**Big Data Analytics over Social Media Data**

Team 3: Benny Lim Yi Jie, Chen JiaJun, Kye Jordan O`Reilly, Lai Wen Jun, Lee Yan Rong, Sophie Wong Shu Ping

Information and Communications Technology

Singapore Institute of Technology 172 Ang Mo Kio Ave 8, Singapore 567739

[2101955@sit.singaporetech.edu.sg](mailto:2101955@sit.singaporetech.edu.sg), [2101351@sit.singaporetech.edu.sg](mailto:2101351@sit.singaporetech.edu.sg), [2102841@sit.singaporetech.edu.sg](mailto:2102841@sit.singaporetech.edu.sg), [2102989@sit.singaporetech.edu.sg](mailto:2102989@sit.singaporetech.edu.sg), [2102608@sit.singaporetech.edu.sg](mailto:2102608@sit.singaporetech.edu.sg), [2102284@sit.singporetech.edu.sg](mailto:2102284@sit.singporetech.edu.sg)

***Abstract*—This paper investigates the application of Big Data Analytics to analyse social media data, focusing on sentiments expressed in employee reviews for the company Accenture. The significance of this study lies in leveraging the potential of Hadoop technologies to efficiently process and analyse large-scale social media data, enabling SIT students to understand employee satisfaction and experiences better. The sentiment analysis utilises Hadoop distributed data processing framework to classify employee reviews into positive, negative, and neutral categories. This process includes data collection, preprocessing, sentiment analysis, and visualisation. The experimental evaluation was conducted on a large dataset of Glassdoor reviews. The results demonstrate the effectiveness and scalability of the proposed approach in extracting meaningful insights from social media data. The findings can assist SIT students in understanding employee sentiments, identifying areas for improvement, and enhancing overall employee satisfaction and workplace experience in the company Accenture for their Work-Study Degree.**

***Keywords— Hadoop, sentiment analysis, reducer, mapper, distributed systems, satisfaction***

# INTRODUCTION

Social media platforms such as Glassdoor and Linkedin are designed to provide people, professionals, and businesses to interact, share information online and even find job opportunities. These platforms allow users to share their reviews and experience as current or former employees. With such an abundance of information contained in these platforms, analysis can be done to gather people's thoughts, concerns, and emotions. This project collects user reviews and analyses the company's branding and attitudes. The analysis result will help us to have a better understanding and quick view of the company's employer brand personality and employee satisfaction levels.

In this paper, analysis has been done to analyse the opinions and experience of both former and current employees towards Accenture. The research aims to reveal the employees' sentiments on working at Accenture. We seek to answer questions like: What is the feeling towards the company over the years? What is the frequent sentiment word the employees use to describe the company? How do sentiment values vary for different job titles within Accenture? In this project, data crawling and processing have been done on Accenture's Glassdoor website for sentiment analysis. Our primary analysis method is lexicon-based sentiment analysis, which has advantages and limitations that will be discussed later in the report. The analysis will be shown in a dashboard view built using python flask.

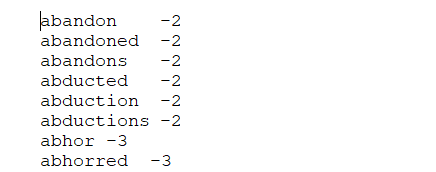
# RELATED WORKS

1. *Lexicon-based model*

Lexicon-based model is a sentiment analysis approach to determine and analyse the semantic and syntactic pattern [1]. Words are labelled as positive, negative, or neutral when referred to a dictionary. This dictionary will provide a list of words with their score based on the emotional valence; thus, it is called a valence dictionary. An example of how sentiment analysis is made is as follows: Providing a sentence, "The content of this book is super bad." With the help of the valence dictionary, the word "super" would be labelled as positive with a value of +3; the word "bad" is negative with a possible value of -3. The other words in the sentence may be neutral. After labelling the words' scores, the sentiment score will be calculated by combining these values mathematically. In this example, the sentiment value for this sentence will be 3-3=0.

1. *AFINN*

AFINN lexicon is a dictionary of terms that has been rated for valence between -5 and +5 by Finn Årup Nielsen. An example of the AFINN dictionary used is shown below.



*Fig 1: Part of the valence dictionary*

# THE PROPOSED APPROACH

This section will cover the detailed approach and implementation of this project. Our team uses Python beautiful soup for data-scraping, Hadoop and MapReduce for pre-processing and analysing, and Python Flask for the web application to display our results in many forms, such as bar plots and word clouds.

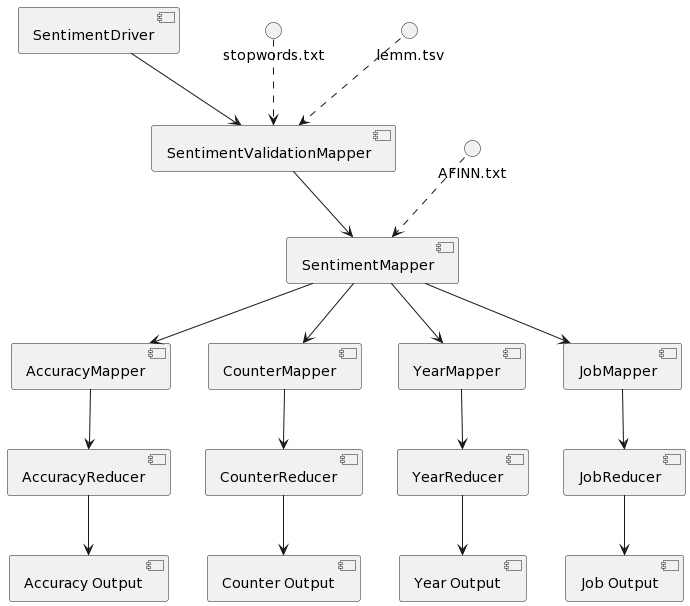
1. *Overview of approach*

The approach to this project is to develop a lexicon based model with the valence dictionary as AFINN. Lexicon based models were selected as lexicon-based model sentiment analysis methods are less expensive because they do not require implementing advanced sentiment analysis algorithms.The sentiment value can be easily calculated by referring to a valence dictionary, and there is no need for data training.

In this project, AFINN-111 has been used as the valence dictionary and insignificant stop words have also been removed, an overview of the implementation of the model to analyse the sentiment is as follows

1. Data validation and cleaning
2. Sentiment analysis of pros and cons
3. Other Analysis such as sentiment by word count, sentiment filter by jobs and years

The diagram below shows a high-level implementation of the MapReduce program. The Driver would call the SentimentValidationMapper to clean the data. The SentimentMapper would do a joined function with the sentimental values, then parse onto three different mappers for three different analyses. Notably, the Stopword.txt and Lemm.txt are being cached and used by SentimentValidationMapper, while the Afiin.txt file is by SentimentMapper. More details will be given in the later part of the report.

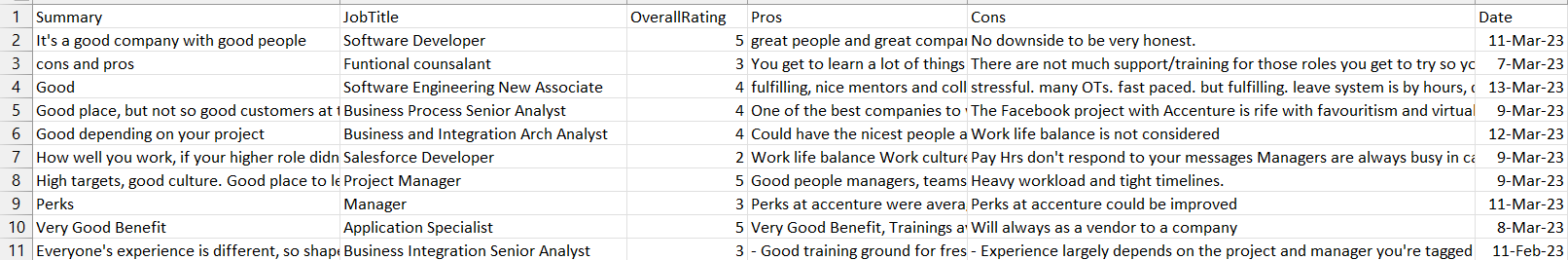


*Fig 2: Overall MapReduce Approach*

1. *Data Collection*

The data collection is done using Python’s API, beautiful soup, to scrape and append data into a CSV file. The total data collected was 116k reviews and the earliest review for Accenture is 2013.

The analysis result is aimed at knowing the review given across multiple job types in Accenture and the employee's difference in review for each year. With that in mind, the dataset we collected includes six columns: Summary, JobTitle, OverallRating, Pros, Cons, and Date. Figure 3 provides a better reference for the dataset. It is important to note that these data may have inherent biases since they cannot capture the nuance or whole experience of the reviewer. For example, people on Glassdoor may be more likely to complain, and sometimes a person's vocabulary may not be expressive enough to convey their feelings.



