
# IS450: Text Mining and Language Processing

## Final Project Report

## 13 April 2023

## Project Title:

## Helping tech companies with branding and recruitment strategies

## *Group 1*


| Team Members: | Student Email ID: |
|---|---|
| Sanofer | sanofer.2020@scis.smu.edu.sg |
| Ho Sheng Yun, Joslyn | joslyn.ho.2019@scis.smu.edu.sg |
| Jennifer Wang Xue'Er | xewang.2020@scis.smu.edu.sg |
| Sneha Murali | snehamurali.2020@economics.smu.edu.sg |
| Illakkiya D/O Ravindran | illakkiyar.2019@scis.smu.edu.sg |
| Sumer Singh Grewal | sumersinghg.2019@economics.smu.edu.sg |

# 1. Introduction

Company profits are being affected by the widespread unhappiness among workers. According to a recent survey, 60% of workers reported being unhappy with their jobs, while 19% claimed to be miserable (Collins, 2022) and only 15% reported being engaged at work (Adeline, 2022). This unhappiness has had a significant impact on employers, with business units that have higher worker engagement having 23% higher profits, while low worker engagement has resulted in a $7.8 trillion loss in productivity or an 11% decline in global productivity (Collins, 2022).

This problem seems to apply more to the tech-industry which has experienced a large number of lay-offs recently. Alphabet, IBM, Meta and Microsoft have let go of 5%, 1.5%, 13% and 11% of their global workforce respectively (Ivanova, 2023). Additionally, employees that remain at these companies do not seem too pleased with their work environment either. According to a 2022 Forbes survey, nearly 4 out of 5 professionals are considering a move away from Oracle, Microsoft and IBM while 46% of Google employees are looking to move to another company as they are unhappy with their compensation (Kelly, 2022).

The employee dissatisfaction and news of major layoffs result in companies wondering how these factors affect their recruitment and retention strategies as well as their branding strategies. Our group aims to help these companies plan these strategies by answering the following questions:

1. How does the news of the layoffs affect the public's view of the company?

   Our Approach: Conduct Sentiment Analysis of tweets is used to analyse the public sentiment of these companies and how they change over time.

2. What are the aspects of the company that employees are satisfied or dissatisfied with?
   Our Approach: Conduct Topic Modelling on the pros and cons of a Glassdoor Reviews dataset to identify the major factors that are causing employees' satisfaction and dissatisfaction.

To illustrate, if a company's overall sentiment is lower in comparison to its counterparts, it would be advisable for the company to implement a branding strategy to enhance its reputation. To improve the sentiments, one of the viable solutions would be to focus on boosting employee satisfaction. This could be achieved by addressing the root causes of employee discontentment.

# 2. Dataset

As mentioned in the Introduction, our group has used 2 related datasets for the project. In this section, we will describe the datasets in detail.

Glassdoor Reviews

The Glassdoor Reviews Dataset was obtained from Kaggle and can be found here. The original dataset had a total of 18 columns and a total of 838,566 rows. However, to identify

the aspects of the company that employees like and dislike, we will be focusing on the firm, pros and cons columns. As we want to identify these aspects for each company, we had to split the original dataset into 5 different datasets. The number of rows for each is shown below:

| Company | Number of Rows |
|---|---|
| Apple | 20,797 |
| Google | 15,995 |
| IBM | 60,436 |
| Microsoft | 26,675 |
| Oracle | 31,941 |

*Table 1 : Glassdoor dataset information*

One example of a row in each dataset is as follows:

| Firm | Pros | Cons |
|---|---|---|
| Apple | The people we work with are great and I can't imagine life w/o the products. However, they are a tool that needs to be taught and used. | You have to be careful because this job can take over your life if you are not careful. You need to learn to separate the 2 and learn to say no when it is necessary. |

*Table 2 : Example of Glassdoor dataset row*

To identify the aspects that employees like about the company, we will perform topic modelling on the pros column. To identify the aspects that employees are dissatisfied with, we perform topic modelling on the cons column. Thus, for each of the 5 companies, we will split the dataset further into 2, one for pros and one for cons, resulting in a total of 10 subsets of the original dataset.

Tweets

The tweets dataset was generated by scraping tweets that contain the company names as well as words from a list of keywords such as “layoffs”, “stock”, “work” and “salary”. We had set the time period to be from December 2022 to February 2023 so as to capture the period before, during and after the layoffs were announced in January 2023. Scraping was done using the snscrape package in Python to capture the Tweet content, date and user. The data was then added to the dataframe using Pandas.

Number of tweets for each company is shown in the table below:

| Company | Number of Rows |
|---|---|
| Apple | 2982 |
| Google | 2992 |
| IBM | 2347 |

| | |
|---|---|
| Microsoft | 2953 |
| Oracle | 2842 |

*Table 3 : Tweet dataset information*

An example of a row in this dataset is:

| Date | User | Tweet Content |
|---|---|---|
| Jan 2023 | cherthedev | Apple's culture encourages a toxic environment that obscures the existence of discrimination, harassment, layoffs, and labour issues.<br><br>WA DOL said I was constructively discharged because they didn't fix the illegal conduct I reported. I hope NLRB agrees. |

*Table 4: Example of Tweet dataset row*

Data Challenges

For the glassdoor dataset upon inspection we found that the data was not clean, where some words were merged together. Some examples of this would be “colleaguesPoor” and “financeVery” as seen in Figure 1. In order to tackle this problem we used the WordNinja library to help us split the words if a particular word is not present in the dictionary.

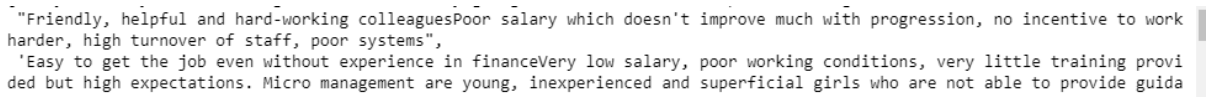



*Figure 1: Words that are combined*

One of the challenges for the Tweets dataset is the presence of tweets written by bots. To handle this, our group analysed the “user” column to identify bots.

Another big challenge faced when using tweets as a dataset was semantic ambiguity. We used the company name ‘Apple’ as a keyword and also other company/job specific words such as ‘stock’ and ‘work’. Even then, we obtained several tweets that used these words in different contexts and meanings due to the lexical ambiguities present. For instance, one of our tweets involved a cooking recipe including the words ‘apple’ and ‘vegetable stock’ which passed our tweet extraction criterion. Hence, we believe the insights obtained could have been more coherent if we had an effective method to filter out these tweets which were semantically irrelevant. However, upon inspection we realised that only around 25 tweets were regarding the fruit, considering it is a very small number in comparison to the total number of tweets, the impact was negligible.

# 3. Solution Overview

### 3.1. Task 1: Sentiment Analysis

For sentiment analysis, we utilised VADER and Naive Bayes Classification models. The rationale behind selecting these models would be that we wanted to explore how a dictionary and a non-dictionary model would perform in identifying the sentiment of the tweets. For the VADER model, we utilised SentimentIntensityAnalyzer to obtain the compound score for the tweets in a particular month for a particular company. For Naive Bayes Classification model, since our tweets dataset does not contain any labels, we extracted an additional 300 tweets and manually labelled them and trained the classifier using this training dataset. As we wanted to compare the performance of both the models to decide which model to base our findings on, we labelled 100 tweets each for 'Microsoft' and 'Oracle' for the time period of December 2022. We then proceeded to utilise both the models separately and obtained its performance.

### 3.2. Task 2: Topic Modelling

For topic modelling, we compared the performance of 2 different models Latent Dirichlet Allocation (LDA) and Latent Semantic Analysis (LSA). As we have to compare the performance of these models across a total of 10 datasets (5 companies, each with a pro and a con model), our group decided that there should be certain constants set in place to allow for better comparison. We decided to set the number of topics to 3 and to standardise the pre-processing steps for both models. We compared the 2 models based on coherence scores and human evaluation scores. Coherence scores are calculated based on the degree of semantic similarity between the words in the topics and higher coherence scores are generally associated with more accurate and interpretable topics. The human evaluation was conducted using 5 evaluators and the final score is averaged from all 5 evaluations. The 5 evaluators were asked to review the topics generated by the LDA and LSA models and score them based on their interpretability and usefulness in capturing the themes in the data. Our final insights were then derived from the best model.

