From [27] author of Vader recommends the standard threshold to classify the text as either positive, negative, or negative, as used in the literature cited on 'About the Scoring Page' as follows:

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

Since my ground truth label only contains two classes, 'positive' and 'negative', and there are no examples like slightly positive and slight negative when I did annotation because all the ground truth labelled examples have shown the clear polarity that can be inspected at human glance, I will slightly alter the threshold to accommodate my ground truth table for my analysis.

1. positive sentiment: compound score $>= 0$
2. negative sentiment: compound score < 0

2. Evaluation and analysis:

1. Evaluate the performance of tool by mentioned metrics to see if it can correctly identify the sentiment behind those news headlines.
2. Error analysis on misclassified examples to find out any insight from where the tool fails.

7.2.2 Results

1. Evaluation:
The table 7.1 shows the result of evaluation metrics produced by comparing Vader classified labels against ground truth labels for headline text and tag text.

1. Accuracy: The accuracy for headline text is about 82%, which is a relatively high accuracy rate and suggests that Vader correctly classified 82.2% examples out of total. The accuracy of tag text decreased to 0.77% compared to headline text, which suggests that the tool performs worse at correctly predicting the sentiment for tag text.
2. Precision: The precision for headline text is about 67%. This suggests that it correctly classified 67% examples out of predicted positive examples. The accuracy of tag text decreased to 61% compared to headline text, which also suggests its disadvantage in predicting sentiment for tag text and both precision metrics of headline and tag text show the tool's weakness at classifying negative examples.
3. Recall: The recall for both headline and tag text is quite high at 95% and 89% respectively. This suggests that it is very good at predicting true positive examples, which conversely leads to lower precision as seen. Again, the tag text recall is less than headline recall.
4. F1-score: The F1-score is an effective metric that handles imbalanced classes. In this case it gives a balanced score between precision and recall. The F1-score for headline at 78% is relatively high, while the tag text F1-score is less than the headline F1-score, which is the same as the rest of the three evaluation metrics.

From the metrics table, we can see that Vader appears to perform poorly comprehensively with tag text than headline across all evaluation metrics, almost 7% lower for each of the metrics. The tool shows high recall and relative lower precision, which indicate better capability to identify positive sentiment but come along with more false positive predictions. The tool showing a relative high F1-score and accuracy presents its fair capability in sentiment analysis.

	Metrics	Headline text	Tag text
0	Accuracy	0.822222	0.77037
1	Precision	0.671756	0.614815
2	Recall	0.946237	0.892473
3	F1-score	0.785714	0.72807

Table 7.1: Evaluation result

2. **Error analysis:**
In this analysis, I filtered the dataframe to get one that only contains misclassified examples for error analysis.

From figure 7.1, it shows the top 20 most common words appear in the corpus for all misclassified false positive headlines. Those common words might give an indication that the most common words might lead the tool to an incorrect classification. Obviously, 'ritual' is the most commonly appearing word, followed by 'sacrifice', 'money'', and 'satanic'. Typically a human being can immediately sense the negative sentiment from words 'sacrifice' or 'satanic', and infer that any combination of words 'satanic', 'sacrifice' like 'satanic ritual', 'sacrifice ritual' and 'money ritual' can absolutely lead to negative sentiment, while this tool fails.

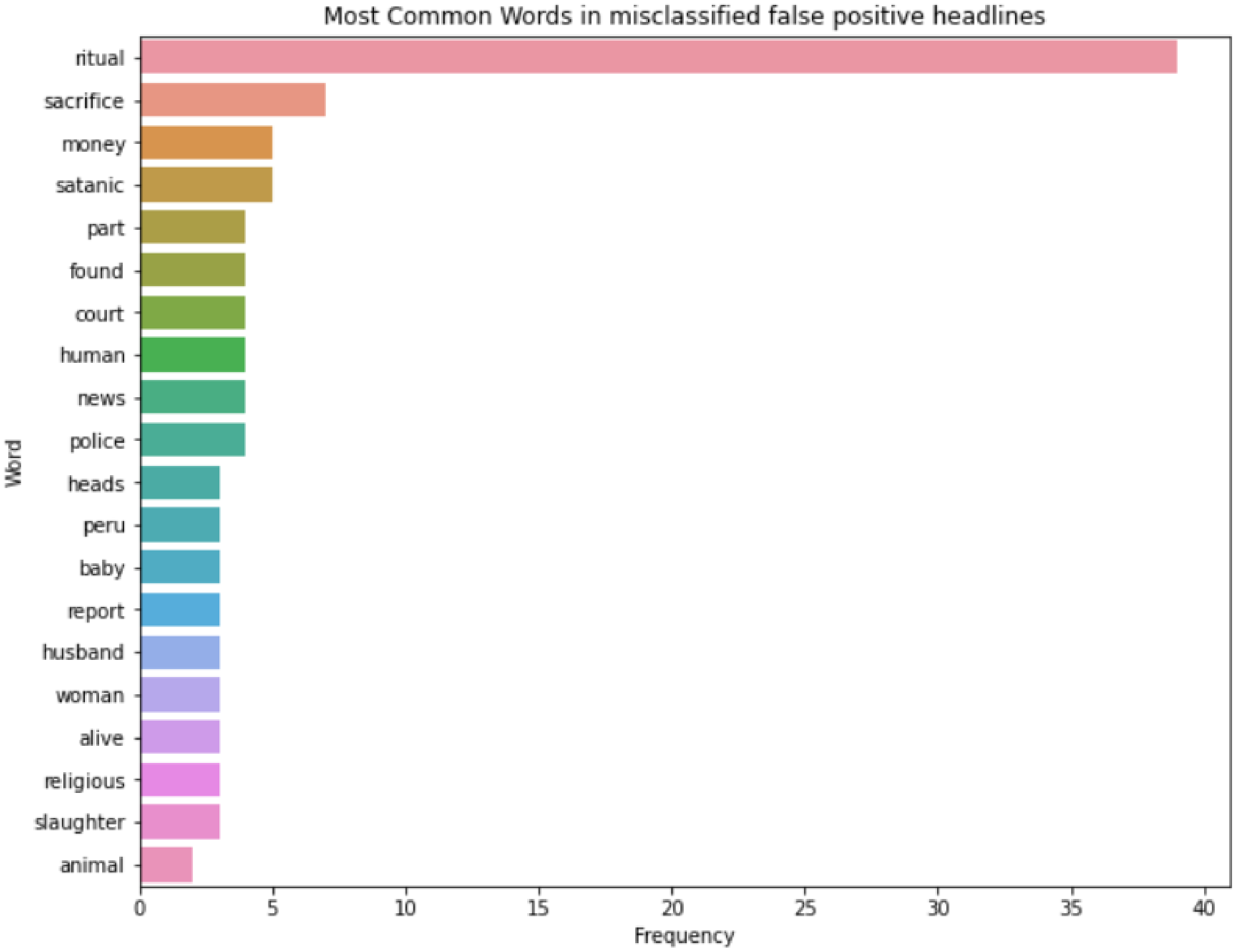


Figure 7.1: Most Common Words in misclassified false positive headlines

As discussed from common words analysis, Vader might have failed to classify some of the common words that have a clear polarity for human beings in misclassified false positive headlines.

From table 7.2, it shows the classified sentiment labels by Vader for the 20 most common words. Interestingly, those key words with extreme polarity, like "satanic", "sacrifice" have been incorrectly classified as neutral compared against human feeling. Such incorrect classification on a single word can lead entire headlines to be classified incorrectly because of the nature of this tool, which relies on the polarity score of each word from the lexicon for sentiment score calculation. This issue can be caused by a missing word and its corresponding polarity in the lexicon.