*Fig 3: Part of dataset collected*

1. *Data Pre-processing*

Pre-processing before analysis is essential as cleaning the data up to avoid computing useless datasets and reducing inaccuracies. Furthermore, pre-processing includes transforming the data into a usable format for analysis. Processing a large amount of data can be time-consuming. Employing Hadoop allows us to store the data in HDFS and then process and analyse it using MapReduce programs, resolving the issue by distributing the computing needs to the distributed system.

SentimentValidationMapper.java is the project's data cleaning mapper and is the first map that is chained in the entire process. The four steps of the pre-processing:

1. Normalising the Data

Each row of data is normalised to lowercase and have whitespace removed to make it easier to analyse individual words.

1. Removing stop words

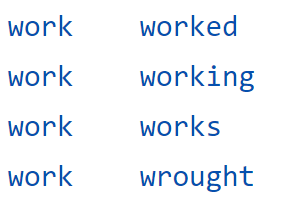
Next, we did Text pre-processing by cleaning up the stop words such as "thus," "that," and "from." We utilised a distributed cache file to add a stopwords.txt file containing a list of commonly used stop words (Source: <insert Source>). A mapper was deployed to clean the data by comparing each word in the input to the stop word list and removing any matches.

1. Data cleaning

To clean the data, we first removed all non-alphanumeric-whitespace characters as we are primarily interested in working with English words and numbers. Removing these characters also helps to prevent any corrupted information from being included in our analysis.

1. Lemmatization

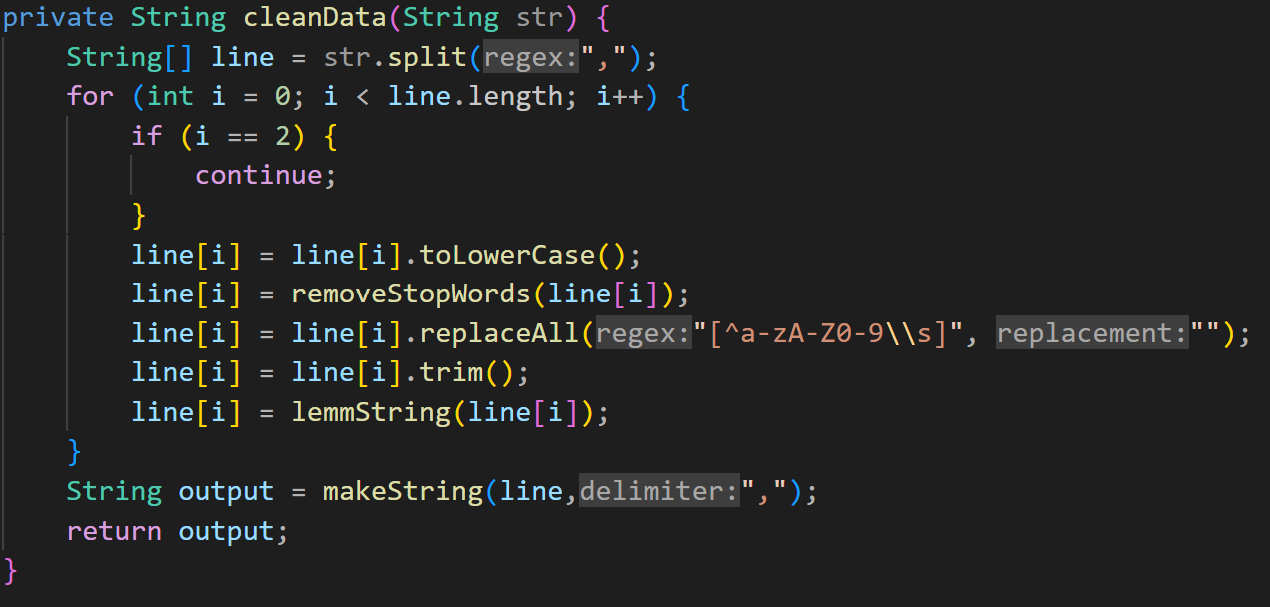
Lemmatization is the process of grouping together different inflected forms of the same word. Such as converting “working”, “worked” to just work. An example can be seen below. The left side is the original word and the right is a variance of the word.



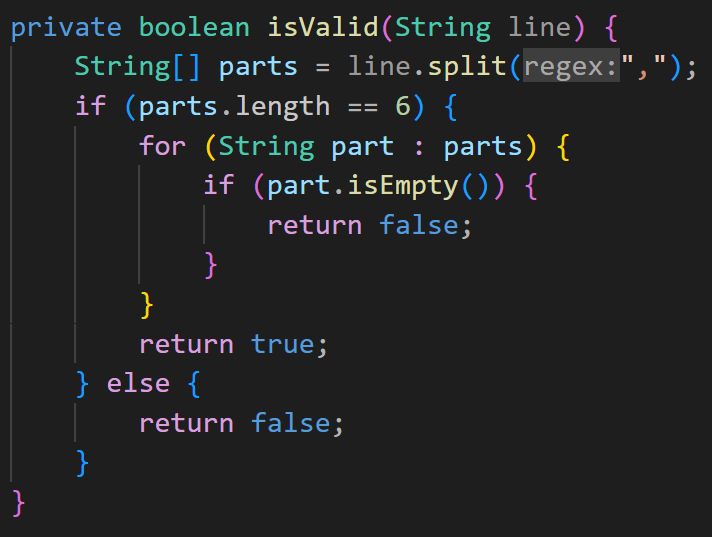
*Fig 4: Example of lemmatization*

1. Verifying Data Integrity

Finally, we checked if the integrity of having six parts still holds up, and if it does, the data would then be sent for further analysis. Else, the row will be discarded.



*Fig 5: Data Processing Code*



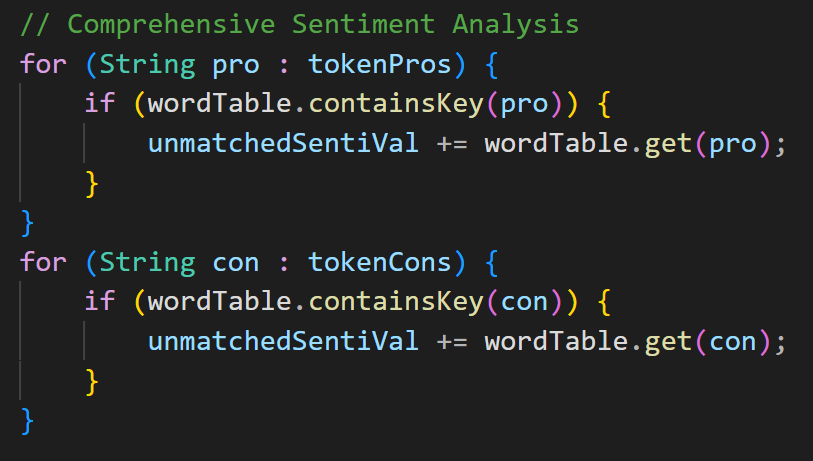
*Fig 6: Data Verifying Code*

1. *Sentimental Analysis*

Sentiment analysis is done following the lexicon based approach. The data is labelled using a mapper which is implemented in SentimentMapper.java. Each word is tokenized in the mapper, after which it will analyse only the pros and cons of each review. The team used a simplistic approach to calculate the sentiment. The formula used is , where the value is a summation of the value attached to the word.

For further analysis and comparison, the team has also implemented two categories of sentiment analysis, the Comprehensive Sentiment Analysis and Simplistic Sentiment Analysis. The former would give a more detailed analysis whilst the latter would give a high level analysis.

The Comprehensive Sentiment Analysis, the “UNMATCH” category would give a more nuanced take as it accounts for more factors like the negative sentiments in the pros, for example, “this company is not bad” the sentiment result will be negative values, but the actual meaning of it is neutral. This category should be use for a more detail view on things or when more subtle changes needed to be analysed.



*Fig 7: UNMATCH code*

The second category,the “MATCH” category, would take only positive valence value for pros and negative valence value for cons. In contrast to the first category, this would give a more simplistic view of the analysis of Accenture, which focuses on the positive valence value of the pros and the negative valence value of the cons. The team will compare the two categories for each analysis to show the information from different perspectives. This category should be use for a more simplistic view on the pros and cons like finding the main negative and positive driver of Accenture.



*Fig 8: MATCH code*

1. *Data analysis*

There are three analyses done, WordCount, Year, and Job. The technique that this project used to do the sentiment analysis is the mapper chain. The AFIINN.txt has been cached into the Hadoop distributed cache for checking. The contents are mapped to a hash table which would be used to check the sentimental values.

Also for this project, the reduction stage isn't very computationally intensive. Too many reducers may be unproductive, and the file size needs to be more extensive to warrant more reducers. If the file size increases or the algorithms become more complex, which requires more computational power, more reducers may be needed to distribute the reduce function effectively.