# 4. Solution Details

### 4.1. Task 1: Sentiment Analysis

#### 4.1.1 Pre-Processing Techniques

For the VADER model, the pre-processing techniques that were utilised were: removal of URLs, removal of HTML reference characters and removal of Twitter handles.

Some of the tweets present in the dataset contained website URLs which we felt were irrelevant to our task and hence we decided to remove it. In addition, we also removed HTML reference characters present in the tweets such as '&amp'. Lastly, we removed twitter handles as we felt that it does not have an impact on our task. We did not perform pre-processing techniques such as stemming, stopword removal, punctuation removal and

lowercasing. We decided to not perform lowercasing as we felt that some words in the tweets could be capitalised to express their displeasure and hence we decided to leave it as it is. We did not perform stopword and punctuation removal as we felt that if we removed them it could potentially alter the meaning of the tweets and this could affect our results. Similar to stemming, we decided to not perform stemming as stemming the words could change the meaning of the tweet. Since the VADER model could handle emojis and emoticons, we did not perform that particular pre-processing step.

For Naive Bayes Classification model, the pre-processing techniques that were utilised were: removal of URLs, removal of HTML reference characters, removal of Twitter handles and replacing emojis and emoticons with words. The rationale behind performing these pre-processing techniques is explained above. Similar to the VADER model, we did not perform stemming, stopword removal, punctuation removal and lowercasing and the explanation is given in the above paragraph. Even though Naive Bayes could handle emojis and emoticons we decided to replace it with words. The rationale behind this step would be that since we are working with a limited dataset and were unsure if the pre-existing dictionary would contain the particular emojis and emoticons. Hence, we decided to replace it with words instead.

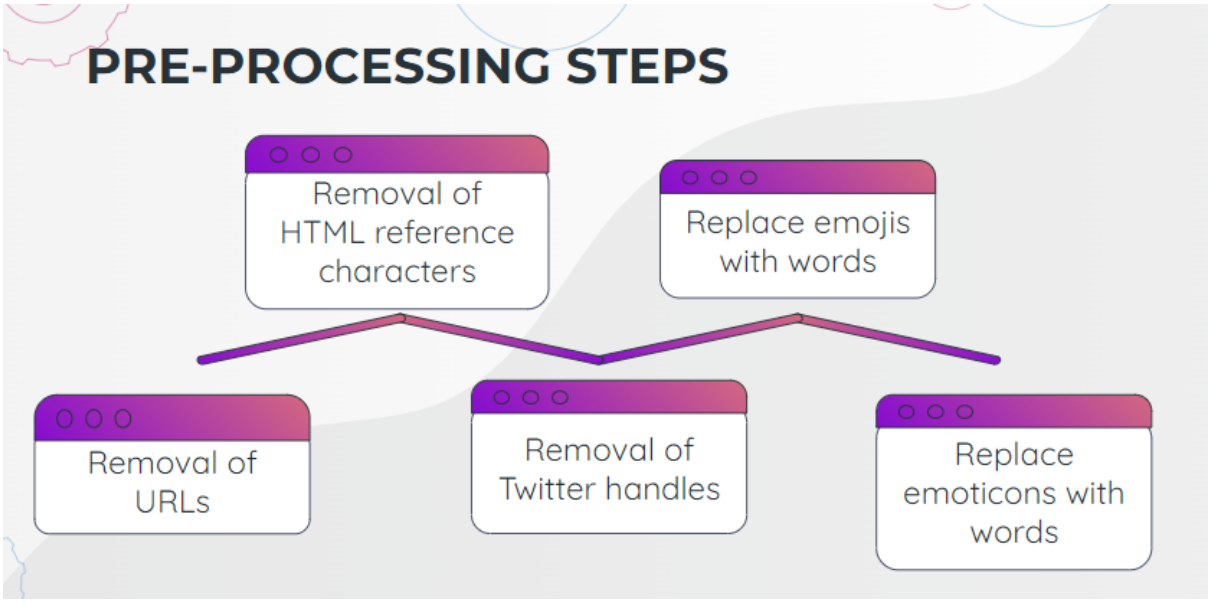



*Figure 2 : Flowchart of preprocessing steps for Sentiment Analysis*

#### 4.1.2 VADER Model

After the pre-processing steps were completed, the tweets were separated by month so we could analyse sentiments across December 2022, January 2023 and February 2023 for all companies. SentimentIntensityAnalyzer from the nltk.sentiment.vader toolkit was used for the analysis. For each tweet in the dataset, the analyser predicted a value of neutrality, positivity, negativity as well as a compound score, which consolidated the overall sentiment of a tweet. The range of the compound score was from -1 to 1, where values below 0 indicated a ‘negative’ tweet, values equating to 0 referred to ‘neutral’ tweets and values above 0 indicated a ‘positive’ tweet. In addition, the average of the compound scores of all

tweets for each company in a particular month were also collected so as to analyse time-series plots of sentiment across the different companies.

#### 4.1.3 Naive Bayes Classification Model

As mentioned in the Solution Overview, since our dataset did not contain any labels, we created our own training dataset by extracting additional 300 tweets and manually labelled the tweets as ‘Positive’, ‘Negative’ and ‘Neutral’. We then proceeded to perform the same pre-processing techniques on the training dataset. Following which, we performed Naive Bayes classification utilising the nltk package. First, we created the training data following which we extracted the features from the training dataset and the end product is a dictionary of tuples containing the words in the tweet and the respective label that we manually labelled. We then proceeded to train the Naive Bayes classifier with our training dataset. With the trained classifier, we used it to perform predictions on our testing dataset that we have split according to the three different time periods. The tweets predictions made for the three different time periods were saved to a ‘csv’ file. Using the results that we have obtained, we proceeded to find out how many of those predictions were positive, negative and neutral respectively. We plotted this trend on Tableau for each of the companies so that we could visualise the trend of the positive, negative and neutral tweets for the various companies according to the different time periods.

### 4.2. Task 2: Topic Modelling

#### 4.2.1 Pre-Processing Techniques

The following pre-processing techniques were used: tokenization, POS-Tagging, lower-case, select alphabets only and stopword removal.

For POS-Tagging, we decided to only include nouns as nouns are able to provide more information on the content of the reviews. It also helps us to make the topics more interpretable as it removes adjectives and adverbs that appear very frequently . This way we can direct our attention to the aspects of the company talked about in the reviews. For stopword removal, we used the NLTK list of English stopwords and also extended the list to include words such as the names of the companies and other words such as “pros”,”cons”. This is because the names of the companies appear in many of the reviews but are not helpful in identifying the aspect of the company that employees are satisfied or dissatisfied with and words such as “pros” and ”cons” appeared frequently but did not offer significant insights.

We removed stopwords to remove words that do not carry much meaning on their own and hence reduce noise in the data, lowercase all words to ensure that two words that are the same are not treated as different words due to capitalization, tokenize words as our model operate at the word level and select alphabets only to remove non-alphabetic characters that is not useful for analysis.

We did not include techniques such as stemming or lemmatization because we felt that multiple words could have the same base word but the original words have different meanings. One example of this are the words “employee” and “employment” which could represent different aspects of the company but have the same base word “employ”.