	Common words	Sentiment label
0	ritual	Neutral
1	sacrifice	Neutral
2	money	Neutral
3	satanic	Neutral
4	part	Neutral
5	found	Neutral
6	court	Neutral
7	human	Neutral
8	news	Neutral
9	police	Neutral
10	heads	Neutral
11	peru	Neutral
12	baby	Neutral
13	report	Neutral
14	husband	Neutral
15	woman	Neutral
16	alive	Positive
17	religious	Neutral
18	slaughter	Neutral
19	animal	Neutral

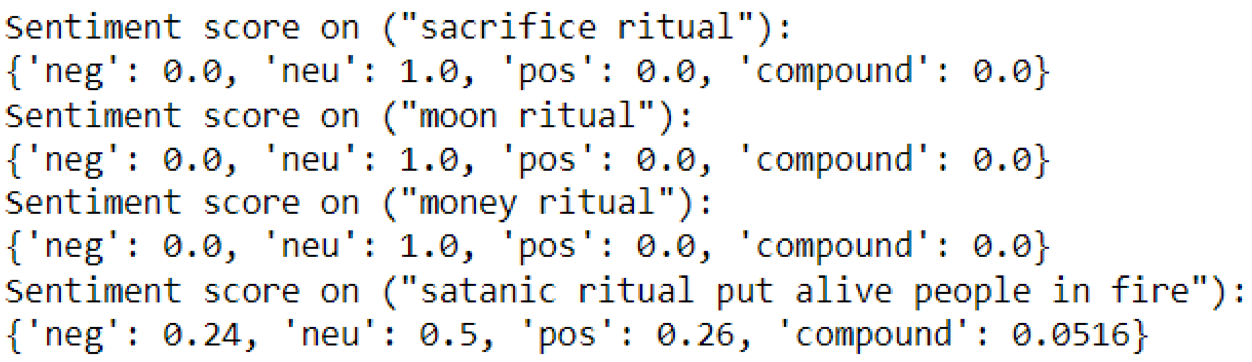
Table 7.2: The sentiment of 20 common words in False Positive examples

From table 7.3, it identifies frequent 2-grams in misclassified headline examples, the indexes of 0, 1 and 2 are frequent 2-grams from false positive examples, and index 3 from false negative. Those two word combinations are most common in misclassified headlines. Again, just like what we discussed in common word analysis, those first three 2-grams have a clear polarity for human-beings to inspect as negative, and the last 2-grams 'moon ritual' as positive.

	n-gram	Frequency
0	(money, ritual)	4
1	(satanic, ritual)	4
2	(sacrifice, ritual)	3
3	(moon, ritual)	2

Table 7.3: N-grams table

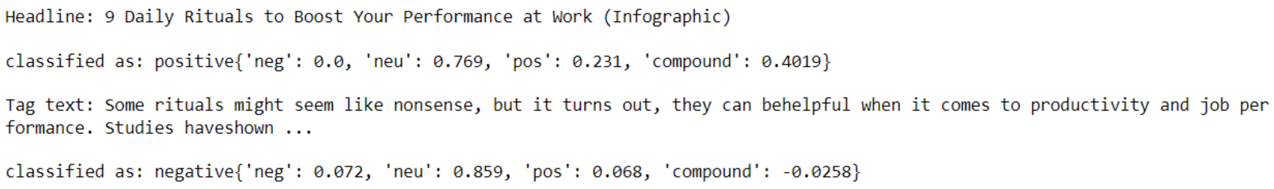
From figure 7.2, it shows returned dictionaries of sentiment scores for the three mentioned 2-grams: 'money ritual', 'sacrifice ritual', and 'moon ritual', the result shows all three grams with clear polarity for human-being inspection are all classified incorrectly. The returned sentiment score for sentence 'satanic ritual put alive people in fire' clearly proven the pitfall of lexicon-based like Vader that when analyzing the sentiment of a text where crucial words in the lexicon have no corresponding score record, this can lead to incorrect sentiment classification for the whole sentence. In this case, the key words 'satanic ritual' with a neutral score, 'alive' with a positive score, as seen in common word analysis, and 'fire' with a negative score slightly less than 'alive', which after calculation put the whole sentiment as positive slightly over 0 from the compound score.



```
Sentiment score on ("sacrifice ritual"):
{'neg': 0.0, 'neu': 1.0, 'pos': 0.0, 'compound': 0.0}
Sentiment score on ("moon ritual"):
{'neg': 0.0, 'neu': 1.0, 'pos': 0.0, 'compound': 0.0}
Sentiment score on ("money ritual"):
{'neg': 0.0, 'neu': 1.0, 'pos': 0.0, 'compound': 0.0}
Sentiment score on ("satanic ritual put alive people in fire"):
{'neg': 0.24, 'neu': 0.5, 'pos': 0.26, 'compound': 0.0516}
```

Figure 7.2: The sentiment score by Vader for frequent 2-grams

The tag text as part of the text body that comes with the headline consistently shows a lower value in all of the evaluation metrics compared against the evaluation result headlines and even classified the opposite sentiment of its headline. From figure 7.3, it shows the classified sentiment and sentiment scores for the headline and its tag text. By human judgement, the tag text should be classified as positive, but Vader did the opposite of its prediction in the corresponding headline, and we know long text often composes multiple sentences, even paragraphs, that depend on each other to derive an argument, which suggests its obvious weakness as lexicon-based and rule-based that it can not capture the context and coherence of composed sentences or paragraphs as well as mixed sentiment across tag text, so leading to misinterpretation of the overall sentiment. And the reason for better performance on headline analysis can be that the headlines are composed with a relatively small number of key words for concisely passing the crucial information, which can provide a more evident sentiment polarity, while the tag text is short text cuts from body content, and this can lead to very sparse sentiment polarity to make it difficult to classify.



```
Headline: 9 Daily Rituals to Boost Your Performance at Work (Infographic)

classified as: positive{'neg': 0.0, 'neu': 0.769, 'pos': 0.231, 'compound': 0.4019}

Tag text: Some rituals might seem like nonsense, but it turns out, they can behelpful when it comes to productivity and job per
formance. Studies haveshown ...

classified as: negative{'neg': 0.072, 'neu': 0.859, 'pos': 0.068, 'compound': -0.0258}
```

Figure 7.3: Headline text vs Tag text

7.3 Tool2: Transformer Based Models for sentiment analysis

As discussed in the related work, transformer based model using transfer learning approaches is state of art for natural language processing in recent year. From article[28], it elaborates the steps that transformer model takes for NLP task in generalized way, in which it takes pre-processed input data like a sequences of tokens depending on models's requirement, pass it into a series of layers that consist of self-attention mechanisms and feed-forward neural networks. Finally apply an activation function like softmax on logits (raw numerical scores generated by output layer of a neural network for each classes) to get mapped probabilities ranging between 0 and 1 for final classification in my case for sentiment analysis task.

The article[29] written by Huggingface specifically details the transformer models in terms of natural language processing, the transformer models for NLP as language models are all trained on large amounts of raw text data in such a way that targets are automatically computed from the inputs of the model, so-called self-supervised learning, the reason to do it is because pretrain model can be advantageous of knowing optimized parameter (learned from previous raw text data) set, these parameter is the weight value model can changes as it learns to minimise error between prediction and ground truth label, optimized parameter can help with achieving the correct output faster than building from scratch. Since different NLP tasks can have different purposes, the general pretrained model has to go through a process called transfer learning, in which the model is trained or 'fine-tuned' using annotated ground truth labels by humans on the given task, which process would benefit by requiring way less data due to the pretrained property.

The general architecture of transformer model [29]:

1. Two blocks: Both the encoder and decoder consist of attention layer and a feed-forward neural network. The encoder takes input data to produce a set of dynamic vectored representations or embeddings (as mentioned in related work), what it mean by dynamic representation is that each of such vectors corresponds to an input token but dynamically changes the value of that input token in the embedding based on the word's context within a sentence, for example the word 'hood' can mean differently in 'jacket hood' and 'cooker hood' so the embeddings for "hood" in these two sentences would differ dynamically based on the word around. The decoder takes the input sequence and generates the output sequence by passing the input sequence through self-attention mechanisms, which repetitively predict the next token based on the previous tokens it has generated.

2. Attention layers: It is the key feature for transformer models. As discussed in the related work, this attention mechanism allows the model to pay attention to the importance of each word or word with another word in the sentence based on dynamic attention weights, which can help with the issue of word context because the meaning of a word can be affected by any adjacent words.

3. Decoder only and Encoder only: Both blocks mentioned above can be used individually based on the task. The encoder only model also called auto-regressive model with encoder only built-in is suitable for tasks tending to text understanding like sentence classification tasks or named entity recognition, because such models often have “bi-directional” attention, which capture the relationship between words in both directions by the attention layer. The pretrain phase of those models often involves word masking to task the model to find masked word or reconstruct the original sentence. Decoder only model also called auto-regressive models model with decoder only built in is suitable for tasks like text generation, because at each stage the model only generate one word based on the words that come before it in the sequence in forward direction. The pretrain phase of those decoder only models often involves predicting the next word in the sentence.

7.3.1 The methods used for test of transformer models:

1. Model selection: The research work pointed out that the encoder only models are good at classification tasks and Bert is a typically popular encoder only model, so i have chosen two Bert based models called RoBerta (introduced in the related work) and distilbert (smaller, faster and lighter version of BERT[30]), both of them come with their own adjustments for improvement.