The team has implemented three different jobs for each analysis. Each job will go through both SentimentValidationMapper.java and SentimentMapper.java.

To perform sentiment analysis on the cleaned dataset, SentimentMapper.java will join the MATCHED and UNMATCHED sentimental values. Map-side join was chosen over reduce-side join due to its superior performance, as it skips the reduce-stage computation and shuffle phase required by reduce-side join. However, map-side join requires the other dataset to be small enough to fit into the cache. The combined size of the three .txt files used in this project is less than one megabyte, which is considered small. If a larger dataset is used, reduced-side-join should be considered instead

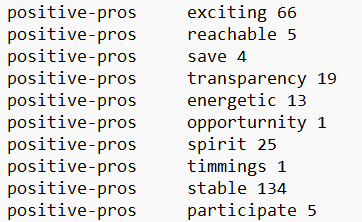
After having the valence value for the data collected, the data will be further analysed by either going through the word count mapper, year mapper or job mapper. The details and implementation for each analysis will be documented and explained as follows.

1. WordCount Analysis:

After going through the sentiment mapper, the data can be further processed for the word count analysis by going through the CounterMapper.java and CounterReducer.java. The word count mapper will map each row of the dataset and check what positive sentiment value word is used in the pro column, what negative sentiment value word is used in the cons column, and what neutral sentiment value word is used.

After mapping, the word count reducer will combine all the words with the neutral, positive or negative tag. There are six different keys for the reducer in the form of sentiment-[pros/cons]

The number of times the word appears is calculated. The result of the analysis will be a list of the type of words, the word, and the number it appears in the dataset, as shown in 9.

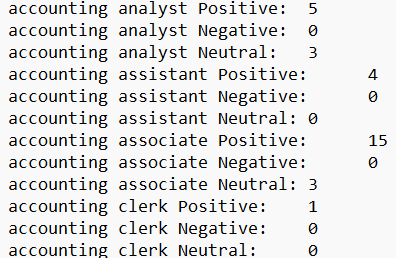


*Fig 9: Result of Word Count Analysis*

1. Job Analysis:

The job analysis aims to get how many positive, negative or neutral sentiments for every job type in Accenture.

The results are sorted by <job title> and the number of positive, negative and neutral words in the review. In order to use the "shuffling" function of the MapReduce, the job title was used as the "key" output value so that the data would be sorted without needing an extra program to sort the results.

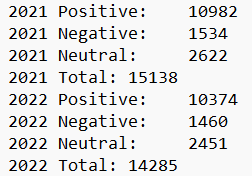


*Fig 10: Result of Job Analysis*

1. Year Analysis:

The year analysis aims to show employees' yearly sentimental value towards Accenture from 2013 until 2023.

Similarly, the data date is used as the key in the mapper, so all the data with the same Date will go to the same reducer. The reducer's output Key is <Year><Sentiment (Positive, negative, neutral, total>. The results are automatically sorted by Date and sentimental values by thoughtfully assigning the Key value.