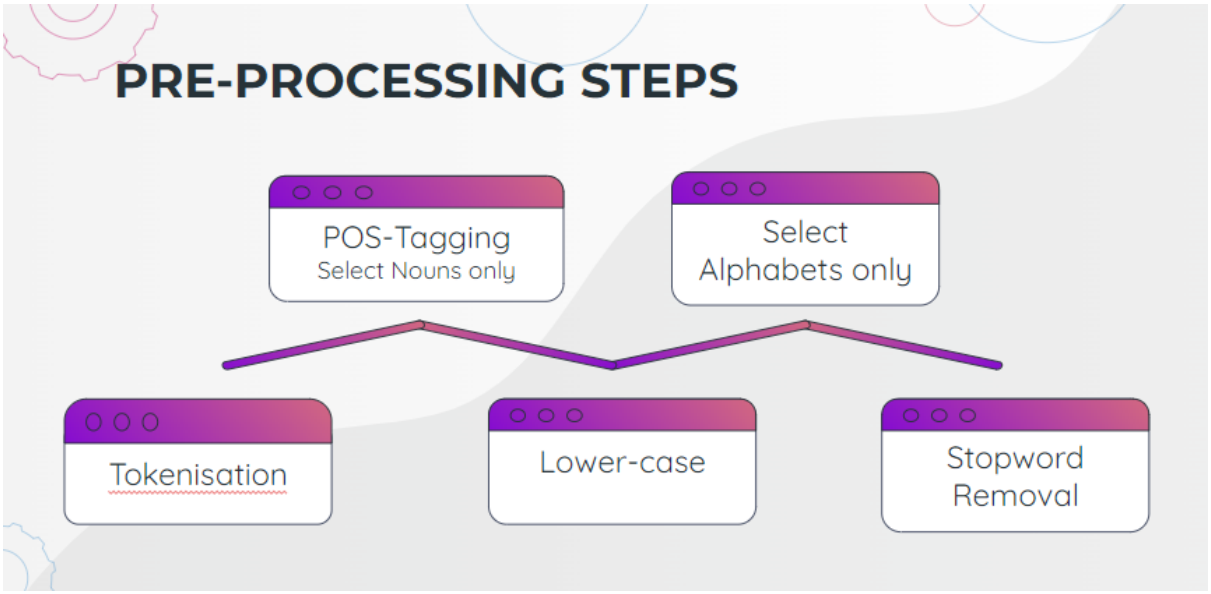



*Figure 3 : Flowchart of preprocessing steps for topic modelling*

#### 4.2.2 Selecting the number of topics for comparison and for final chosen model

As mentioned in the Solution Overview, we decided to keep the number of topics constant at 3 to allow for easier comparison between models. This number was selected as we wanted to gain a high level understanding on the aspects employees like/dislike so we choose a smaller number of topics with broader details instead of having a huge number of topics with more granular detail. By using a smaller number of topics, it was also less complex and less time consuming to compare the two models.

Once we completed the comparison, we concluded that the LDA model performed better than the LSA model based on coherence scores and human evaluation scores, meaning that the model was more coherent and interpretable and did better in capturing the themes in the data. We then calculated the coherence scores against the number of topics for the LDA model with a range of 2 to 10 topics. This was determined by examining the coherence scores for each number of topics from 2 to 10 and selecting a suitable number of topics that resulted in consistently higher coherence score across all 10 datasets. We found that 3 topics consistently resulted in the higher coherent scores across all datasets (Refer to Appendix B Figure 2). Thus, we decided to build our final LDA model using 3 topics.

#### 4.2.3 Building the LDA model

The LDA model was built using the gensim LDA model, which provides an implementation of the LDA algorithm. Before creating the LDA model, we first conducted a number of pre-processing steps mentioned above on the original documents.

Afterwards, we created a dictionary using the corpora.Dictionary() method, which created a mapping between each unique word and their unique integer ID. Vectors were then created using the doc2bow() method, which counts the frequency of each word in the document and

maps them to their corresponding integer IDs. These vectors were used to generate the Term Frequency-Inverse Document Frequency (TF-IDF) values for each word in the corpus, which provides a measure of how important each word is in each document.

Using the generated vectors and dictionary, we then applied the gensim LDA model for each pro and con of a company. The LDA model was trained on the pre-processed text data to identify the underlying topics and their corresponding word distributions. Coherence scores were computed using the CoherenceModel() and get_coherence() methods.

#### 4.2.4 Building the LSA model

The LSA model was built using the gensim LSA model, which provides an implementation of the LSA algorithm. The pro-processing steps used in the LDA model were also applied to the data for the LSA model.

Firstly, we created a dictionary and bag-of-words representation of the reviews using the corpora.Dictionary() and the doc2bow() methods. Then we built the LSA model using the LsiModel() method. The “num_topics” parameter specifies the number of topics to extract from the corpus, and in this case, we have chosen to extract 3 topics based on the same reasoning as the . The “id2word” parameter specifies the dictionary created earlier.

Using the generated vectors and dictionary, we then applied the LSA model for each pro and con of a company. The LSA model was trained on the pre-processed text data to identify the underlying topics and their corresponding word distributions. Afterwhich the coherence scores were computed using the CoherenceModel() and get_coherence() methods.

# 5. Results and Analysis

## 5.1. Task 1: Sentiment Analysis

We wanted to compare VADER and Naive Bayes classification models using evaluation metrics to see how these models perform in correctly predicting the sentiments of the tweets. As our dataset was not initially labelled, we had to manually label some of our data so as to obtain the goldtruth (true labels). Hence, due to time constraints, we limited our labelling to the first 100 tweets for Microsoft and Oracle each in December 2022.

We first compared the labelled tweets with the labels that were predicted by the VADER model. For Microsoft, 54 tweets were correctly labelled out of 100 tweets and for Oracle, 46 tweets were correctly labelled out of 100 tweets. Hence, by computing 54+46/200, we obtained an overall accuracy of 50% for the VADER model. Following which, we proceeded to calculate the count of true positives, true negatives, false positives, false negatives for both Microsoft and Oracle respectively for each class of sentiment - positive, negative and neutral. We then calculated the precision and recall score for each class of tweets. Since our labelled dataset was slightly imbalanced with a larger number of ‘neutral’ tweets, we decided

to use a weighted method to calculate the overall macro-averaged precision, recall scores and F1. (Refer to Table 1 below)

As for Naive Bayes, we utilised the same classifier that we have trained using the 300 tweets that we manually labelled. We tested the classifier using the dataset containing the 200 tweets that we labelled as test data. We compared the predicted sentiments returned by the classifier with the goldtruth. For Microsoft, 37 tweets were correctly labelled out of 100 tweets and for Oracle 49 tweets were correctly labelled out of 100 tweets. Hence by computing 37+49/200, we obtained an overall accuracy of 43% for Naive Bayes Classification Model. Following which, we repeated the above-mentioned steps to obtain the overall macro-averaged precision, recall and F1 scores. (Refer to Table  below)

Snippers of the code used for above computations can be found in the Appendix section. (Refer to Appendix A Figures 1-10 for VADER and Figure 11-20 for Naive Bayes)

| Evaluation Metric | VADER | NB |
|---|---|---|
| Accuracy | 50% | 43% |
| Precision | 58% | 50% |
| Recall | 47% | 59% |
| F1 | 52% | 54% |

*Table 5 : Evaluation Metrics Scores for VADER and Naive Bayes*

Upon inspection of the evaluation metrics results above, we realise both VADER and Naive Bayes outperformed one another in different metrics. For the sake of our project, we prioritised accuracy as a metric (especially since we believed our dataset was not too imbalanced) and hence chose VADER as the better model to utilise for further analysis.

The visualisations that we have derived utilising Tableau for the results obtained from VADER and Naive Bayes are shown in the Appendix E from Figures 1 to 4 (VADER) and Figures 5 to 7 (NB). Based on the visualisations obtained from the model with better accuracy (VADER), we noticed 2 key insights from the sentiment analysis results.

Referring to Appendix C Figure 1 which shows the time series plots of the compound sentiment scores for all 5 companies we can observe that, firstly, all 5 companies faced a dip in compound sentiment scores in January 2023. Given the tech layoffs that were announced in January 2023 for all of these big technological firms, it is logical that the general public sentiments have dipped for that month. It was also interesting to note the variations in the dip for the different companies. For instance, Oracle faced a rather mild dip comparatively and based on our research, we believe it may have been because Oracle's layoffs were not as publicised compared to the other companies. We also noted that the compound sentiment scores of all companies increased after the dip, indicating that public sentiments returned to their original states within a month. For a business user, it may be insightful to

look into what activities these companies were engaged in post-announcement of layoffs to understand if they engaged in anything to uplift the public sentiments.

Secondly, we can observe the companies having a clear ranking in sentiment in this order (starting with the best): Oracle, Microsoft, Google, Apple, IBM. This order may potentially represent their standing in terms of public reputation/branding strategies with respect to counterparts and competitors. However, it is good to take note that bigger companies such as Google/Apple may perhaps face more negativity purely because their products and services are well known amongst the public and they are talked about more as compared to the rest of the companies.