Building the model from scratch or fine-tuning the model by the own data is not realistic since the annotated data is quite limited, so i have used three open fine-tuned models for sentiment analysis on Hugging face:

1. Siebert: According to the brief documentation[31][32], this model is Roberta based but the version that further fine-tuned and evaluated on 15 datasets from diverse text sources to improve generalization across different types of texts (reviews, tweets, etc.), so this pretrained and fine-tuned models would be suitable for me to do test. This model also claimed to be better than DistilBERT-based model that trained on only one type of text like movie reviews from the popular SST-2 benchmark when used on new or unseen data.
2. DistilBERT base uncased finetuned SST-2[33]: This model is also BERT based but distilled version of BERT that features smaller, faster, cheaper, and lighter properties. The model fine-tuned by the popular Stanford Sentiment Treebank(sst2) corpora for sentiment analysis, the sst2[34][35] is a corpus with fully labelled parse trees for each sentence. The corpus consists of 11855 single sentences extracted from movie reviews; a total of over 215,000 phrases have been extracted from the parse trees of these sentences, and each one has been annotated with sentiment labels by three human judges. Since Siebert claimed to perform better than it and the 'ritual' headlines are a type of text completely different from movie reviews, tweets, etc. it is interesting to compare their performance on the unique dataset of this project.

2. Model accessing and output score mapping: Transformers library by Hugging Face[29] offers the built-in function 'pipeline', which takes three parameters: the type of NLP task, the path to the desired model, and the tokenizer associated with the model. The created sentiment analysis pipeline function can directly output classified sentiment with its polarity score by passing in input text. The processes behind pipeline function can be broken down into three main steps:

1. Preprocessing: Recall that the transformer model need to convert the raw textual data into vector representation, so in this case the tokenizer splitting the input text into words, subwords, or symbols as tokens, map each token to an integer, and add additional input that might be useful.
2. Pass the preprocessed input through the model: Feed preprocessed input into model to produces logits.
3. Post preprocessing: Map the logits produced by model into target class.

Output score mapping:

1. Siebert and DistilBERT base uncased finetuned SST-2: Both models are fine-tuned for binary classification, which produces the probabilities of being classified as positive and negative respectively after the output logits processed by the softmax function. My ground truth labels only contain positive and negative labels, so i can directly use pipeline function to get the classified sentiment for each input.

3. Evaluation and analysis:

1. Evaluate the performance of models by mentioned metrics to see if it can correctly identify the sentiment behind those news headlines.
2. Error analysis on misclassified examples to find out any insight from where it fails.

7.3.2 Results:

1. Evaluation:

The table 7.4 shows the result of evaluation metrics produced by comparing both Siebert and DistilBERT classified labels against ground truth labels for both headline text and tag text.

1. Accuracy: The Siebert accuracy for headline text is about 93%, which is quite high and suggests that Siebert correctly classified 93% examples out of total, while the DistilBERT accuracy decreases to about 91% compared to Siebert. DistilBERT has a slightly better accuracy of 81% than Siebert accuracy of 80% in terms of tag text. The accuracy of tag text for both models decreases compared to their headline text and this suggests that both models perform worse at sentiment prediction for tag text.

2. Precision: The Siebert precision for headline text is about 84%. This suggests that it correctly classified 84% examples out of predicted positive examples, while the DistilBERT has better precision increasing to about 90% compared to Siebert. DistilBERT has a better precision than Siebert precision in terms of both headline text and tag text showing Siebert's weakness at classifying negative examples, while DistilBERT is better. The precision of tag text for both models decreases compared to their headline text and suggest its weak capability in predicting sentiment for tag text.

3. Recall: The Siebert recall for headline text is very high at 99%, while DistilBERT recall is lower at 81%. This suggest that Siebert is extremely good at predicting true positive examples, while DistilBERT is much weaker. Again, the recall for tag text is lower than the recall for headline text.

4. F1-score: The F1-score gives a balanced score between precision and recall. Siebert F1-score for headline text is quite high at 91%, while DistilBERT F1-score is lower at 86% compared to Siebert. The F1-score for tag text is lower than the F1-score for headline text, which is the same as the rest of the three evaluation metrics.

From the metrics table, we can see the same situation as Vader on both models: the tool performs poorly comprehensively with tag text than headline text across all evaluation metrics. Both models showing high F1-scores and accuracy scores demonstrate their good capability in sentiment analysis. Their precision and recall results also show their strengths differently, Siebert with higher recall and relative lower precision indicates better capability to identify positive sentiment compared to DistilBERT but comes along with more false positive predictions, while conversely DistilBERT with higher precision and lower recall indicates better capability to identify negative sentiment compared to Siebert but comes along with more false negative predictions. As mentioned in the documentation of Siebert[31], it claims to perform better than DistilBERT due to using diverse text sources in the fine-tune phase improving generalization across different types of texts for sentiment analysis, while DistilBERT uses a movie review dataset called SST2. The result of the evaluation metrics shows Siebert having both a higher accuracy and F1-score than DistilBERT on this unique dataset, which indicates the importance of training data used during training will directly affect the model's performance and strength focus.

	Metrics	Headline text	Tag text
0	Siebert Accuracy	0.933333	0.803704
1	Siebert Precision	0.844037	0.66129
2	Siebert Recall	0.989247	0.88172
3	Siebert F1 Score	0.910891	0.75576
4	DistilBERT-SST-2 Accuracy	0.907407	0.814815
5	DistilBERT-SST-2 Precision	0.904762	0.786667
6	DistilBERT-SST-2 Recall	0.817204	0.634409
7	DistilBERT-SST-2 F1 Score	0.858757	0.702381

Table 7.4: Siebert and DistilBERT-SST-2 evaluation result

2. Error analysis:

In this analysis, I filtered the dataframe to get one that only contains misclassified examples for error analysis. Both Siebert and DistilBERT are the same based on transformers, so their disadvantages remain the same. I will use the dataframe of Siebert to do this error analysis as it has better performance.

From figure 7.4, it shows the top 20 most common words appear in the corpus for all misclassified false positive headlines. Those common words might give an insight that any common words involved in incorrect classification. Obviously, 'ritual' is the most commonly appearing word, followed by 'police', 'child', 'money', 'satanic', 'ogun', 'nigeria' and 'killings'. Typically a human being can immediately sense the negative sentiment from words 'killings' or 'satanic', and infer that any combination of words 'satanic', 'killings' like 'satanic ritual', 'ritual killings' can lead to negative sentiment, while this tool fails.

As discussed from common words analysis, Siebert might have failed to classify some of the common words that have a clear polarity for human beings in misclassified false positive headlines.

From table 7.5, it shows the classified sentiment labels by Siebert for the 20 most common words. Those key words with extreme polarity, like "satanic" and "killings" have been correctly classified as negative, which match human feeling. Something interesting was discovered here on two same-type words: "nigeria" and "poland" both stand for country, however "nigeria" is classified as negative while "poland" is classified as positive, which immediately raises an issue of bias. Either "poland" or "nigeria" as a single word incurs no bad sentiment. The reason such a bias issue can be raised is because of the nature of deep learning models, which rely on a large amount of training data to optimise the parameters of the model. As is known, those transformer models are pretrained on a large amount of internet data and if the data used for pretraining or fine-tuning is not selected carefully, the model can learn bias. In this case, if the training data used involving african country "nigeria" has been dominantly mentioned in negative contexts while the training data involving the European country "poland" is dominantly mentioned in positive or neutral contexts, this data imbalance can lead to a biased classification.

	Common words	Sentiment label
0	ritual	Negative
1	police	Positive
2	child	Negative
3	money	Negative
4	satanic	Negative
5	nigeria	Negative
6	killings	Negative
7	poland	Positive