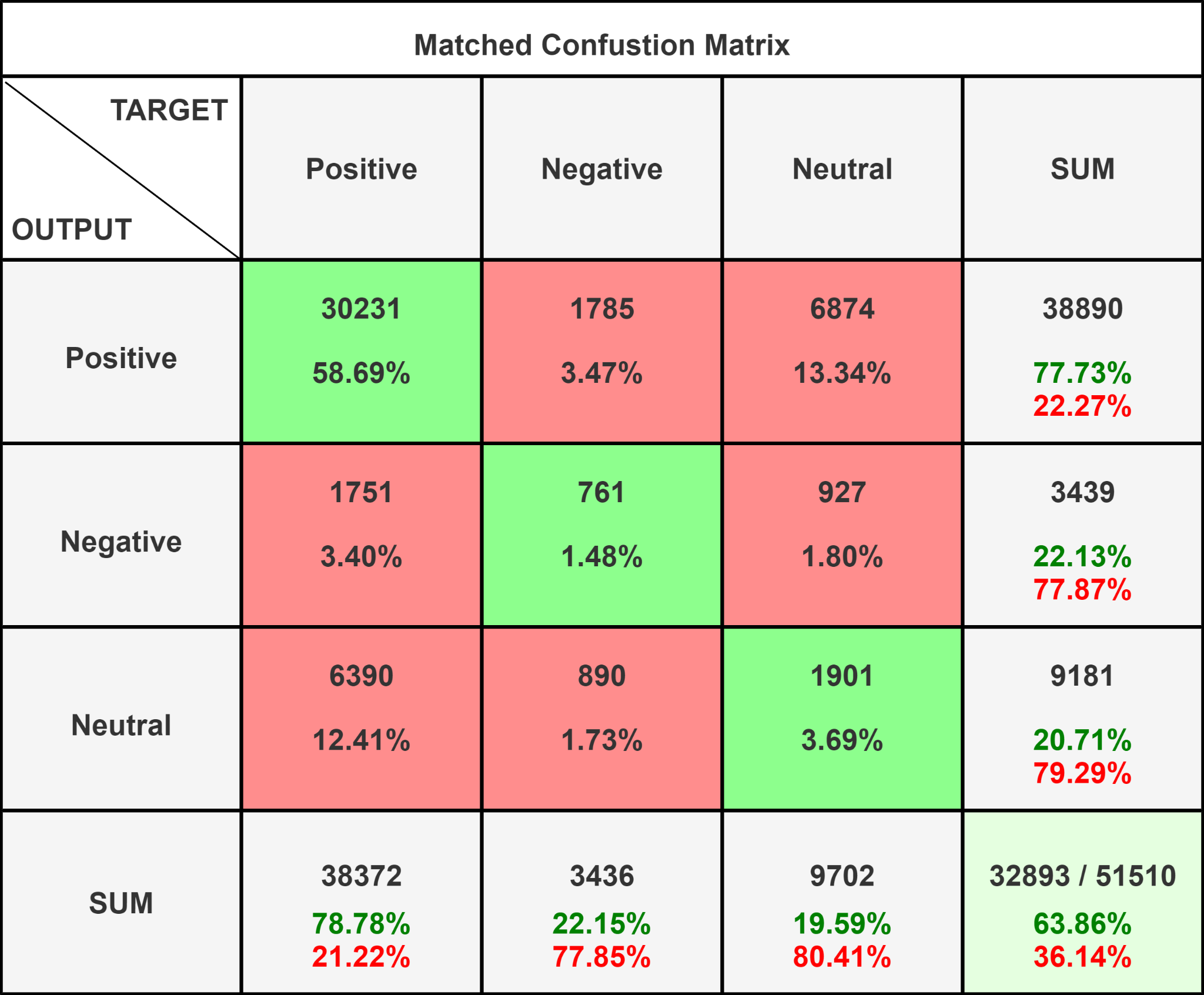
*Fig 11: Result of Year Analysis*

*F. Result and Insight*

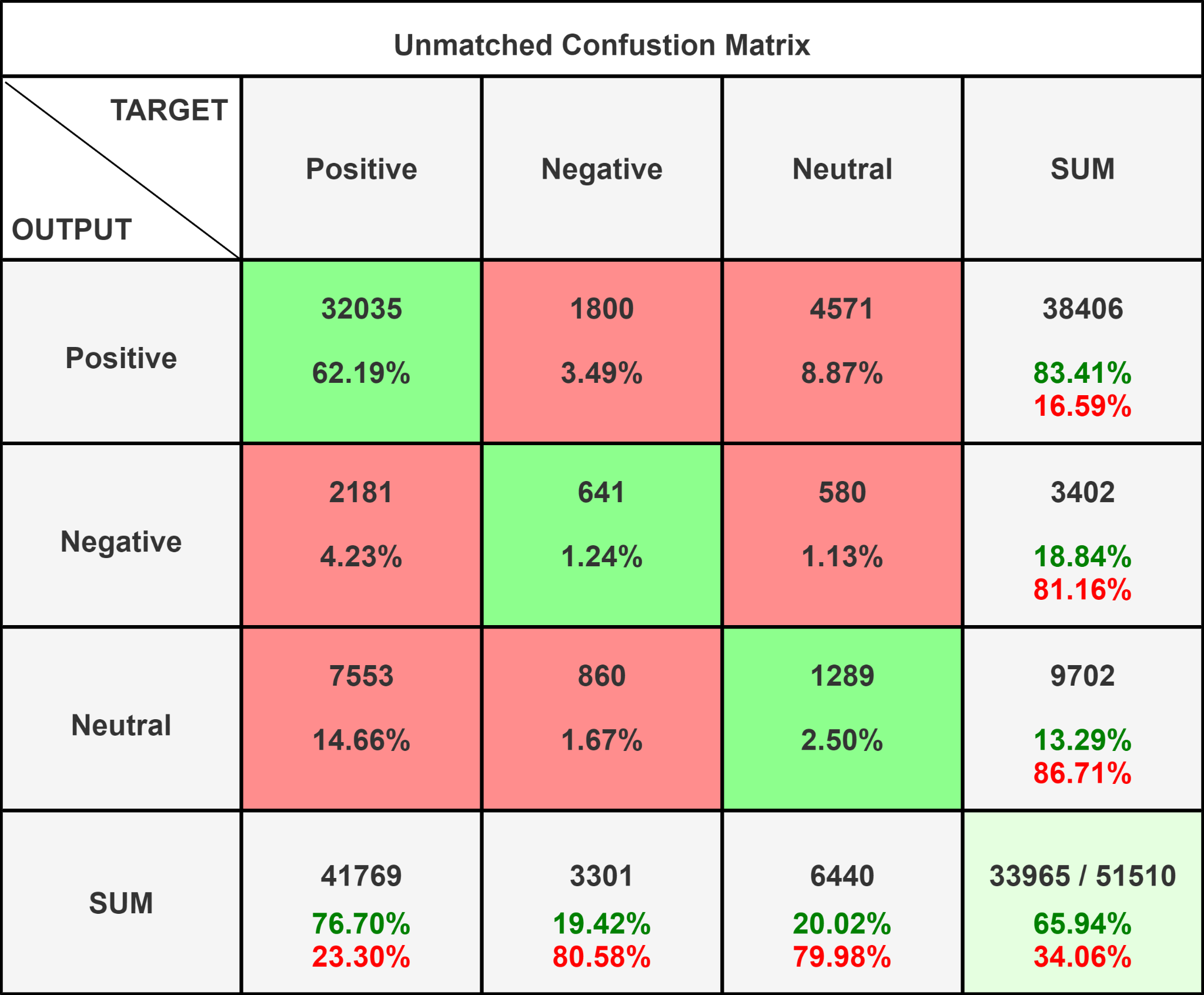
1. Accuracy of the model

The team has analysed the confusion matrix and worked out the accuracy of our model. The team compared the generated sentiment with the reviewer’s rating. Rating of 3.0 will be neutral, above 3.0 will be positive and below 3.0 will be negative.

The result of the “MATCH” category resulted in a 63.86% accuracy rate while “UNMATCH” had a higher accuracy rating of 65.94%. The confusion matrix can be seen in figure 12 and figure 13.



*Fig 12: Matched Accuracy*



*Fig 13: Unmatched Accuracy*

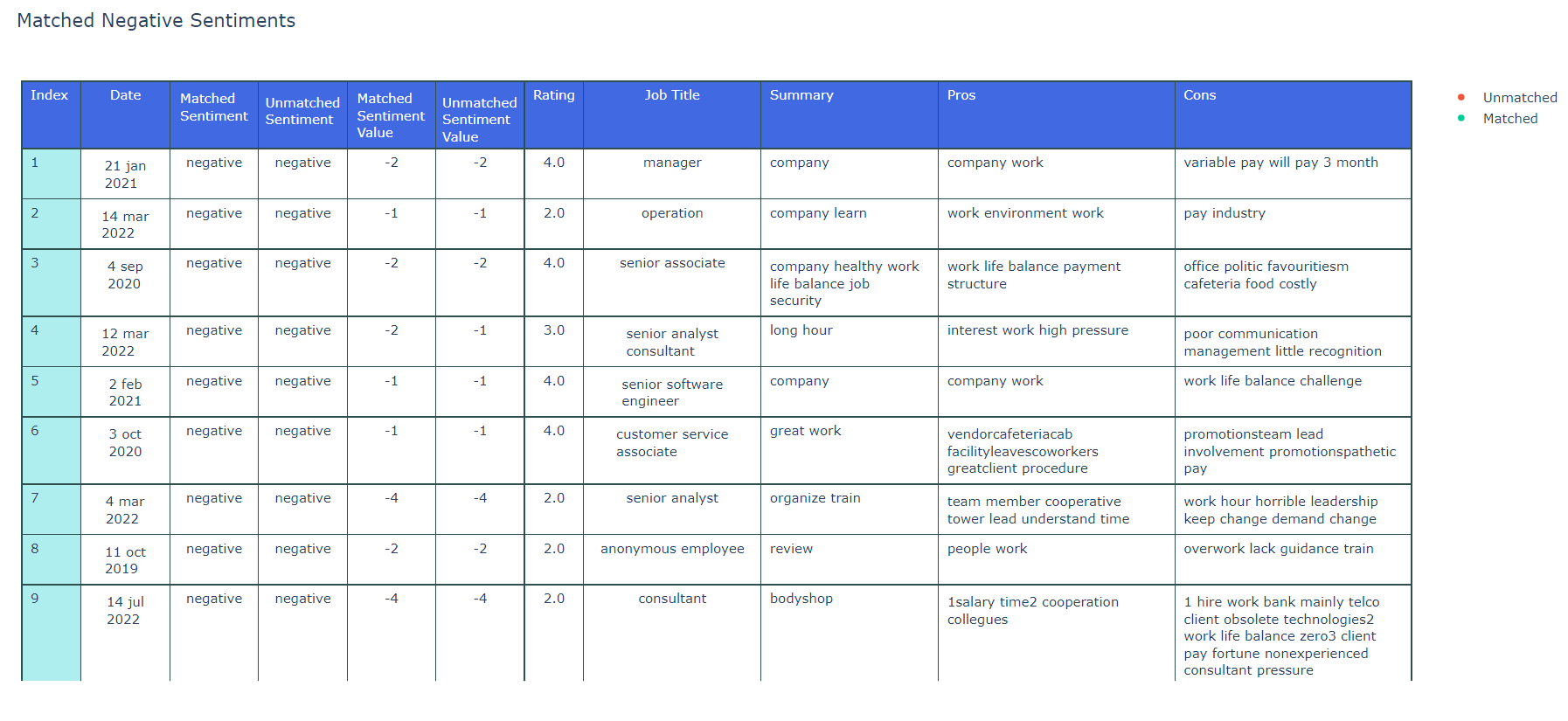
1. Insights

After analysing the data, all the sentiment analysis results are presented in the form of word cloud, table and scatter plot, showing the most commonly used words in each sentiment category, positive, negative and neutral. Each of the sentiment categories are separated into two categories: matched sentiment and unmatched sentiment.

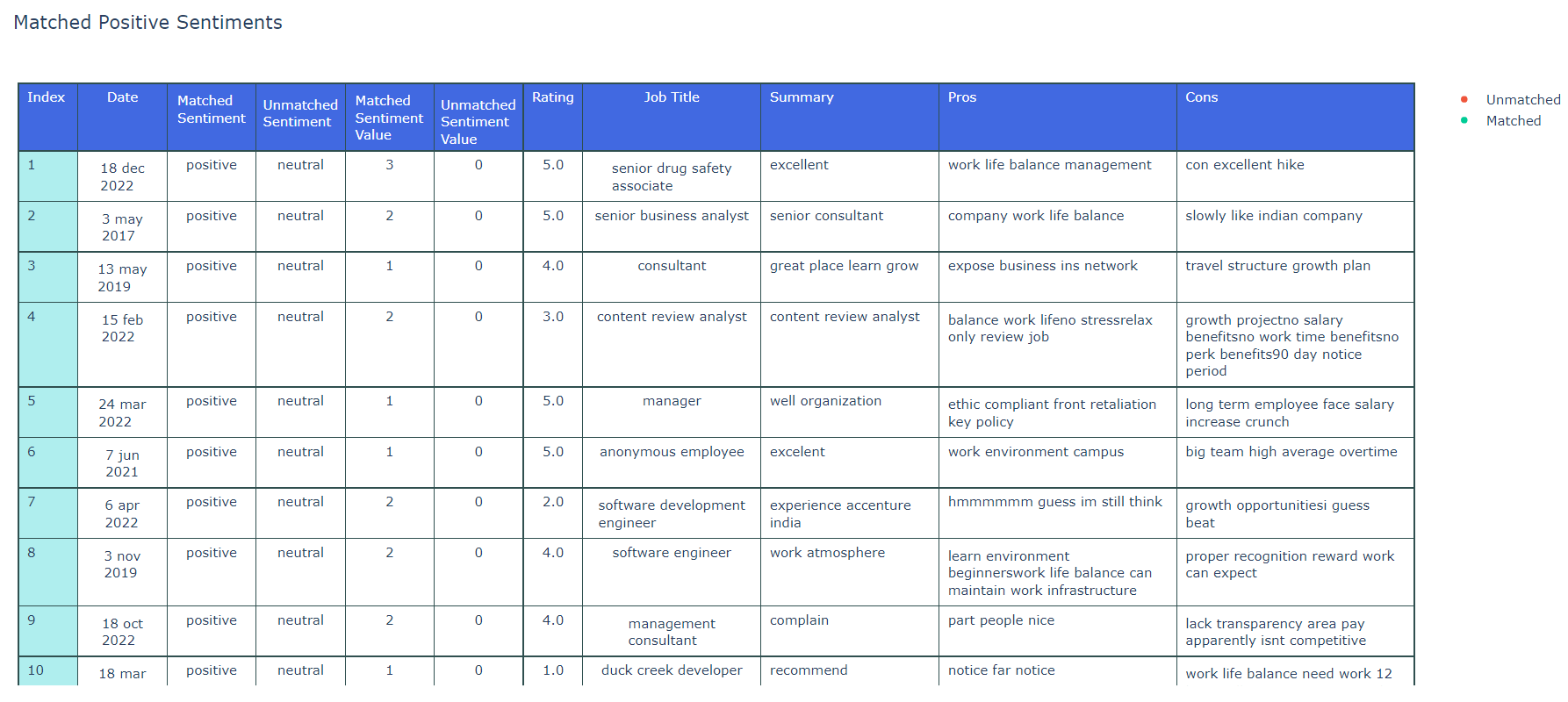
The matched sentiment displays the data from the analysis where the sentiment word appearing in the pros section is always positive and the sentiment in the cons is always negative. The unmatched sentiment is to display data based on the words used no matter whether it appears in the pro section or the con section.