## 5.2. Task 2: Topic Modelling

The final topics produced by the gensim models for each company's pros and cons are shown in the Appendix F from Tables 1 to 10 (LSA) and Tables 11 to 20 (LDA). For comparison between LDA and LSA, we decided to use coherence score as well as the average coherence rating given by 5 human evaluators. A rating of 1 would signify that the evaluator felt that the words in the topics were random and unrelated, while a rating of 3 signified that the words in the topics were coherent and meaningful. The final comparison results are as follows:

**LDA:**

| Company | Pro/Con Column | Coherence Score | Average Coherence Rating from (Human Evaluation) |
|---|---|---|---|
| Apple | Pro | 0.3671 | 1.6 |
| | Con | 0.6379 | 2 |
| Oracle | Pro | 0.3512 | 1.2 |
| | Con | 0.5874 | 1.6 |
| Google | Pro | 0.4446 | 1.2 |
| | Con | 0.5797 | 1.4 |
| IBM | Pro | 0.4131 | 2 |
| | Con | 0.6037 | 1.6 |
| Microsoft | Pro | 0.4062 | 1.6 |
| | Con | 0.5712 | 1 |
| | Average | 0.4992 | 1.52 |

*Table 6 : Evaluation for LDA Model*

**LSA:**

| Company | Pro/Con Column | Coherence Score | Average Coherence Rating from (Human Evaluation) |
|---|---|---|---|
| Apple | Pro | 0.2598 | 1.2 |
| | Con | 0.5974 | 1.8 |
| Oracle | Pro | 0.2921 | 1 |
| | Con | 0.6843 | 1.2 |
| Google | Pro | 0.4063 | 1 |
| | Con | 0.4760 | 1.8 |
| IBM | Pro | 0.3120 | 2.2 |
| | Con | 0.5935 | 1.6 |
| Microsoft | Pro | 0.2710 | 1.4 |
| | Con | 0.5508 | 1 |
| | Average | 0.4443 | 1.42 |

*Table 7 : Evaluation for LSA Model*

After calculating the average coherence score and coherence rating from all 10 models, we can conclude that LDA had performed better than LSA in terms of generating more coherent and meaningful topics.

To enhance our analysis on LDA, we incorporated bigrams as tokens using the gensim phrases function to detect phrases that are commonly occurring in the dataset. Some examples would be “co_worker”, “tuition_reimbursement” and “part_time”. However, we noticed that despite including bigrams, none of the top 20 words in each topic of the 10 models included any bigrams (Refer to Appendix F Tables 21 to 30). Thus, we concluded that the use of bigrams did not provide any significant value in improving the accuracy of our final LDA model and thus decided to remove the use of bigrams.

To enhance our analysis of the LDA model, we attempted to identify themes for each of the 3 topics. Using Tableau, we created an interactive visualisation to facilitate this process (Refer to Appendix G Figures 1 to 10). The Tableau interface is useful for business users in several ways, and in this case refers to tech company business owners.

Firstly, it provides a quick and easy way to understand the key themes that emerge from employee reviews. By visualising the most frequently occurring words in the positive and

negative reviews separately, the interface helps to identify common topics that customers are discussing. This can be essentially useful for identifying areas where the company is doing well and where improvements are needed.

Secondly, the interface allows users to compare the pros and cons of their own company's employee reviews with those of competitors. This allows for business users to gain a better understanding of their position in the market with regards to employee satisfaction. By analysing the common themes and sentiments across multiple companies' reviews, business users can gain a deeper understanding of the factors that are most important to employees and identify areas where their own company may be falling short or excelling as compared to other companies. The business users can identify the competitive advantages that their company has over other companies, and use this information to leverage on in recruitment and retention efforts.

After careful consideration, we were able to identify coherent topics. However, we excluded any words that lacked coherence or did not fit into a clear topic. Below are the topics that we identified for each company's pros and cons:

| Company | Pro/Con Column | Topics |
|---|---|---|
| Apple | Pro | 1. Work Culture<br>2. People and Work Environment<br>3. Employee Privileges |
| | Con | 1. (incoherent topic)<br>2. Work-Life Balance<br>3. Management |
| Oracle | Pro | 1. Workplace and Salary<br>2. Work Environment<br>3. Work-life Balance |
| | Con | 1. Hikes and Team<br>2. Work Politics<br>3. Management and Growth |
| Google | Pro | 1. Work Environment<br>2. Employee Benefits<br>3. Work-life Balance |
| | Con | 1. Work-life Balance<br>2. Work Hours and Pressure<br>3. Management and Politics |
| IBM | Pro | 1. Work Culture<br>2. Technology and Resources<br>3. Opportunities |
| | Con | 1. Monetary Benefits |

| | | |
|---|---|---|
| | | 2. (incoherent topic)<br>3. Work-Life Balance |
| Microsoft | Pro | 1. Work-life Balance<br>2. Technology<br>3. Career growth |
| | Con | 1. Work-life Balance<br>2. Management and Culture<br>3. (incoherent topic) |

*Table 8 : Topic identified for each Company's Pro and Con*

From the table presented above, companies can easily view what employees like and dislike about them, giving organisations an opportunity to view areas that need improvements and areas that are performing well.

While analysing the topics, we found some to be incoherent as the top words did not indicate a clear theme. Interestingly, all of these topics contained the word "nothing" in the top words. At first glance, this may suggest that there were no negative aspects, but upon closer examination, we found that the word "nothing" was often parked with other words to emphasise the severity of the issue, such as "nothing positive". Despite our efforts, we were not able to identify a common theme among these topics in the cons.

One thing that stands out is that employees across all companies highly value a positive work culture, which includes a pleasant work environment and a focus on employee well-being. Additionally, providing employee benefits, such as competitive salaries and comprehensive packages, is highly valued and can ultimately lead to a more engaged and productive workforce.

On the other hand, work-life balance is a common major area of concern for employees across all companies. Companies that prioritise work-life balance and offer flexible work arrangements tend to have more satisfied and productive employees. Another common concern is the lack of management and growth opportunities and can greatly impact employee retention rates with the company. Companies that provide opportunities for career development and offer transparent management practices tend to have a higher employee retention rate.

It is also worth noting that issues with management and workplace politics greatly impact employee satisfaction across companies. Employers who feel undervalued or unsupported by their managers may be less engaged and productive, while bad workplace policies can create a toxic work environment.

## 5.3 Overall Analysis

After conducting sentiment analysis, we discovered that Apple and IBM have been performing consistently poorer among the 5 companies. To improve their performance, we suggest that the management closely examine the topics that we have identified via topic modelling as 'Pros' to be emphasised and continued. Meanwhile, the topics identified for

‘Cons’ are looked closely into and addressed. By doing so, there is a high chance that Apple and IBM can perform better in terms of employee satisfaction.

By reviewing the topics identified as Pros, companies can determine which aspects of their organisation are highly valued by their employees. By reinforcing these positive aspects, companies can sustain high levels of employee contentment and involvement, as well as draw in and keep exceptional performers.

On the other hand, topics identified as Cons can help companies identify areas that require improvement. By taking steps to address the identified issues, companies can create a more positive working environment and increase employee satisfaction, which can have a significant impact on their overall happiness working for the organisation. Employees who are more satisfied with their jobs are more likely to be engaged and productive, which can ultimately benefit the company in terms of increased productivity, innovation and customer satisfaction. Additionally, happy employees are more likely to remain with the organisation long-term, which can reduce employee turnover and associated costs.

With the presence of social media, companies benefit greatly from it as they can stay updated on what people think about their company. This information can be used by companies to improve their performance and also do damage control quickly. By analysing public sentiments frequently, companies would be able to understand about the company performance better from time to time. If there is any negative sentiment, they can look into it and rectify it quickly.

Overall, the insights provided by the sentiment analysis and topic modelling can help companies make data-driven decisions to improve their organisational culture and ultimately contribute to the success of the organisation, thereby contributing to their long-term success.