Table 7.5: The sentiment of 8 common words in False Positive examples

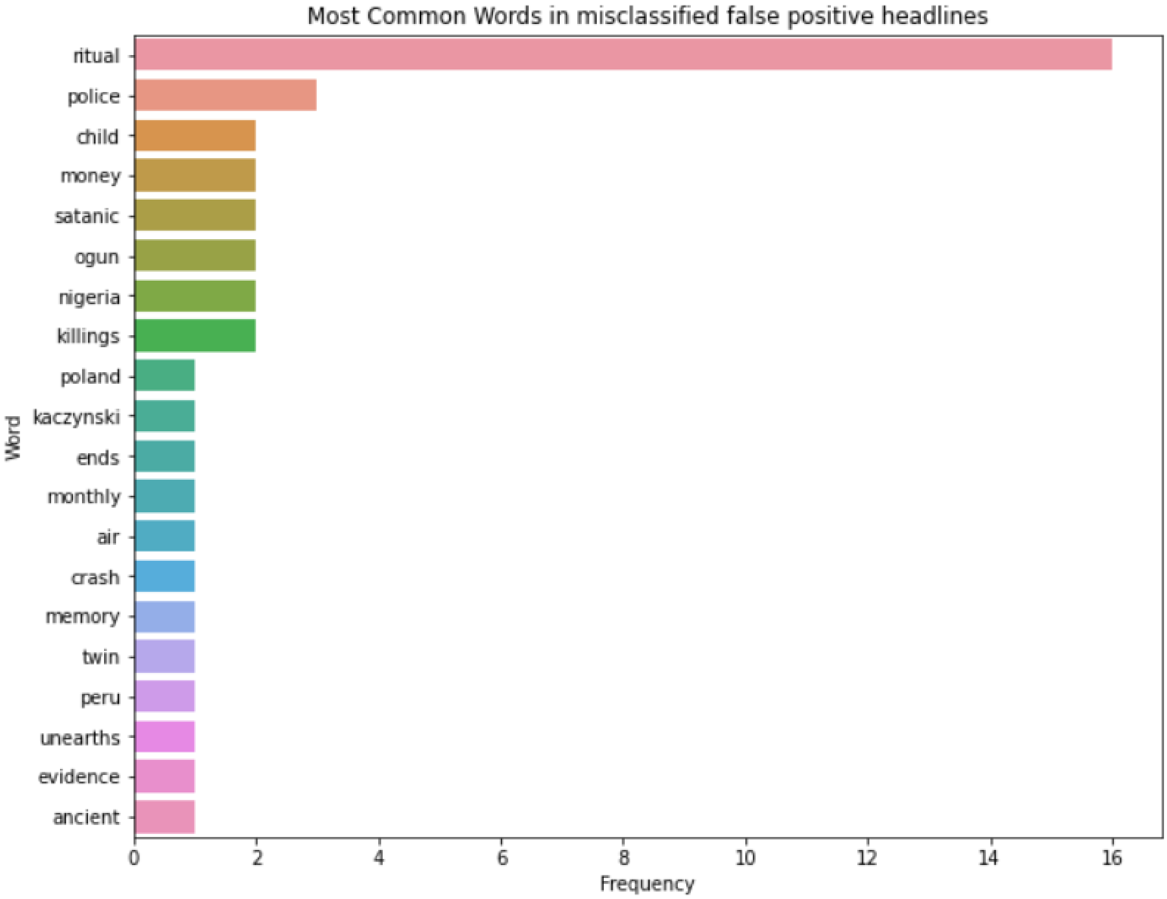


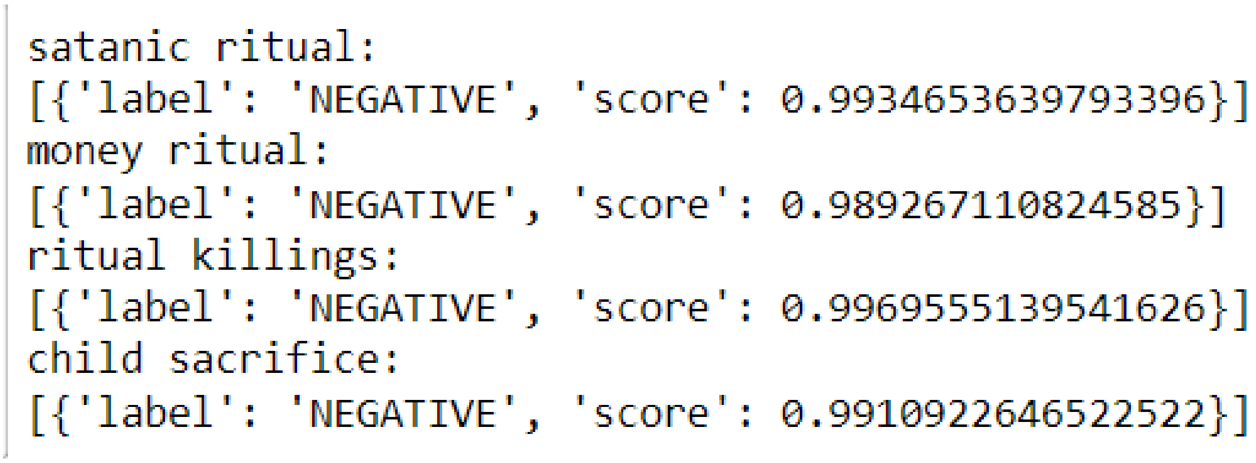
Figure 7.4: Most Common Words in misclassified false positive headlines

From table 7.6, it identifies frequent 2-grams in misclassified headline examples, the indexes of 0, 1, 2 and 3 are frequent 2-grams from false positive examples. Those two word combinations are most common in misclassified headlines and they have a clear polarity for human- beings to inspect as negative.

	n-gram	Frequency
0	(ritual, killings)	2
1	(satanic, ritual)	2
2	(money, ritual)	1
3	(child, sacrifice)	1

Table 7.6: N-grams table

From figure 7.5, it shows returned dictionaries of sentiment for four mentioned 2-grams:'money ritual', 'child sacrifice', 'satanic ritual' and 'ritual killings'. The result shows all four grams with clear polarity are correctly classified by Siebert. So what makes the model predict wrong when those 2-grams comes to the whole headline?



```
satanic ritual:
[{'label': 'NEGATIVE', 'score': 0.9934653639793396}]
money ritual:
[{'label': 'NEGATIVE', 'score': 0.989267110824585}]
ritual killings:
[{'label': 'NEGATIVE', 'score': 0.9969555139541626}]
child sacrifice:
[{'label': 'NEGATIVE', 'score': 0.991092264652252}]
```

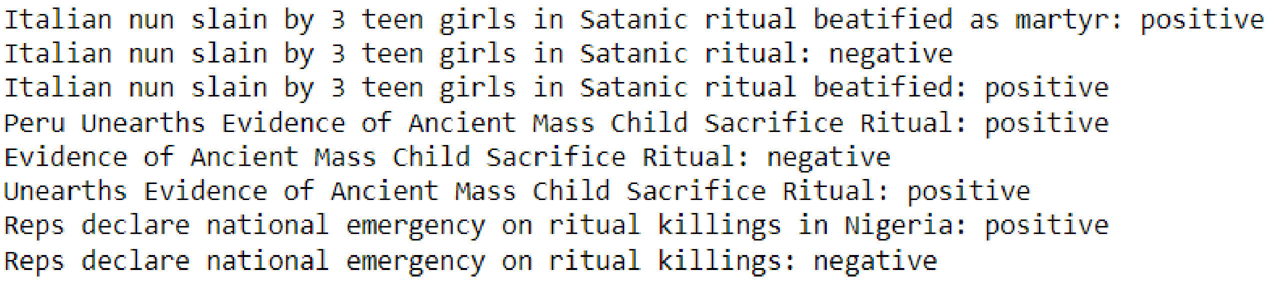
Figure 7.5: The sentiment score by Siebert for frequent 2-grams

From figure 7.6, it shows three misclassified false positive headlines, for each of the misclassified false positive headlines contains its corresponding correctly classified 2-gram mentioned previously. By removing one to three irrelevant words(or the combination of words) to the core from three sentences "Italian nun slain by 3 teen girls in Satanic ritual beatified as martyr", "Peru Unearths Evidence of Ancient Mass Child Sacrifice Ritual" and "Reps declare national emergency on ritual killings in Nigeria", the truncated sentences "Italian nun slain by 3 teen girls in Satanic ritual", "Evidence of Ancient Mass Child Sacrifice Ritual" and "Evidence of Ancient Mass Child Sacrifice Ritual" then are all correctly classified by Siebert, which raises an issue that the core sentiment from sub-sentence of headline can be drastically shifted by irrelevant words around. This drawback can be caused due to the complicated mechanism of BERT by the following:

1. Bidirectional feature: Encoder only transformer model like BERT, it considers the context of a word by looking at both the words that come before and after that word, so in this case the sentiment of a word can change based on the words around it. Adding or removing words can change the context of the entire sentence and lead to a sentiment shift.

2. Self-Attention Mechanism: Recall that this feature can weight the importance of each word in relation to other words, so the model dynamically adjust the assigned weight of each word based on how relevant the word is in terms of the whole sentence. Adding or removing words can change the dynamics of these attention weights and lead to a sentiment shift.

3. Imbalanced training data and bias: As a drawback discussed in common word analysis, if the added word appears dominant in either negative or positive contexts in the training data, this might lead to the model overfitting to irrelevant features that shift the sentiment.



```
Italian nun slain by 3 teen girls in Satanic ritual beatified as martyr: positive
Italian nun slain by 3 teen girls in Satanic ritual: negative
Italian nun slain by 3 teen girls in Satanic ritual beatified: positive
Peru Unearths Evidence of Ancient Mass Child Sacrifice Ritual: positive
Evidence of Ancient Mass Child Sacrifice Ritual: negative
Unearths Evidence of Ancient Mass Child Sacrifice Ritual: positive
Reps declare national emergency on ritual killings in Nigeria: positive
Reps declare national emergency on ritual killings: negative
```

Figure 7.6: The sentiment score by Siebert for misclassified example based on 2-grams

7.4 Tool3: Chatgpt for sentiment analysis

Chatgpt as discussed in the related work, it is currently one of the most popular large language models that specifically fine-tuned for conversational tasks. This model is also transformer model but based on GPT(generative pre-trained transformer) architecture, unlike BERT we introduced, GPT is decoder only model mainly designed for understanding and generating human-like text based on the received input. Recall that decoder only model uses autoregression that at each stage generate one word based on the words that come before it in the sequence in forward direction until the response is completed, and its pretrain phase often involves predicting the next word in the sentence.