1. Table View:

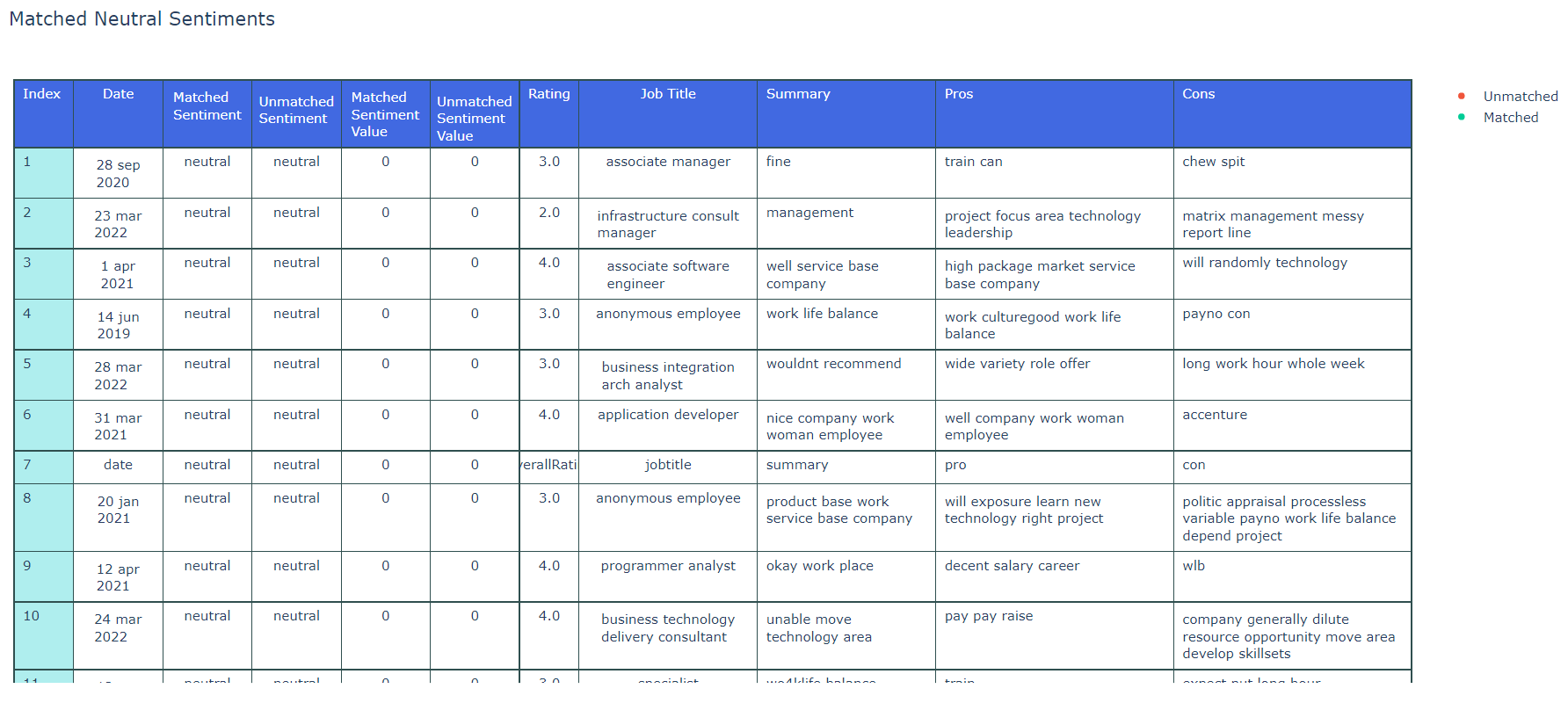
The table view will display the sentiment value and result of the dataset. It displays the date, matched sentiment, unmatched sentiment, matched sentiment value, unmatched sentiment value, rating, job title, summary, pros and cons.



*Fig 14: Table of Matched Negative Sentiment*



*Fig 15: Table of Matched Positive Sentiment*



*Fig 16: Table of Matched Neutral Sentiment*

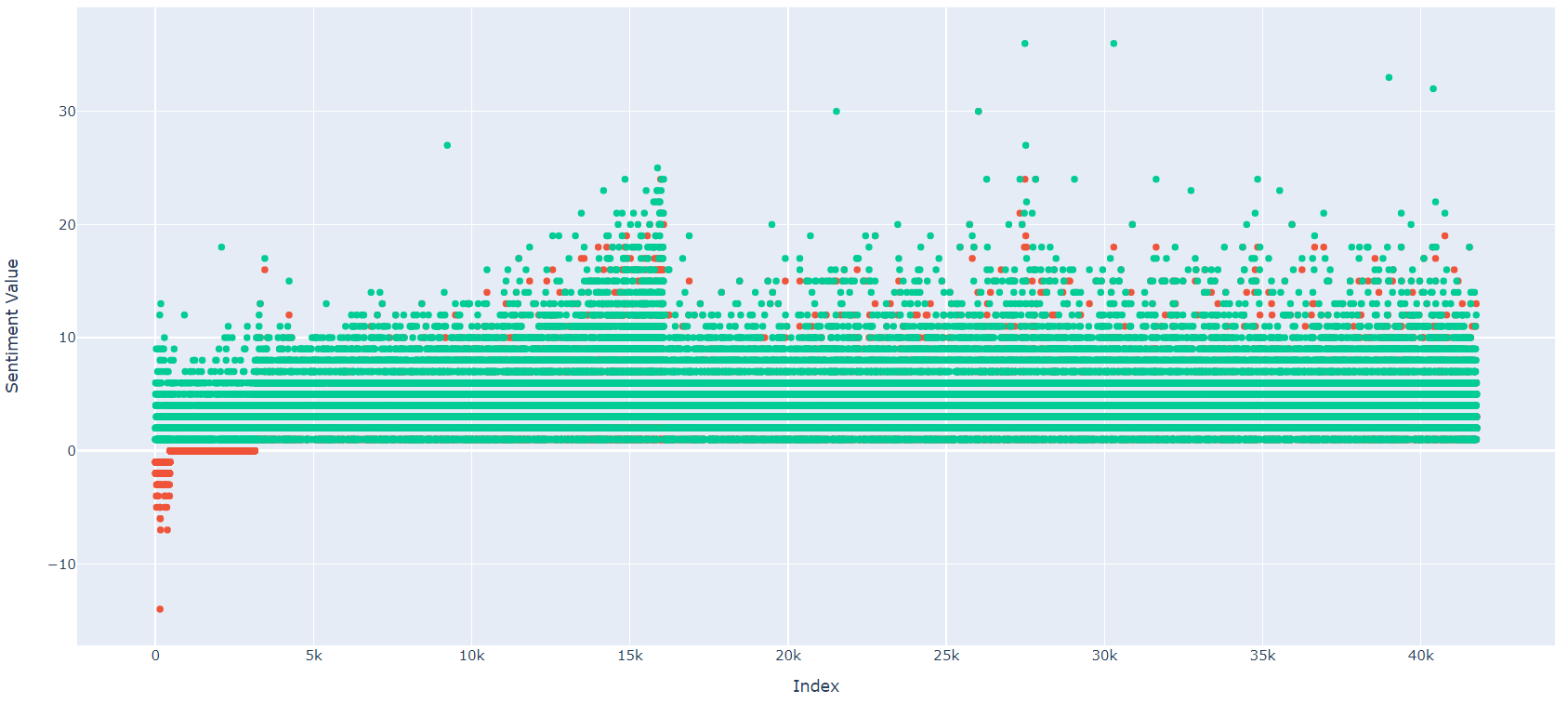
1. Scatter Plot

The scatter plot displayed shows the difference of matched and unmatched sentiment value results for every sentiment category. Green dots represent matched sentiment values, while red dots represent unmatched sentiment values.