# 6. Discussions and Gap Analysis

## 6.1. Task 1: Sentiment Analysis

Something we noticed after obtaining predictions from the VADER model was that it was limited in picking up sarcasm, possibly because it is a dictionary-based model. This may have affected our sentiment analysis as the predicted compound scores may actually have been higher and more positive for the sarcastic comments (ideally meant to be negative).

## 6.2. Task 2: Topic Modelling

One of the challenges that we faced when conducting topic modelling was deciding on the specific POS tags to run LDA on. While our team concluded that using only nouns would give us more information on the aspects of the company, we understand that there are some verbs that may provide crucial information as well. As a result, some aspects may not have been adequately represented in the topics, making it challenging for the company to develop a retention strategy and enhance employee satisfaction. To overcome this, we could attempt to run the topic models on different combinations of POS tags to discover new insights that may have been missed.

Another challenge was that our group members lack the domain knowledge needed to correctly identify the topics that were presented from the topic modelling. Thus, we were unable to understand the context of some words, which hindered our ability to correctly identify topics. One example would be the word ‘hike’ found in the cons, which after researching and looking through the datasets in detail, we found that it referred to the lack of ‘salary hike’ in the company. However, a lot of time is required for us to do this research for each word that we are unfamiliar with. Thus, we could only make educated guesses when naming the topics.

On the other hand, both LSA and LDA methods are largely linear models at the core where they assume a linear relationship between terms and topics, which may not always be appropriate (Goyal, 2021). This means that capturing complex non-linear relationships between words and topics can be challenging. Hence to overcome this we could try alternative models such as topic modelling using Neural Networks.

On a good note, preprocessing of data for topic modelling went well, specifically in terms of using NLTK’s dictionary of stopwords. As most reviews used common words in the English Language and there were not any specific domain specific words that we had to account for.

# 7. Future Work and Conclusion

## 7.1. Task 1: Sentiment Analysis

Moving forward, we should consider using ML models that are trained to distinguish between different meanings of words to avoid semantic ambiguity. An example of this is Word2Vec which allows us to capture the meaning of words in a way that is amenable to mathematical manipulation. We could also use models that incorporate context and background knowledge to better interpret the meaning of the words. This would help in detecting sarcasm. One model for this is DeepMoji which recognizes sarcasm and other forms of irony by taking into account the context in which a word or phrase is used. Lastly, we should consider looking at tweets with a higher number of engagements as these tweets would have a higher influence on overall sentiment. One way to do this would be to retrieve tweets from influential users as their tweets mostly result in higher engagement.

## 7.2. Task 2: Topic Modelling

There are several potential uses for topic modelling on the Glassdoor dataset that can be performed in the future.

Firstly, we could explore the likes and dislikes of current and former employees by analysing the reviews based on their employment status stated in their reviews. Secondly, we can examine the branch information of companies to perform a more targeted analysis of employee satisfaction in branches of particular countries or regions. Thirdly, we can utilise job titles and department information provided to identify common themes and issues specific to certain roles or departments within the company.

Finally, we could conduct a time series analysis. By analysing reviews of the companies for each year separately, we can identify how topics of employee satisfaction or dissatisfaction

have evolved over time. This analysis could help to identify trends in employee satisfaction and whether companies have taken steps to address the cons identified by the employees or maintain the pros in the previous years.

After performing these additional Topic Modelling techniques, we can use Natural Language Processing techniques to automatically label the topic based on the most representative words in each topic. By assigning labels automatically, it saves time and effort to label it manually, allowing us to quickly graph the essence of each topic and provides us with an easier comparison across multiple analyses.

### 7.3. Conclusion

In conclusion, our project offers valuable insights to companies in the Technology industry to help them identify changes in their branding and improve their retention strategies. The change in sentiment over time could be a signal to the companies that there are some things that should be changed. The topics identified by our topic models can serve as a starting point for companies to work on improving their branding, and ultimately, employee satisfaction and retention. By utilising the techniques we have demonstrated, companies can gain a deeper understanding of their employees' experiences and take proactive steps to improve their workplace culture.

# 8. Project Experiences/Reflections

| Name | Learning |
|---|---|
| ILLAKKIYA DO RAVINDRAN | Through this project, I got to explore more things outside of the classroom. Even though the NB classifier was taught to us in class, when it came to our project the context was different and I had to explore how the NB classifier could be utilised in our context. In addition, another new thing which I learnt was replacing emojis and emoticons with words, when pre-processing the tweets. It was cool seeing how the library utilised could perceive the meaning of the emoji and replace it with the appropriate words. |
| HO SHENG YUN, JOSLYN | Through this project, I learnt that pre-processing steps should be altered based on the business question or use case the analysis is trying to answer. For example, when deciding which POS tags to keep, our group thought that it may be irrelevant to keep adjectives as it may not help us answer the question of what aspects of the company employees like/dislike. This project also made me think about how applicable some pre-processing techniques are. One example would be the decision not to use techniques such as stemming and lemmatization as it would convert multiple different words into the same base word. This would have caused us to lose some important information when engaging in topic modelling. I also felt that the project challenged me to think about what other methods could be applied to improve the coherence or interpretability of the topics. |
| SANOFER | I learnt about the differences between LDA and LSA. LDA,is a |

| | |
|---|---|
| | generative probabilistic model that assumes that each document in a corpus is a mixture of topics, and each topic is a probability distribution over a set of words whereas LSA is a mathematical technique that uses linear algebra to analyse the relationships between words and documents in a corpus. I also learnt that for the task of topic modelling usually human evaluation is better than using metrics like coherence scores as they are a better gauge of the actual usability of the topics, however we should continue to learn more on how we can standardise human evaluations so that they are comparable among different tasks |
| SNEHA MURALI | I learnt about the challenges faced in sentiment analysis as a whole, from tweets extraction all the way to testing the model. Pre-processing steps have to be done differently for tweets - for example, no stopword removal. There were also issues with tweets extractions due to lexical ambiguity. I also had the chance to understand the workings of the VADER model, appreciate its ease of implementation but also consider how it did not take into account misspellings/sarcasm well. |
| SUMER SINGH GREWAL | I learnt about the various packages that can be used to retrieve tweets (Tweepy, Snscrape etc.) and how I can manipulate my code to retrieve relevant tweets using keywords, date range etc. I had the chance to experience how to conduct pre-processing for tweets and learnt more about how helpful sentiment analysis can be when working with social media platforms. |
| JENNIFER WANG XUE'ER | I learnt about how data preprocessing steps is very dependent on what insights the analysis aims to get and how different types of visualisation graphs can be used to better identify the potential insights presented by the final result. For LDA, I used word cloud to better understand the top words gathered to formulate topics. For VADAR and naive bayes, I used time series graphs to show how the sentiments of the 5 companies changes over time according to their company activities. |

# References


1. Collins, L. (2022, August 12). *Job unhappiness is at a staggering all-time high, according to Gallup*. CNBC. https://www.cnbc.com/2022/08/12/job-unhappiness-is-at-a-staggering-all-time-high-according-to-gallup.html
2. Adeline, M. S. (2022, May 1). *Why Most People Hate 9 – 5 Jobs*. https://www.linkedin.com/pulse/why-most-people-hate-9-5-jobs-mih-sih-adeline/
3. Ivanova, I. (2023, February 18). *Tech layoffs: Paypal, Google, Microsoft and more are cutting jobs*. CBS News. https://www.cbsnews.com/news/tech-layoffs-sector-google-recession-2023-02-07/
4. Kelly, J. (2022, March 17). *Tech Talent At Top Companies Are Unhappy About Their Compensation And Open To Pursuing New Opportunities*. Forbes. https://www.forbes.com/sites/jackkelly/2022/03/17/tech-talent-at-top-companies-are-unhappy-about-their-compensation-and-open-to-pursuing-new-opportunities/?sh=51d846cc4b80
5. Goyal, C. (2021, June 26). *Part 16 : Step by step guide to master NLP - topic modelling using LSA*. Analytics Vidhya. Retrieved April 12, 2023, from https://www.analyticsvidhya.com/blog/2021/06/part-16-step-by-step-guide-to-master-nlp-topic-modelling-using-lsa/

# Appendix A: Sentiment Analysis Model Accuracy Evaluation Code

### loading the goldtruth for 100 microsoft dec tweets

In [15]:
```
micro_goldtruth = pd.read_excel(r'microsoft_goldtruth.xlsx')
micro_goldtruth.head(10)
```