7.4.1 The methods used for test of chatgpt:

1. Model accessing: OpenAI Python library by openai[36] offers a handy way to access openai API by setting an API key for making a request in python environment. The method 'client.chat.completions.create()' can be called to create a new chat completion, which mainly takes two parameters: 'message' parameter takes the input text as a prompt for response; 'model' parameter specifies the desired model. The response for promopt would be returned by chat completion method. The article[37] mentioned that Chatgpt3 is one of the largest language models developed by OpenAI in 2020 has over 175 billion parameters and has trained around 570GB of datasets, including web pages, books, and other sources. I used Chatgpt4 for sentiment analysis in this project because it is the latest maintained model by OpenAi and trained on more data up through December 2023 with more parameters than Chatgpt3. For each input text, i set their prompt fixed as "Perform binary sentiment analysis of the following text (respond with a single word, positive or negative only): 'text'", the response by chatgpt4 contains either positive or negative only. As OpenAI set API access limit, some code from [38] used to solve the issue of exceeding the API's rate limits.

7.4.2 Results:

1. Evaluation:

The table 7.7 shows the result of evaluation metrics produced by comparing chatgpt classified label against ground truth labels for headline text and tag text. The accuracy for headline text is 100%, which suggests that chatgpt correctly classified all of the headline text. However the accuracy for tag text is about 91%, which indicates an obvious performance gap between classifying headline text and tag text, just like the other introduced tools.

	Metrics	Headline text	Tag text
0	Accuracy	1	0.914815
1	Precision	1	0.914815
2	Recall	1	0.914815
3	F1-score	1	0.914815

Table 7.7: Evaluation result

The confusion matrix7.7 shows the number of misclassified examples by chatgpt for three sentiment classes, though the prompt asks for binary classification. As is known, all the headlines are annotated for binary classification, and there are examples classified as neutral. This suggests there could be a potential issue of input data affecting performance on tag text.

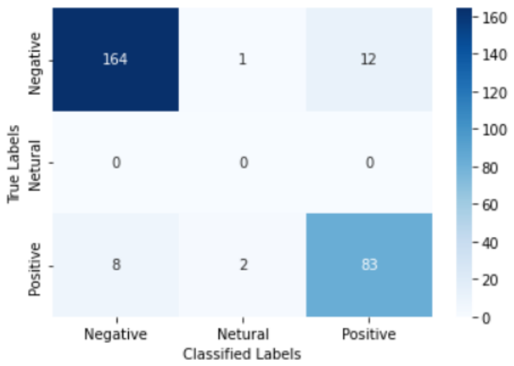
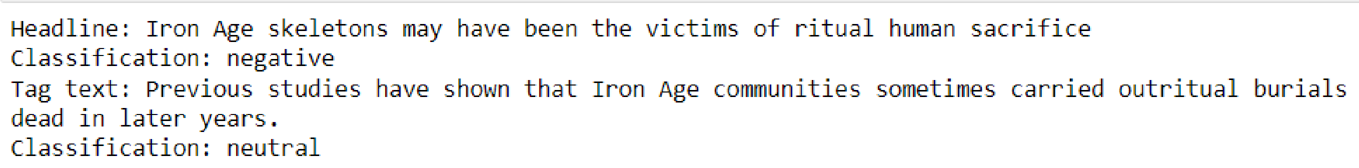


Figure 7.7: Confusion matrix for tag text

As we have seen the obvious performance gap between classifying headline text and tag text, the figure7.8 shows one of the examples where the headline is correctly classified while the accompanied tag text is classified as neutral. By human inspection of this example, we can feel negative sentiment from the headline 'Iron Age skeletons may have been the victims of ritual human sacrifice', its tag text 'Previous studies have shown that Iron Age communities sometimes carried outritual burials in pits, often exhuming the dead in later years.' indeed conveys no sign of either positive or negative sentiment. The nature of tag text is text that is cut from a piece of a whole article in the raw dataset, which can convey a very different sentiment from the headline. As all the annotations i labelled were based on headline text, this can lead to misclassification. Due to this data issue, i will not take tag text into account for further comparative analysis.



```
Headline: Iron Age skeletons may have been the victims of ritual human sacrifice
Classification: negative
Tag text: Previous studies have shown that Iron Age communities sometimes carried outritual burials
dead in later years.
Classification: neutral
```

Figure 7.8: Tag text classified as neutral

7.5 Comparative analysis

In this section, the comparative analysis will be performed across those tools based on evaluation metrics and error analysis. By comparing evaluation metrics across the tools, we can statistically determine which tool performs better on this given dataset. The accuracy and F1 score would be two metrics used for comparison because accuracy provides overall performance and F1 score gives a balanced score against the imbalanced data distribution7.9. Error analysis across tools can also be useful by looking at where tools fail, which provides insight about their advantages and disadvantages if one tool fails on a certain headline and another does not.

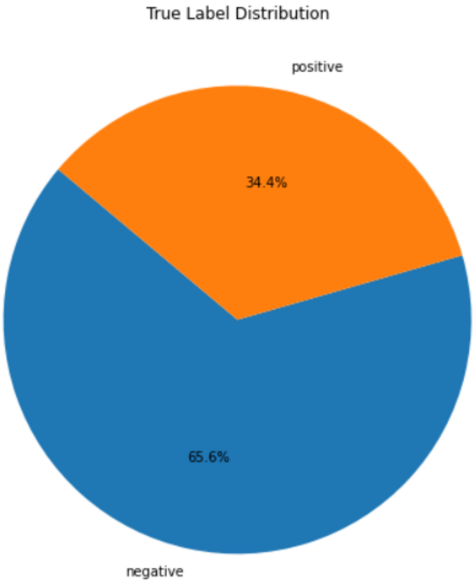


Figure 7.9: True label distribution

The table 7.8 shows the results of evaluation metrics across all introduced sentiment analysis tools. These metrics are calculated by comparing classified labels against ground truth labels.

It appears that all of the sentiment tools perform relatively well, each with scores over %75, Vader has the worst performance compared to other three tools with about 82% accuracy and 78% F1 score, which suggests the lexicon-based approach is the worst performer on this dataset.

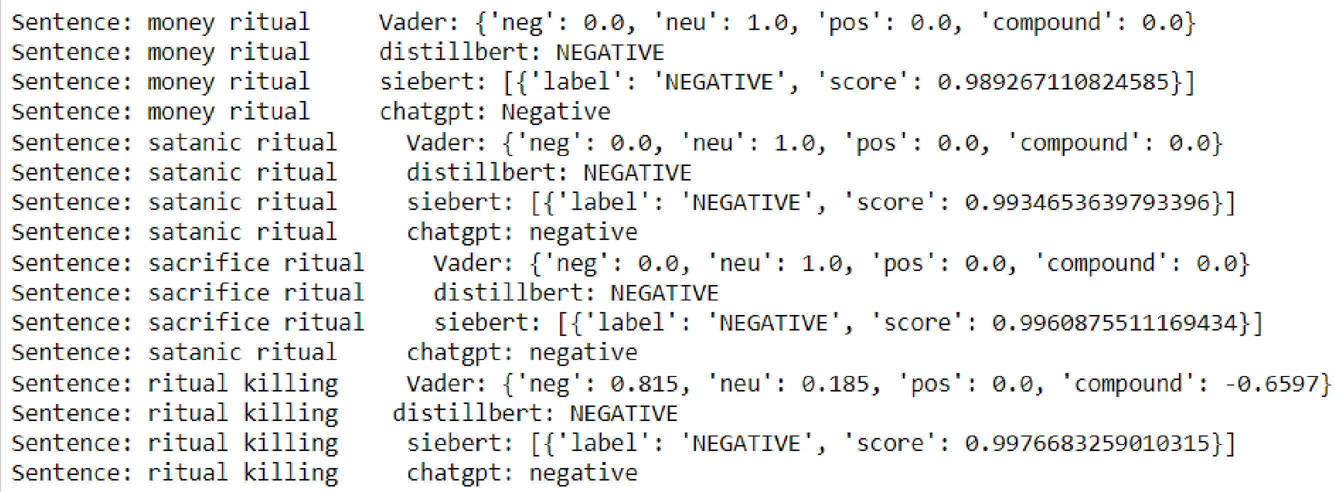
DistillBert(fine-tuned by movie review corpus sst2) and Siebert (fine-tuned by diverse text source) are both encoder only models based on the transformer architecture. It is mainly used for tasks like text classification due to its dynamic vectored representations and "bi-directional" property that can capture better relationships between words. DistillBert ranks second worst compared to other tools with about 91% accuracy and 86% F1 score. Siebert has better performance than DistillBert with about 93% accuracy and 91% F1 score, while DistillBERT has higher precision, which suggests its better capability to identify negative sentiment than DistillBert, and Siebert conversely with higher recall performs better at positive sentiment classification. The performance gap and difference in precision and recall suggest the importance of data quality like diversity and coverage when it comes to the training phase, as it directly affects performance and model tendency focus. The performance of both transformer based models increases compared to Vader, which indicates that deeper learning approach such as transformer based models have better performance compared to lexicon based approaches.