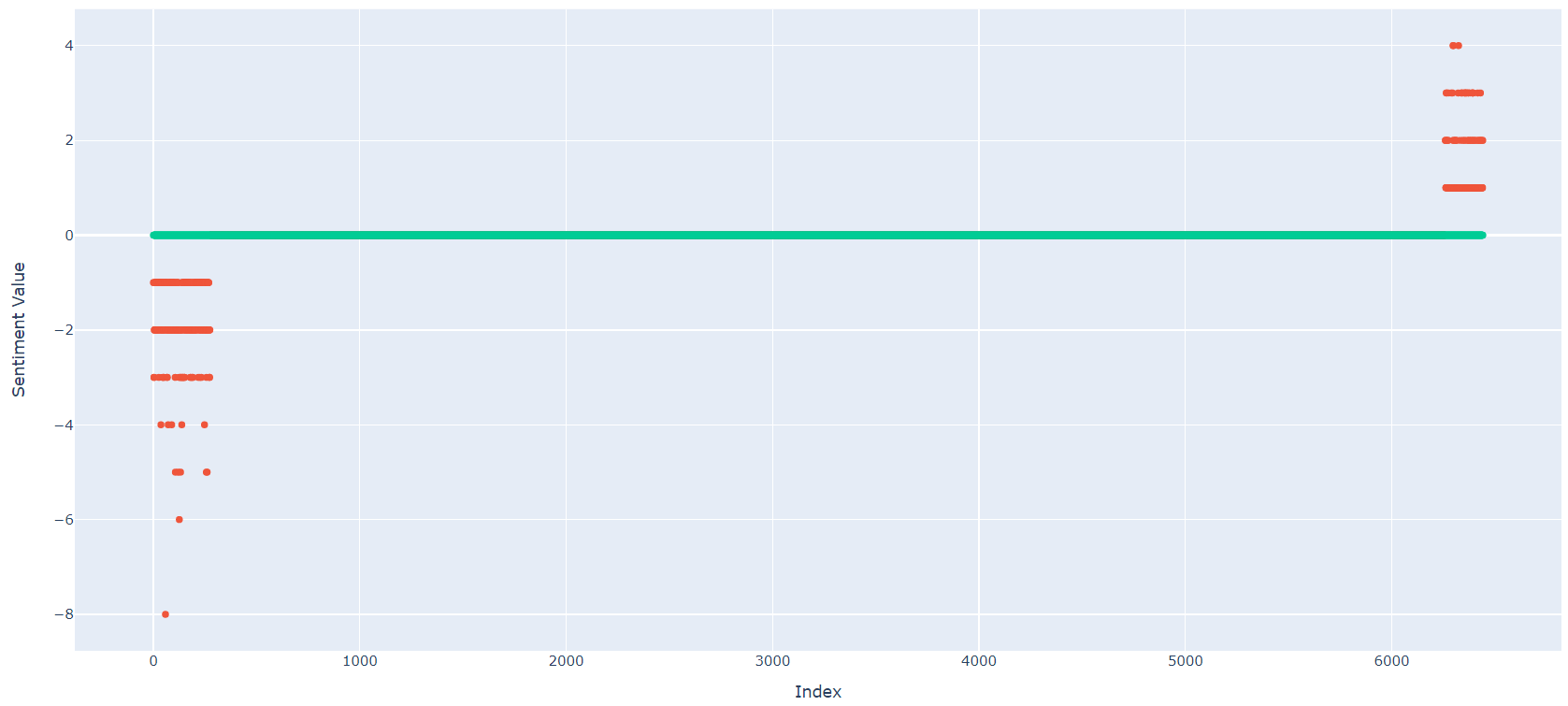
*Fig 17: Scatter Plot of Negative Sentiment*

As seen in the scatter plot of negative sentiment, most of the matched and unmatched values range from -1 to -5. Some outliers seen are the unmatched sentiment values that are at the top of the scatter plot, which are more positive than the matched sentiment values. This shows that when we do not force the sentiment values to be negative due to it being a negative review, the words used could be more positive than expected.



*Fig 18: Scatter Plot of Positive Sentiment*

Similarly in the scatter plot for positive sentiment, there are outliers showing that when unmatched, positive reviews may be more negative than expected.



*Fig 19: Scatter Plot of Neutral Sentiment*

Lastly, for neutral sentiments, most of them are at value 0, a true neutral sentiment. Some unmatched values can be seen to be both more positive or negative than expected.

With differences seen between matched and unmatched values, it highlights the importance of looking at both matched analysis and unmatched analysis, to get further insight on the data.

1. Word Cloud

The word cloud takes in the result data from the word count analysis. The matched word cloud is getting the result of the matched sentiment word whereas the unmatched word cloud is getting the result of the unmatched sentiment word.



*Fig 20: Matched Pros Word Cloud*



*Fig 21: Matched Cons Word Cloud*



*Fig 22: Unmatched Pros Word Cloud*



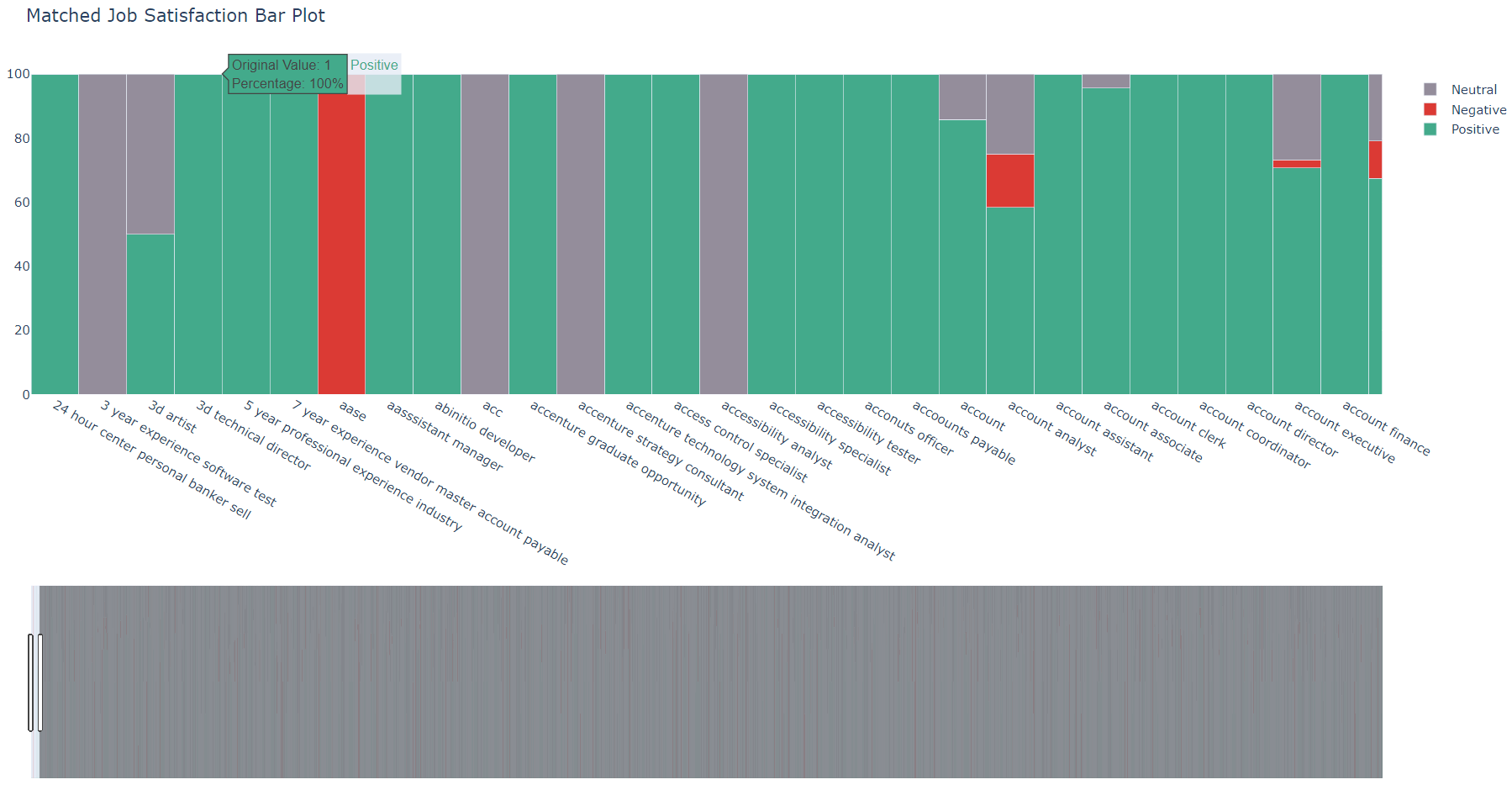
*Fig 23: Unmatched Cons Word Cloud*

Using the different types of charts and tables, some useful view is displayed to have a quick look at the sentiment analysis result.