Out[15]:

| | Datetime | Text | goldtruth |
|---|---|---|---|
| 0 | 2022-12-01 | SOCIALIZED MEDIA: HUNDREDS OF ISRAEL,ÄôS UNIT ... | negative |
| 1 | 2022-12-01 | RT HuffPostWomen: The Microsoft mogul gave $5 ... | neutral |
| 2 | 2022-12-01 | The Microsoft mogul gave $5 billion to the Bil... | neutral |
| 3 | 2022-12-01 | Microsoft SQL/C Developer at RBC\nCome Work wi... | positive |
| 4 | 2022-12-01 | Large NYC taxi fleet looking to hire a SECRETA... | neutral |
| 5 | 2022-12-01 | Large NYC taxi fleet looking to hire a SECRETA... | neutral |
| 6 | 2022-12-01 | @TolandsJournal @destinytrack Yes but Microsof... | negative |
| 7 | 2022-12-01 | Interesting read on hybrid work - we need to l... | positive |
| 8 | 2022-12-01 | @Z_is_real_2402 i will now work offline. you n... | neutral |
| 9 | 2022-12-01 | what the heck microsoft. RD client 8 is no mor... | negative |

In [16]:
```
goldtruthvalues = list(micro_goldtruth['goldtruth'])
#goldtruthvalues is the true labels of the 100 microsoft dec tweets
```

### count number of correct predictions for microsoft dec 22

In [17]:
```
microsoft_accuracycount = 0
for i in range(len(dec_predictedlabels)):
    if dec_predictedlabels[i] == goldtruthvalues[i]:
        microsoft_accuracycount += 1

print(microsoft_accuracycount)

# 54 correct values out of 100, for microsoft
```

```
54
```

*Figure 1: Number of tweets correctly labelled by VADER Model - Microsoft*

### loading oracle goldtruth for first 100 tweets of dec 2022

In [26]:
```
oracle_goldtruth = pd.read_excel('oracle_goldtruth.xlsx')
oracle_goldtruth.head(10)
```

Out[26]:

| | Datetime | Text | goldtruth |
|---|---|---|---|
| 0 | 2022-12-01 | â€ Drywall's "Work The Dumb Oracle" album's wa... | positive |
| 1 | 2022-12-01 | @ORACLE_ECHO @snakeeyes828 @CPD1617Scanner Wha... | neutral |
| 2 | 2022-12-01 | My copy of FANTASTIC FRIGHTS arrived, marking ... | positive |
| 3 | 2022-12-01 | Look at any university in the US, they have cl... | positive |
| 4 | 2022-12-01 | @MsJoyceTarot Ms Joyce, Iâ€™ve followed your w... | positive |
| 5 | 2022-12-01 | @lightworker4441 Another beautiful addition to... | positive |
| 6 | 2022-12-01 | @Fact_Oracle @AJemaineClement you really had t... | positive |
| 7 | 2022-12-01 | @benmudowaya @GonyeMukototsi @Yvonne_Maphosa C... | negative |
| 8 | 2022-12-01 | @Henryjeromekni2 @SimbaGolden1 @JoJoFromJerz S... | negative |
| 9 | 2022-12-01 | I've managed to do all this with 2 kids, one a... | positive |

In [27]:
```
oracle_goldtruthvalues = list(oracle_goldtruth['goldtruth'])
```

### count number of correct predictions for oracle dec 22

In [28]:
```
oracle_accuracycount = 0
for i in range(len(oracle_dec_predictedlabels)):
    if oracle_dec_predictedlabels[i] == oracle_goldtruthvalues[i]:
        oracle_accuracycount += 1

print(oracle_accuracycount)

# 46 correct values out of 100, for microsoft
```

```
46
```

*Figure 2: Number of tweets correctly labelled by VADER Model - Oracle*

### overall accuracy score for vader

In [30]:
```
#Get accuracy of VADER model
totalaccuracy = (oracle_accuracycount + microsoft_accuracycount)/200
print(totalaccuracy)
```

```
0.5
```

*Figure 3: Overall Accuracy Score of VADER Model*

In [19]:
```
## for the 'positive' class, count number of TP, TN, FP and Fn

tp_microsoft_pos = 0
for i in range(len(dec_predictedlabels)):
    if dec_predictedlabels[i] == 'positive' and goldtruthvalues[i] == 'positive':
        tp_microsoft_pos += 1


tn_microsoft_pos = 0
for i in range(len(dec_predictedlabels)):
    if dec_predictedlabels[i] != 'positive' and goldtruthvalues[i] != 'positive':
        tn_microsoft_pos += 1

fp_microsoft_pos = 0
for i in range(len(dec_predictedlabels)):
    if dec_predictedlabels[i] == 'positive' and goldtruthvalues[i] != 'positive':
        fp_microsoft_pos += 1

fn_microsoft_pos = 0
for i in range(len(dec_predictedlabels)):
    if dec_predictedlabels[i] != 'positive' and goldtruthvalues[i] == 'positive':
        fn_microsoft_pos += 1

print(tp_microsoft_pos)
print(tn_microsoft_pos)
print(fp_microsoft_pos)
print(fn_microsoft_pos)
```

```
26
39
30
5
```

*Figure 4: TP/TN/FP/FN Calculation for 'Positive' Tweets - Microsoft*

In [20]:
```
## for the 'negative' class, count number of TP, TN, FP and Fn

tp_microsoft_neg = 0
for i in range(len(dec_predictedlabels)):
    if dec_predictedlabels[i] == 'negative' and goldtruthvalues[i] == 'negative':
        tp_microsoft_neg += 1


tn_microsoft_neg = 0
for i in range(len(dec_predictedlabels)):
    if dec_predictedlabels[i] != 'negative' and goldtruthvalues[i] != 'negative':
        tn_microsoft_neg += 1

fp_microsoft_neg = 0
for i in range(len(dec_predictedlabels)):
    if dec_predictedlabels[i] == 'negative' and goldtruthvalues[i] != 'negative':
        fp_microsoft_neg += 1

fn_microsoft_neg = 0
for i in range(len(dec_predictedlabels)):
    if dec_predictedlabels[i] != 'negative' and goldtruthvalues[i] == 'negative':
        fn_microsoft_neg += 1

print(tp_microsoft_neg)
print(tn_microsoft_neg)
print(fp_microsoft_neg)
print(fn_microsoft_neg)
```

```
14
62
5
19
```

*Figure 5: TP/TN/FP/FN Calculation for 'Negative' Tweets - Microsoft*

In [21]:
```
## for the 'neutral' class, count number of TP, TN, FP and Fn

tp_microsoft_neu = 0
for i in range(len(dec_predictedlabels)):
    if dec_predictedlabels[i] == 'neutral' and goldtruthvalues[i] == 'neutral':
        tp_microsoft_neu += 1


tn_microsoft_neu = 0
for i in range(len(dec_predictedlabels)):
    if dec_predictedlabels[i] != 'neutral' and goldtruthvalues[i] != 'neutral':
        tn_microsoft_neu += 1

fp_microsoft_neu = 0
for i in range(len(dec_predictedlabels)):
    if dec_predictedlabels[i] == 'neutral' and goldtruthvalues[i] != 'neutral':
        fp_microsoft_neu += 1

fn_microsoft_neu = 0
for i in range(len(dec_predictedlabels)):
    if dec_predictedlabels[i] != 'neutral' and goldtruthvalues[i] == 'neutral':
        fn_microsoft_neu += 1

print(tp_microsoft_neu)
print(tn_microsoft_neu)
print(fp_microsoft_neu)
print(fn_microsoft_neu)
```

```
14
53
11
22
```