Chatgpt with 100% of evaluation metrics has the best performance across all the tools. Although it is mainly used for text generation tasks as decoder only model, which claims to be worse for text classification tasks, unlike DistillBert and Siebert fine-tuned using a smaller dataset and parameter-adjusted by individual developers, OpenAI is a large and well-funded company and chatgpt4 has over 175 billion parameters and trained more than 570GB of datasets. The training data used for large language models for chatgpt is carefully selected from a broad range of sources to ensure diversity and coverage across topics and contexts. So the quality and quantity of training data are the core factors in terms of performance improvement for transformer models.

	Sentiment analysis tools	Accuracy	Precision	Recall	F1 Score
0	Vader (lexicon-based)	0.822222	0.671756	0.946237	0.785714
1	DistillBert (Fine-tuned by sst2)	0.907407	0.904762	0.817204	0.858757
2	Siebert (Fine-tuned by diverse text sources)	0.933333	0.844037	0.989247	0.910891
3	Chatgpt	1	1	1	1

Table 7.8: Evaluation metrics across the tools

From figure 7.10, it shows the classified sentiment by all the tools for frequent 2-grams that have a clear polarity for human- beings to inspect as negative. Transformer based models all correctly classified those 2-grams, while Vader failed with it. Since Vader as lexicon-based approach uses a fixed polarity score corresponding to each word or phrase in the lexicon to calculate the overall sentiment score, it could be difficult to do sentiment analysis for texts that have a strong cultural background. In this case, if money ritual is not scored as a phrase with negative polarity in the lexicon, breaking down money ritual into two words money and ritual that have a neutral polarity score for sentiment score calculation would lead to a neutral overall sentiment. The last example of 'ritual killing' is correctly classified as negative since killing has a negative polarity score in the lexicon and ritual has a neutral polarity score, which leads to the negative sentiment. Unlike Vader the transformer based models learn from the relationship between tokens in the input text, the models fed by the diverse training data that covers those phrases in the ritual topic can know how to adjust their parameters during inference for correct classification.

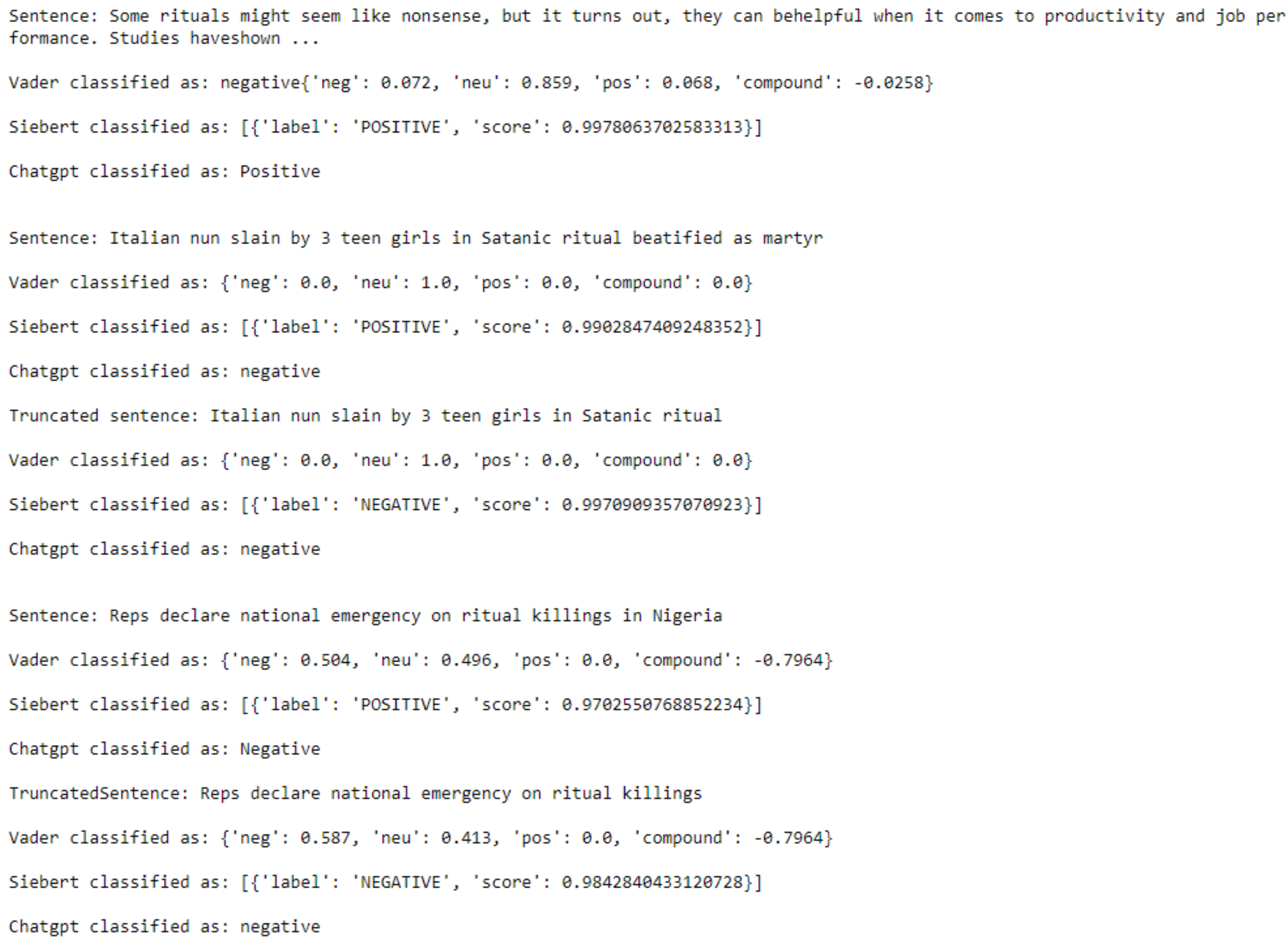


```
Sentence: money ritual     Vader: {'neg': 0.0, 'neu': 1.0, 'pos': 0.0, 'compound': 0.0}
Sentence: money ritual     distillbert: NEGATIVE
Sentence: money ritual     siebert: [{'label': 'NEGATIVE', 'score': 0.989267110824585}]
Sentence: money ritual     chatgpt: Negative
Sentence: satanic ritual     Vader: {'neg': 0.0, 'neu': 1.0, 'pos': 0.0, 'compound': 0.0}
Sentence: satanic ritual     distillbert: NEGATIVE
Sentence: satanic ritual     siebert: [{'label': 'NEGATIVE', 'score': 0.9934653639793396}]
Sentence: satanic ritual     chatgpt: negative
Sentence: sacrifice ritual     Vader: {'neg': 0.0, 'neu': 1.0, 'pos': 0.0, 'compound': 0.0}
Sentence: sacrifice ritual     distillbert: NEGATIVE
Sentence: sacrifice ritual     siebert: [{'label': 'NEGATIVE', 'score': 0.9960875511169434}]
Sentence: satanic ritual     chatgpt: negative
Sentence: ritual killing     Vader: {'neg': 0.815, 'neu': 0.185, 'pos': 0.0, 'compound': -0.6597}
Sentence: ritual killing    distillbert: NEGATIVE
Sentence: ritual killing     siebert: [{'label': 'NEGATIVE', 'score': 0.9976683259010315}]
Sentence: ritual killing     chatgpt: negative
```

Figure 7.10: The sentiment by tools for frequent 2-grams

The figure 7.11 shows the classified sentiment by all the tools for misclassified sentence. The first sentence 'Some rituals might seem like nonsense, but it turns out, they can behelpful when it comes to productivity and job performance. Studies haveshown ...' uses a contrastive structure through the phrase "but it turns out" to contrast the initial bad perception (in this case, as "nonsense") with an argument that suggests a good perspective (here that such rituals "can be helpful when it comes to productivity and job performance"). Vader failed to classify the first sentence correctly even with its conjunction rule applied ("but it turns out"), which indicates lexicon-based approach's weakness that since it can not capture the context and coherence of the sentence, the calculation of the inappropriate words polarity score ("nonsense" and "behelpful") with the rule applied still lead to a misinterpretation of the overall sentiment in the mixed sentiment text. The transformer based models correctly classified the first sentence; unlike Vader they learn from massive amounts of training data. The self-attention mechanism dynamically weights the importance of each word in a sentence based on the surrounding words, and contextual embedding contains the numerical representation of a word that can change based on its surrounding words. These models rely on self-attention mechanism to generate a weighted combination of all word embeddings from the input token, so they can learn from contextual embeddings during their training phase. Their mechanism of using contextual embedding and self-attention mechanism supports capturing the context based on words dependent on each other in the text for better handling natural language processing tasks.