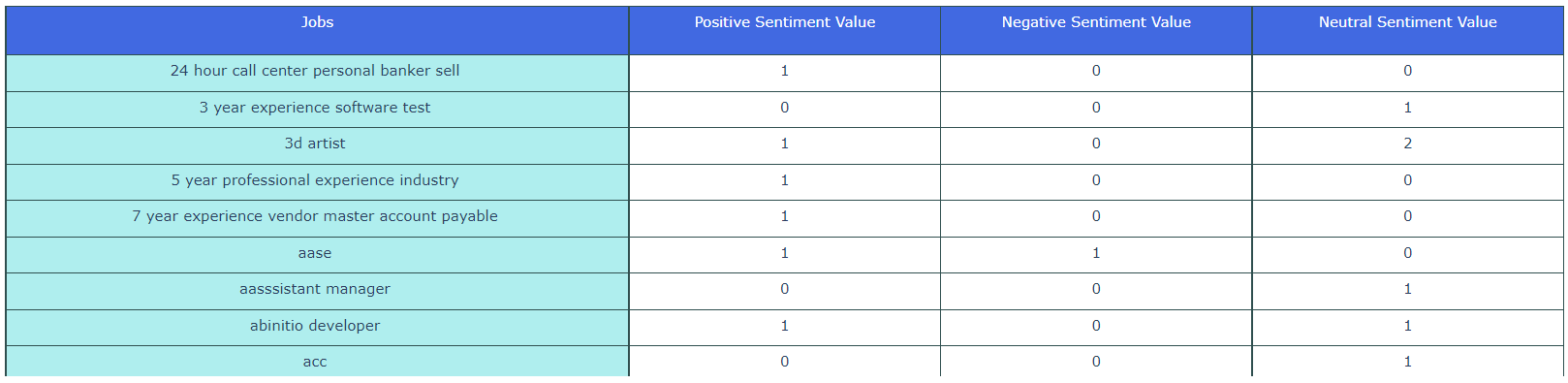
1. Job Satisfaction

Using the result from the job reducer, the data has been displayed in a bar plot and a table. A bar plot with different colours was used, allowing the viewer to easily compare the sentiment values for each job. The table will show the matched job satisfaction for every job type and the number of positive, negative and neutral sentiment values.

The sentiment category (neutral, negative, positive) for different job types is also displayed in the bar plot counted by the percentage of each sentiment value. Figure 24 and figure 25 show the matched job satisfaction and unmatched job satisfaction. Both of the bar plots show roughly the same results, with slight differences in certain jobs.

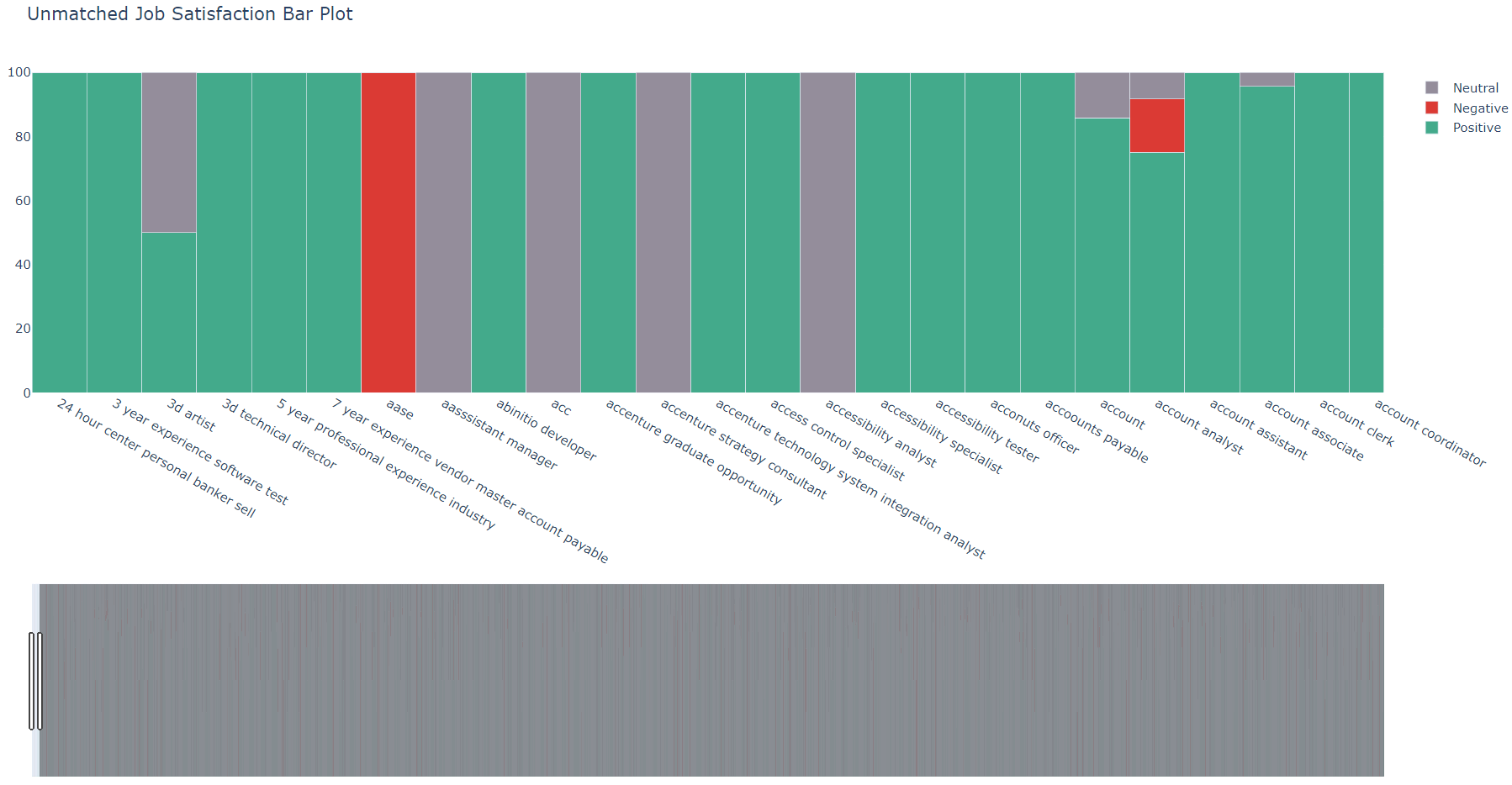


*Fig 24: Matched Job Satisfaction Bar Plot*



*Fig 25: Matched Job Satisfaction Table*

Each bar plot comes with a table of all the mapped info on the bar plot, for easy browsing and viewing of data.