*Figure 6: TP/TN/FP/FN Calculation for 'Neutral' Tweets - Microsoft*

In [36]:
```
## for the 'positive' class, count number of TP, TN, FP and Fn

tp_oracle_pos = 0
for i in range(len(oracle_dec_predictedlabels)):
    if oracle_dec_predictedlabels[i] == 'positive' and oracle_goldtruthvalues[i] == 'positive':
        tp_oracle_pos += 1


tn_oracle_pos = 0
for i in range(len(oracle_dec_predictedlabels)):
    if oracle_dec_predictedlabels[i] != 'positive' and oracle_goldtruthvalues[i] != 'positive':
        tn_oracle_pos += 1

fp_oracle_pos = 0
for i in range(len(oracle_dec_predictedlabels)):
    if oracle_dec_predictedlabels[i] == 'positive' and oracle_goldtruthvalues[i] != 'positive':
        fp_oracle_pos += 1

fn_oracle_pos = 0
for i in range(len(oracle_dec_predictedlabels)):
    if oracle_dec_predictedlabels[i] != 'positive' and oracle_goldtruthvalues[i] == 'positive':
        fn_oracle_pos += 1

print(tp_oracle_pos)
print(tn_oracle_pos)
print(fp_oracle_pos)
print(fn_oracle_pos)
```

```
24
31
40
5
```

*Figure 7: TP/TN/FP/FN Calculation for 'Positive' Tweets - Oracle*

In [37]:
```
## for the 'negative' class, count number of TP, TN, FP and Fn

tp_oracle_neg = 0
for i in range(len(oracle_dec_predictedlabels)):
    if oracle_dec_predictedlabels[i] == 'negative' and oracle_goldtruthvalues[i] == 'negative':
        tp_oracle_neg += 1


tn_oracle_neg = 0
for i in range(len(oracle_dec_predictedlabels)):
    if oracle_dec_predictedlabels[i] != 'negative' and oracle_goldtruthvalues[i] != 'negative':
        tn_oracle_neg += 1

fp_oracle_neg = 0
for i in range(len(oracle_dec_predictedlabels)):
    if oracle_dec_predictedlabels[i] == 'negative' and oracle_goldtruthvalues[i] != 'negative':
        fp_oracle_neg += 1

fn_oracle_neg = 0
for i in range(len(oracle_dec_predictedlabels)):
    if oracle_dec_predictedlabels[i] != 'negative' and oracle_goldtruthvalues[i] == 'negative':
        fn_oracle_neg += 1

print(tp_oracle_neg)
print(tn_oracle_neg)
print(fp_oracle_neg)
print(fn_oracle_neg)
```

```
12
66
6
16
```

*Figure 8: TP/TN/FP/FN Calculation for 'Negative' Tweets - Oracle*

In [38]:
```
## for the 'neutral' class, count number of TP, TN, FP and Fn

tp_oracle_neu = 0
for i in range(len(oracle_dec_predictedlabels)):
    if oracle_dec_predictedlabels[i] == 'neutral' and oracle_goldtruthvalues[i] == 'neutral':
        tp_oracle_neu += 1


tn_oracle_neu = 0
for i in range(len(oracle_dec_predictedlabels)):
    if oracle_dec_predictedlabels[i] != 'neutral' and oracle_goldtruthvalues[i] != 'neutral':
        tn_oracle_neu += 1

fp_oracle_neu = 0
for i in range(len(oracle_dec_predictedlabels)):
    if oracle_dec_predictedlabels[i] == 'neutral' and oracle_goldtruthvalues[i] != 'neutral':
        fp_oracle_neu += 1

fn_oracle_neu = 0
for i in range(len(oracle_dec_predictedlabels)):
    if oracle_dec_predictedlabels[i] != 'neutral' and oracle_goldtruthvalues[i] == 'neutral':
        fn_oracle_neu += 1

print(tp_oracle_neu)
print(tn_oracle_neu)
print(fp_oracle_neu)
print(fn_oracle_neu)
```

```
10
49
8
33
```

*Figure 9: TP/TN/FP/FN Calculation for 'Neutral' Tweets - Oracle*

In [39]:
```
# adding up total labels across microsoft/oracle
# negative : 28 + 33 =61
# neutral : 43 + 36 = 79
# positive: 29+ 31 = 60


# positive class
tp_pos = tp_oracle_pos + tp_microsoft_neg
tn_pos = tn_oracle_pos + tn_microsoft_neg
fp_pos = fp_oracle_pos + fp_microsoft_neg
fn_pos = fn_oracle_pos + fn_microsoft_neg

#negative class

tp_neg = tp_oracle_neg + tp_microsoft_neg
tn_neg = tn_oracle_neg + tn_microsoft_neg
fp_neg = fp_oracle_neg + fp_microsoft_neg
fn_neg = fn_oracle_neg + fn_microsoft_neg

#neutral class

tp_neu = tp_oracle_neu + tp_microsoft_neu
tn_neu = tn_oracle_neu + tn_microsoft_neu
fp_neu = fp_oracle_neu + fp_microsoft_neu
fn_neu = fn_oracle_neu + fn_microsoft_neu
```

In [40]:
```
# positive class

precision_pos = tp_pos / (tp_pos + fp_pos)
recall_pos = tp_pos / (tp_pos + fn_pos)

# negative class

precision_neg = tp_neg / (tp_neg + fp_neg)
recall_neg = tp_neg / (tp_neg + fn_neg)

# neutral class

precision_neu = tp_neu / (tp_neu + fp_neu)
recall_neu = tp_neu / (tp_neu + fn_neu)
```

In [41]:
```
#weights for each class since we did not have a labelled dataset

w_Positive = 200 / (3 * 60)

w_Negative = 200 / (3 * 61)

w_Neutral = 200 / (3 * 79)
```

In [42]:
```
Macroaveraged_Precision = (w_Positive*precision_pos + w_Negative*precision_neg + w_Neutral * precision_neu) / 3

Macroaveraged_Recall = (w_Positive*recall_pos + w_Negative*recall_neg + w_Neutral*recall_neu) / 3

f1 = 2 * ( ( (Macroaveraged_Precision) * (Macroaveraged_Recall) ) / ( (Macroaveraged_Precision) + (Macroaveraged_Recall
```

In [43]:
```
print('Macroaveraged_Precision:',Macroaveraged_Precision)
print('Macroaveraged_Recall:',Macroaveraged_Recall)
print('F1 Score:', f1)
```

```
Macroaveraged_Precision: 0.5825621317709787
Macroaveraged_Recall: 0.46773245165286825
F1 Score: 0.5188700740416946
```

*Figure 10: Evaluation Metrics Calculation - VADER*

## Calculate Accuracy of Naive Bayes Classification Model

In [14]:
```
#Load dataset
df = pd.read_csv('Dataset/Comparing Accuracy/Microsoft_NBClassifier_CompareAccuracy.csv', encoding='latin-1')

#Calculate Accuracy
microsoft_accuracycount = 0
for i in range(len(df)):
    if df['goldtruth'][i] == df['predicted sentiment'][i]:
        microsoft_accuracycount += 1

print(microsoft_accuracycount)
```

```
37
```

*Figure 11: Number of tweets correctly labelled by Naive Bayes Model - Microsoft*

## Calculate Accuracy of Naive Bayes Classification Model

In [14]:
```
#Load dataset
df = pd.read_csv('Dataset/Comparing Accuracy/Oracle_NBClassifier_CompareAccuracy.csv', encoding='latin-1')

#Calculate Accuracy
oracle_accuracycount = 0
for i in range(len(df)):
    if df['goldtruth'][i] == df['predicted sentiment'][i]:
        oracle_accuracycount += 1

print(oracle_accuracycount)
```

```
49
```

*Figure 12: Number of tweets correctly labelled by Naive Bayes Model - Oracle*

## Calculate Overall Accuracy of Naive Bayes Model

In [32]:
```
overall_accuracy = (oracle_accuracycount + microsoft_accuracycount) / 200
print(overall_accuracy)
```

```
0.43
```

*Figure 13: Overall Accuracy Score of Naive Bayes Model*

In [15]:
```
## for the 'positive' class, count number of TP, TN, FP and Fn

tp_microsoft_pos = 0
for i in range(len(df)):
    if df['predicted sentiment'][i] == 'pos' and df['goldtruth'][i] == 'pos':
        tp_microsoft_pos += 1


tn_microsoft_pos = 0
for i in range(len(df)):
    if df['predicted sentiment'][i] != 'pos' and df['goldtruth'][i] != 'pos':
        tn_microsoft_pos += 1

fp_microsoft_pos = 0
for i in range(len(df)):
    if df['predicted sentiment'][i] == 'pos' and df['goldtruth'][i] != 'pos':
        fp_microsoft_pos += 1

fn_microsoft_pos = 0
for i in range(len(df)):
    if df['predicted sentiment'][i] != 'pos' and df['goldtruth'][i] == 'pos':
        fn_microsoft_pos += 1

print(tp_microsoft_pos)
print(tn_microsoft_pos)
print(fp_microsoft_pos)
print(fn_microsoft_pos)
```

```
9
55
14
22
```