The classified sentiment for the second sentence and its truncated sentence suggest the drawbacks of tools. The incorrect 100 percent of the neutral sentiment classified by Vader for both sentences is because it relies on the polarity score of each word in the lexicon for sentiment score calculation, so those missing words and their corresponding polarity in the lexicon can lead to a completely default neutral sentiment. Siebert correctly classified a truncated sentence that removed words "beatified as martyr" without affecting the core content, but it failed on the whole sentence, which indicates that those models can still fail to capture the main concept even with what they called the contextual understanding mechanism to learn from data that adjusts the assigned weight of each word based on how relevant the word is in terms of the whole sentence, since the core sentiment of the sentence can be shifted by adding or removing words around. The classified sentiment for the final sentence and its truncated sentence also show that the classified sentiment result by Siebert shifted by irrelevant words that stand for country "in Nigeria", which suggests the potential issue of imbalanced training data and bias discussed in common word analysis from 7.3 that if the added word "Nigeria" appears dominant in positive contexts in the training data, this might lead to the model overfitting to this country word "Nigeria" that shifts the sentiment, and the bias issue is mentioned in that the model classifies "poland" as positive while "Nigeria" as negative. Vader has no such problem since it relies on polarity score of each word in the lexicon for sentiment score calculation. Chatgpt, the same as Bert based on transformer has correctly classified all the sentiment. This tool has unidirectional attention that learns from previously seen text and predicts the next word in a sentence in one direction. It is supposed to be worse for sentiment classification, but it has better performance than Bert with bi-directional attention that learns full context by discovering the relationship between words in both directions, which again emphasises the importance of quality and quantity for training data.



```
Sentence: Some rituals might seem like nonsense, but it turns out, they can behelpful when it comes to productivity and job per
formance. Studies haveshown ...

Vader classified as: negative{'neg': 0.072, 'neu': 0.859, 'pos': 0.068, 'compound': -0.0258}

Siebert classified as: [{'label': 'POSITIVE', 'score': 0.9978063702583313}]

Chatgpt classified as: Positive


Sentence: Italian nun slain by 3 teen girls in Satanic ritual beatified as martyr

Vader classified as: {'neg': 0.0, 'neu': 1.0, 'pos': 0.0, 'compound': 0.0}

Siebert classified as: [{'label': 'POSITIVE', 'score': 0.9902847409248352}]

Chatgpt classified as: negative

Truncated sentence: Italian nun slain by 3 teen girls in Satanic ritual

Vader classified as: {'neg': 0.0, 'neu': 1.0, 'pos': 0.0, 'compound': 0.0}

Siebert classified as: [{'label': 'NEGATIVE', 'score': 0.9970909357070923}]

Chatgpt classified as: negative


Sentence: Reps declare national emergency on ritual killings in Nigeria

Vader classified as: {'neg': 0.504, 'neu': 0.496, 'pos': 0.0, 'compound': -0.7964}

Siebert classified as: [{'label': 'POSITIVE', 'score': 0.9702550768852234}]

Chatgpt classified as: Negative

TruncatedSentence: Reps declare national emergency on ritual killings

Vader classified as: {'neg': 0.587, 'neu': 0.413, 'pos': 0.0, 'compound': -0.7964}

Siebert classified as: [{'label': 'NEGATIVE', 'score': 0.9842840433120728}]

Chatgpt classified as: negative
```

Figure 7.11: The sentiment by tools for misclassified sentences

7.6 Conclusion

The lexicon-based approach using a fixed polarity score corresponding to each word or phrase in the lexicon to calculate the overall sentiment score is not ideal. The main limitations of this approach as following:

1. Missing words or incorrect word polarity in Lexicon: In my findings, i noticed phrases like "satanic ritual", "money ritual" and "moon ritual" are all classified as neutral by Vader. This can be caused by word polarity defined as neutral incorrectly, or missing words defaulted as netural in the lexicon. So when the crucial words or phrases in the sentence are missing in the lexicon, those missing words or phrases will not be considered at all for the calculation of the overall sentiment score, which can lead to incorrect sentiment classification for the whole sentence.

2. Fixed Sentiment Scores: The sentiment polarity scores assigned to words in the lexicon are fixed, so words with a fixed polarity score can lead to incorrect classification as the sentiment of words can be different depending on context. In my finding, the sentence "satanic ritual

put alive people in fire" was classified as positive, just because the word "alive" with a higher positive score offset "fire"'s lower negative score when "satanic ritual" is not considered.

3. Incapability of understanding coherence and context: In my finding, Vader failed to classify the contrastive sentence that mixed different sentiment 'Some rituals might seem like nonsense, but it turns out, they can behelpful when it comes to productivity and job performance. Studies haveshown ...' correctly even with its conjunction rule applied ("but it turns out") to balance the score between "nonsense" and "behelpful", which indicates lexicon-based approach can not capture the context and coherence within the sentence.

Even though Vader has those limitations because of its mechanism, it can still be used for sentiment analysis for specific texts like feedback and reviews because such texts is usually straightforward for expressing emotion as they are composed of strong polarity words.

The transformer based approaches using pretrained model to optimize inner parameters and further train the pre-trained model for sentiment analysis have better performance than Vader, in which Chatgpt correctly classified all headlines with 100% accuracy followed by Siebert and DistillBert with over 90% accuracy. Unlike Vader, the transformer models all correctly classified those cultural phrases like "money ritual", "satanic ritual" and "sacrifice ritual", It is because the transformer based models learn from the relationship between tokens in the input text, the models fed by the diverse training data that covers those phrases in the ritual topic can know how to adjust their parameters during inference for the correct classification. One feature of transformer models that I noticed differs from Vader's fixed score in the lexicon is its self-attention mechanism, which dynamically weights the importance of each word based on the surrounding words to generate a weighted combination of all word embeddings, which allows the model to better capture context from text. However, it has limitations:

1. Data quality: The transformer models, whether Bert(encoder-only) or chatgpt(decoder-only) heavily rely on the training data, so the quality and quantity of training data are the core factors for the model to improve performance. The performance difference in order of chatgpt4 -> Siebert -> DistillBert has proven it, in which Siebert fine-tuned by a dataset of diverse text sources performs better than DistillBert by a movie dataset, while Chatgpt4 constantly maintained and developed by a well-funded company has the best performance because of its more diverse dataset with better quality and more parameters inside of models to learn from training data.

2. Data biases and overfitting by data imbalanced: If the data used for pretraining or fine-tuning is mainly from certain cultures or bias, then the model can learn bias and may not perform well on data from different cultural background. In my finding, "nigeria" and "poland" both stand for country, however "nigeria" is classified as negative while "poland" is classified as positive. This can be caused if the training data used involving the african country "nigeria" has been dominantly mentioned in negative contexts while the training data involving the European country "poland" is dominantly mentioned in positive or neutral contexts.

3. Incorrect sentiment shift by its attention mechanism: The sentiment of the sentence can be shifted by adding or removing irrelevant words around, the sentence "Italian nun slain by 3 teen girls in Satanic ritual" is correctly classified, however the sentiment shift by adding "beatified as martyr" at the end without affecting the core sentiment, which indicates that those models can still fail to capture the main concept even with what they called the contextual understanding mechanism to learn from data that adjusts the assigned weight of each word based on how relevant the word is in terms of the whole sentence.

4. Length Limitation: Most transformer models have a maximum input length like Bert with up to 512 tokens due to limited capability, and chatgpt4 has a longer input length; truncating the input sequences can help with that but could result in the loss of information since human language is context-dependent.

From here, we can draw some general conclusions from this project: The transformer based model benefiting from attention mechanisms is better than the lexicon-based approach like Vader for sentiment analysis. If the quality and quantity of training data are guaranteed as well as more parameters, the decoder-only model like Chatgpt4 can perform better than encoder-only models like Bert which claimed to be better at classification tasks. Although the performance of Chatgpt4 is remarkably fantastic, rather than sentiment analysis chatgpt was initially designed for understanding and generating human-like text by predicting the next word based on the previous words. So, can current sentiment analysis tools like Bert or even text generation model like chatgpt4 reach human judgmental levels in terms of sentiment analysis? The answer is no. Rather than comprehension like human beings, their understanding is based on patterns or relationships learned from data, which adjust the parameters, and the model infers the output from the adjusted parameters for the natural language processing task. So even with the more advanced mechanism of so-called self-attention, the model can still struggle to capture the main clause (holding the core sentiment) from text and fails to classify sentiment (sentiment shifted by irrelevant clauses while humans don't) due to the complexity of human language where the sentiment is subjective and can be impacted by context and the coherence of the text. For instance, human beings from different cultural backgrounds can judge a statement differently if it mixes both positive and negative sentiments.