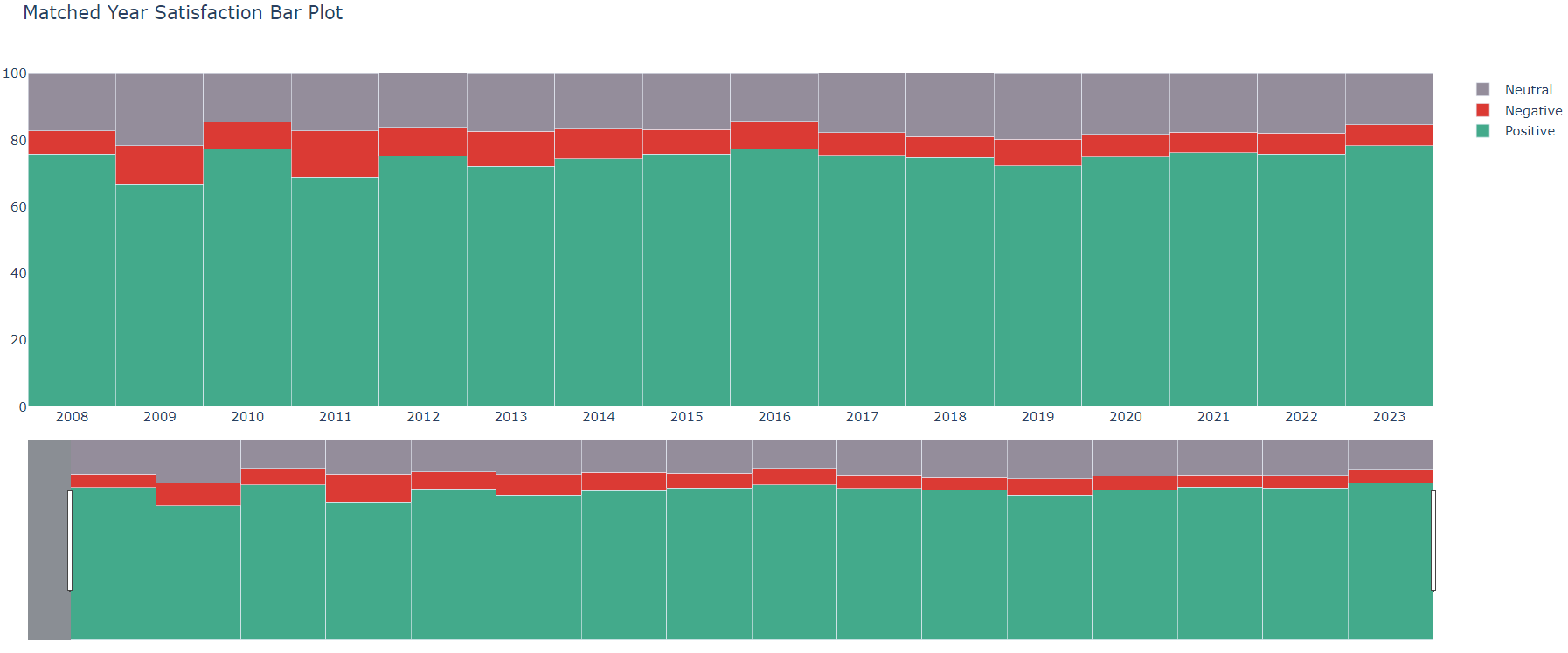


*Fig 26: Unmatched Job Satisfaction Bar Plot*

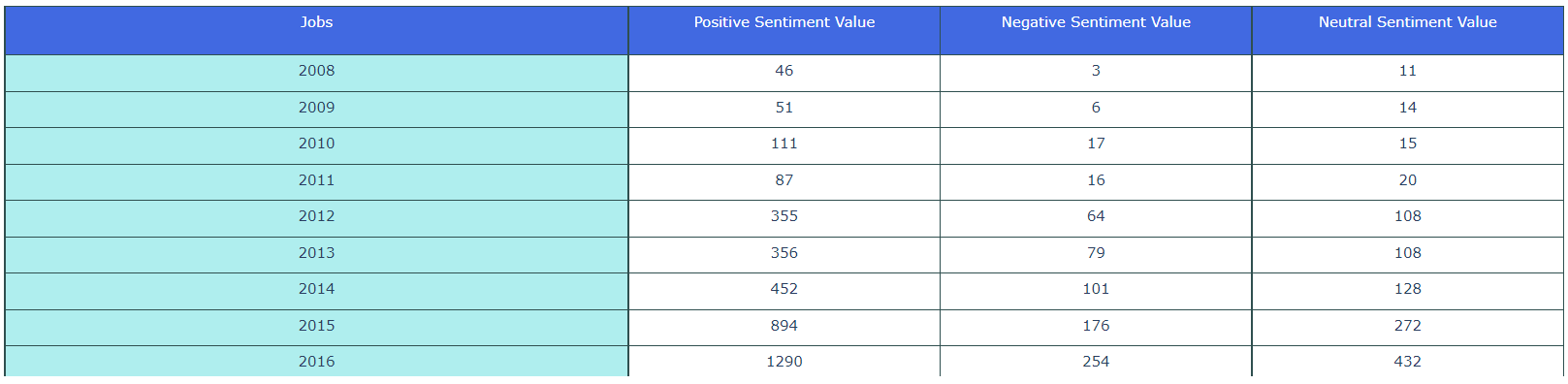
1. Year Satisfaction

Using the result from the year reducer, the data has also been displayed in a bar plot and table. The table shows the matched sentiment values across the year for every sentiment category.

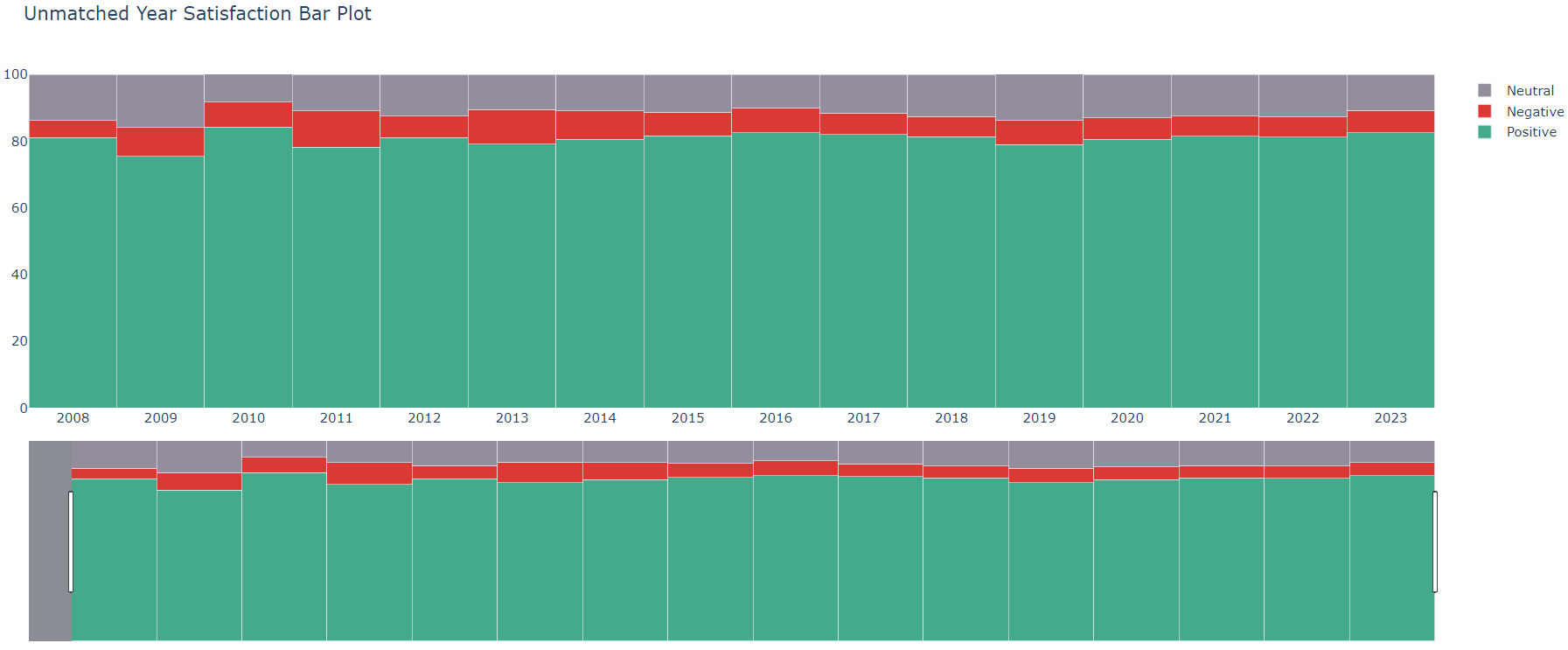
The sentiment category (neutral, negative, positive) across the year is also displayed in the bar plot counted by the percentage of each sentiment value. Figure 27 and figure 28 show the matched year satisfaction and unmatched year satisfaction. It shows that both of the bar plots show roughly the same result.



*Fig 27: Matched Year Satisfaction Bar Plot*



*Fig 28: Matched Year Satisfaction Table*



*Fig 29: Unmatched Year Satisfaction Bar Plot*

# CONCLUSION

1. Challenges

Some challenges of using a lexicon based approach to do sentiment analysis is that lexicon based methods are very strict and often lack context. It is unable to understand irony, sarcasm and various other expressions that humans might say.

This approach is also weak to spelling mistakes which can cause poor analysis of the sentiment.

1. Summary of Project

All in all, the team has built a model from scratch, using basic tools such as Hadoop, python and java.

While the accuracy is insufficient at 65.93% for unmatched analysis and 63.85% for forced matching analysis.

From our analysis, there is a need to stem the reasons for the high amounts of false negatives.

1. Potential Future Work

Machine-learning training model on the same dataset and observe the difference in results against the current model.

Explore more data pre-processing methods, such as Stemming, to attempt to improve the dataset results.

1. Improvements

An improvement that can be done to this project is to improve the data cleaning process, as lexicon based models need good data cleaning for better accuracy.

Another improvement to the research could be increasing the amount of the dataset to 300k - 400k by scraping from other social media sites like Indeed or LinkedIn. Incorporating this would make the result more reliable and accurate.

Lastly, other forms of analysis can be incorporated into the model.

REFERENCES

[1] Yilmaz, B."Sentiment Analysis Methods," AIMultiple, August 8, 2022, [Online]. Available: https://research.aimultiple.com/sentiment-analysis-methods/. [Accessed: Apr. 01, 2023].