It's worth noting that current research and development like attention mechanism or increasing more parameters in the tools for better handling natural language processing are aimed at addressing the challenges of natural language processing, which makes these tools like chatgpt4 more sophisticated and closer to the level of human-like understanding over time. However, reaching human judgmental levels still can be a significant challenge for current tools in sentiment analysis.

Acknowledgements

For my paper, i would like to acknowledge Fred Cummins for helping me with this research project.

Bibliography

1. Barney, N. What is sentiment analysis (opinion mining)? Definition from SearchBusinessAnalytics. https://www.techtarget.com/searchbusinessanalytics/definition/opinion-mining-sentiment-mining (2023).
2. Mäntylä, M. V., Graziotin, D. & Kuutila, M. The evolution of sentiment analysis—A review of research topics, venues, and top cited papers. *Computer Science Review* **27,** 16–32. ISSN: 1574-0137. https://www.sciencedirect.com/science/article/pii/S1574013717300606 (2018).
3. AWS. What is Sentiment Analysis? - Sentiment Analysis Explained - AWS. https://aws.amazon.com/what-is/sentiment-analysis/ (2023).
4. Ziani, A. *et al. Recommender System Through Sentiment Analysis* 2017. https://hal.science/hal-01683511.
5. Hartman, S. Enhancing Customer Sentiment Analysis. https://www.uniphore.com/blog/how-sentiment-analysis-can-improve-customer-experience/ (2023).
6. Edwards, K. How to Do Social Media Sentiment Analysis in Politics. https://www.determ.com/blog/social-media-sentiment-analysis-in-politics/ (2021).
7. Roldós, I. Major Challenges of Natural Language Processing (NLP). https://monkeylearn.com/blog/natural-language-processing-challenges/ (2020).
8. Taj, S., Shaikh, B. B. & Fatemah Meghji, A. *Sentiment Analysis of News Articles: A Lexicon based Approach* in *2019 2nd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)* (2019), 1–5. https://ieeexplore.ieee.org/document/8673428/authors#authors.
9. Wankhade, M., Annavarapu, Chandra Sekhara Rao & Kulkarni, C. A survey on sentiment analysis methods, applications, and challenges. *Artificial Intelligence Review* **55,** 5731–5780. ISSN: 1573-7462. https://doi.org/10.1007/s10462-022-10144-1 (2022).
10. Vivekanandan, M. Feature Extraction - Natural Language Processing. https://www.linkedin.com/pulse/feature-extraction-natural-language-processing-vivekanandan/ (2023).
11. Alzyout, M., AL Bashabsheh, E., Najadat, H. & Alaiad, A. *Sentiment Analysis of Arabic Tweets about Violence Against Women using Machine Learning* in *2021 12th International Conference on Information and Communication Systems (ICICS)* (2021), 171–176. https://ieeexplore.ieee.org/abstract/document/9464600.
12. Gupta, N. & Agrawal, R. *Chapter 1 - Application and techniques of opinion mining* in *Hybrid Computational Intelligence* (eds Bhattacharyya, S., Snášel, V., Gupta, D. & Khanna, A.) (Academic Press, 2020), 1–23. ISBN: 978-0-12-818699-2. https://www.sciencedirect.com/science/article/pii/B9780128186992000019.
13. Kian, Long Tan, Chin, Poo Lee & Kian, Ming Lim. A Survey of Sentiment Analysis: Approaches, Datasets, and Future Research. https://www.mdpi.com/2076-3417/13/7/4550 (2023).
14. Lee, K. C. Sentiment Analysis — Comparing 3 Common Approaches: Naive Bayes, LSTM, and VADER. https://towardsdatascience.com/sentiment-analysis-comparing-3-common-approaches-naive-bayes-lstm-and-vader-ab561f834f89 (2021).
15. Vaswani, A. *et al. Attention Is All You Need* 2023. arXiv: 1706.03762 [cs.CL]. https://arxiv.org/abs/1706.03762v7.

16. Dhola, K. & Saradva, M. *A Comparative Evaluation of Traditional Machine Learning and Deep Learning Classification Techniques for Sentiment Analysis* in *2021 11th International Conference on Cloud Computing, Data Science Engineering (Confluence)* (2021), 932–936. https://ieeexplore.ieee.org/abstract/document/9377070.

17. Hutto C.J. Gilbert, E. VADER: A Parsimonious Rule-Based Model for Sentiment Analysis of Social Media Text. *Proceedings of the International AAAI Conference on Web and Social Media* **8,** 216–225. https://ojs.aaai.org/index.php/ICWSM/article/view/14550 (May 2014).

18. Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv: 1810.04805 [cs.CL]. https://arxiv.org/abs/1810.04805 (2019).

19. Liu, Y. *et al.* RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv: 1907.11692 [cs.CL]. https://arxiv.org/abs/1907.11692 (2019).

20. OpenAI. Introducing ChatGPT. https://openai.com/blog/chatgpt (2022).

21. Bird, S., Klein, E. & Loper, E. *Natural language processing with Python: analyzing text with the natural language toolkit* https://www.nltk.org/ (" O'Reilly Media, Inc.", 2009).

22. Chollet, F. *et al. Keras* https://github.com/fchollet/keras.

23. Yang, Z. *et al.* XLNet: Generalized Autoregressive Pretraining for Language Understanding. arXiv: 1906.08237 [cs.CL]. https://arxiv.org/abs/1906.08237 (2020).

24. Shan Lee, V. L., Gan, K. H., Tan, T. P. & Abdullah, R. Semi-supervised Learning for Sentiment Classification using Small Number of Labeled Data. *Procedia Computer Science* **161.** The Fifth Information Systems International Conference, 23-24 July 2019, Surabaya, Indonesia, 577–584. ISSN: 1877-0509. https://www.sciencedirect.com/science/article/pii/S1877050919318708 (2019).

25. Agrawal, S. K. Metrics to evaluate your classification model to take the right decisions, Analytics Vidhya. https://www.analyticsvidhya.com/blog/2021/07/metrics-to-evaluate-your-classification-model-to-take-the-right-decisions/ (2024).

26. Hutto, C. Welcome to VaderSentiment's documentation! - VaderSentiment 3.3.1 documentation. https://vadersentiment.readthedocs.io/en/latest/index.html (2022).

27. Hutto C.J. Gilbert, E. VADER-Sentiment-Analysis. https://github.com/cjhutto/vaderSentiment (2014).

28. IBM. What is a Transformer Model? https://www.ibm.com/topics/transformer-model#:~:text=Transformer%20models%20work%20by%20processing,mechanisms%20and%20feedforward%20neural%20networks. (Unknow).

29. Face, H. *The Hugging Face Course, 2022* https://huggingface.co/course. 2022.

30. Sanh, V., Debut, L., Chaumond, J. & Wolf, T. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv: 1910.01108 [cs.CL]. https://arxiv.org/abs/1910.01108 (2020).

31. Jochen, Hartmann, M., Heitmann, C., Siebert, C. & Schamp. SiEBERT - English-Language Sentiment Classification. https://huggingface.co/siebert/sentiment-roberta-large-english (2023).

32. Hartmann, J., Heitmann, M., Siebert, C. & Schamp, C. More than a Feeling: Accuracy and Application of Sentiment Analysis. *International Journal of Research in Marketing* **40,** 75–87. https://www.sciencedirect.com/science/article/pii/S0167811622000477 (2023).

33. Face, H. DistilBERT base uncased finetuned SST-2. https://huggingface.co/distilbert/distilbert-base-uncased-finetuned-sst-2-english#training.

34. Face, H. Dataset card for the Stanford Sentiment Treebank. https://huggingface.co/datasets/stanfordnlp/sst2.

35. Socher, R. *et al. Recursive Deep Models for Semantic Compositionality Over Sentiment Treebank* in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing* (Association for Computational Linguistics, 2013), 1631–1642. https://www.aclweb.org/anthology/D13-1170.

36. openai. OpenAI Python API library. https://github.com/openai/openai-python (2023).

37. Lammertyn, M. 60+ ChatGPT Statistics And Facts You Need to Know in 2024. https://blog.invgate.com/chatgpt-statistics (2024).

38. Openai. openai cookbook. https://github.com/openai/openai-cookbook/blob/main/examples/How_to_handle_rate_limits.ipynb.