*Figure 14: TP/TN/FP/FN Calculation for 'Positive' Tweets - Microsoft*

In [16]:
```
## for the 'negative' class, count number of TP, TN, FP and Fn

tp_microsoft_neg = 0
for i in range(len(df)):
    if df['predicted sentiment'][i] == 'neg' and df['goldtruth'][i] == 'neg':
        tp_microsoft_neg += 1


tn_microsoft_neg = 0
for i in range(len(df)):
    if df['predicted sentiment'][i] != 'neg' and df['goldtruth'][i] != 'neg':
        tn_microsoft_neg += 1

fp_microsoft_neg = 0
for i in range(len(df)):
    if df['predicted sentiment'][i] == 'neg' and df['goldtruth'][i] != 'neg':
        fp_microsoft_neg += 1

fn_microsoft_neg = 0
for i in range(len(df)):
    if df['predicted sentiment'][i] != 'neg' and df['goldtruth'][i] == 'neg':
        fn_microsoft_neg += 1

print(tp_microsoft_neg)
print(tn_microsoft_neg)
print(fp_microsoft_neg)
print(fn_microsoft_neg)
```

```
26
21
46
7
```

*Figure 15: TP/TN/FP/FN Calculation for 'Negative' Tweets - Microsoft*

In [17]:
```
## for the 'neutral' class, count number of TP, TN, FP and Fn

tp_microsoft_neu = 0
for i in range(len(df)):
    if df['predicted sentiment'][i] == 'neu' and df['goldtruth'][i] == 'neu':
        tp_microsoft_neu += 1


tn_microsoft_neu = 0
for i in range(len(df)):
    if df['predicted sentiment'][i] != 'neu' and df['goldtruth'][i] != 'neu':
        tn_microsoft_neu += 1

fp_microsoft_neu = 0
for i in range(len(df)):
    if df['predicted sentiment'][i] == 'neu' and df['goldtruth'][i] != 'neu':
        fp_microsoft_neu += 1

fn_microsoft_neu = 0
for i in range(len(df)):
    if df['predicted sentiment'][i] != 'neu' and df['goldtruth'][i] == 'neu':
        fn_microsoft_neu += 1

print(tp_microsoft_neu)
print(tn_microsoft_neu)
print(fp_microsoft_neu)
print(fn_microsoft_neu)
```

```
2
61
3
34
```

*Figure 16: TP/TN/FP/FN Calculation for 'Neutral' Tweets - Microsoft*

In [32]:
```
## for the 'positive' class, count number of TP, TN, FP and Fn

tp_oracle_pos = 0
for i in range(len(df)):
    if df['predicted sentiment'][i] == 'pos' and df['goldtruth'][i] == 'pos':
        tp_oracle_pos += 1


tn_oracle_pos = 0
for i in range(len(df)):
    if df['predicted sentiment'][i] != 'pos' and df['goldtruth'][i] != 'pos':
        tn_oracle_pos += 1

fp_oracle_pos = 0
for i in range(len(df)):
    if df['predicted sentiment'][i] == 'pos' and df['goldtruth'][i] != 'pos':
        fp_oracle_pos += 1

fn_oracle_pos = 0
for i in range(len(df)):
    if df['predicted sentiment'][i] != 'pos' and df['goldtruth'][i] == 'pos':
        fn_oracle_pos += 1

print(tp_oracle_pos)
print(tn_oracle_pos)
print(fp_oracle_pos)
print(fn_oracle_pos)
```

```
18
45
26
11
```

*Figure 17: TP/TN/FP/FN Calculation for 'Positive' Tweets - Oracle*

In [33]:
```
## for the 'negative' class, count number of TP, TN, FP and Fn

tp_oracle_neg = 0
for i in range(len(df)):
    if df['predicted sentiment'][i] == 'neg' and df['goldtruth'][i] == 'neg':
        tp_oracle_neg += 1


tn_oracle_neg = 0
for i in range(len(df)):
    if df['predicted sentiment'][i] != 'neg' and df['goldtruth'][i] != 'neg':
        tn_oracle_neg += 1

fp_oracle_neg = 0
for i in range(len(df)):
    if df['predicted sentiment'][i] == 'neg' and df['goldtruth'][i] != 'neg':
        fp_oracle_neg += 1

fn_oracle_neg = 0
for i in range(len(df)):
    if df['predicted sentiment'][i] != 'neg' and df['goldtruth'][i] == 'neg':
        fn_oracle_neg += 1

print(tp_oracle_neg)
print(tn_oracle_neg)
print(fp_oracle_neg)
print(fn_oracle_neg)
```

```
24
47
25
4
```

*Figure 18: TP/TN/FP/FN Calculation for 'Negative' Tweets - Oracle*

In [34]:
```
## for the 'neutral' class, count number of TP, TN, FP and Fn

tp_oracle_neu = 0
for i in range(len(df)):
    if df['predicted sentiment'][i] == 'neu' and df['goldtruth'][i] == 'neu':
        tp_oracle_neu += 1


tn_oracle_neu = 0
for i in range(len(df)):
    if df['predicted sentiment'][i] != 'neu' and df['goldtruth'][i] != 'neu':
        tn_oracle_neu += 1

fp_oracle_neu = 0
for i in range(len(df)):
    if df['predicted sentiment'][i] == 'neu' and df['goldtruth'][i] != 'neu':
        fp_oracle_neu += 1

fn_oracle_neu = 0
for i in range(len(df)):
    if df['predicted sentiment'][i] != 'neu' and df['goldtruth'][i] == 'neu':
        fn_oracle_neu += 1

print(tp_oracle_neu)
print(tn_oracle_neu)
print(fp_oracle_neu)
print(fn_oracle_neu)
```

```
7
57
0
36
```

*Figure 19: TP/TN/FP/FN Calculation for 'Neutral' Tweets - Oracle*

## Weighted Precision/Recall/F1 Score Computation - Microsoft & Oracle

In [35]:
```
# positive class
tp_pos = tp_oracle_pos + tp_microsoft_neg
tn_pos = tn_oracle_pos + tn_microsoft_neg
fp_pos = fp_oracle_pos + fp_microsoft_neg
fn_pos = fn_oracle_pos + fn_microsoft_neg

#negative class

tp_neg = tp_oracle_neg + tp_microsoft_neg
tn_neg = tn_oracle_neg + tn_microsoft_neg
fp_neg = fp_oracle_neg + fp_microsoft_neg
fn_neg = fn_oracle_neg + fn_microsoft_neg

#neutral class

tp_neu = tp_oracle_neu + tp_microsoft_neu
tn_neu = tn_oracle_neu + tn_microsoft_neu
fp_neu = fp_oracle_neu + fp_microsoft_neu
fn_neu = fn_oracle_neu + fn_microsoft_neu
```

In [36]:
```
# positive class

precision_pos = tp_pos / (tp_pos + fp_pos)
recall_pos = tp_pos / (tp_pos + fn_pos)

# negative class

precision_neg = tp_neg / (tp_neg + fp_neg)
recall_neg = tp_neg / (tp_neg + fn_neg)

# neutral class

precision_neu = tp_neu / (tp_neu + fp_neu)
recall_neu = tp_neu / (tp_neu + fn_neu)
```

In [37]:
```
#weights for each class since we did not have a labelled dataset

w_Positive = 200 / (3 * 60)

w_Negative = 200 / (3 * 61)

w_Neutral = 200 / (3 * 79)
```

In [38]:
```
averaged_Precision = (w_Positive*precision_pos + w_Negative*precision_neg + w_Neutral * precision_neu) / 3

averaged_Recall = (w_Positive*recall_pos + w_Negative*recall_neg + w_Neutral*recall_neu) / 3

2 * ( ( (Macroaveraged_Precision) * (Macroaveraged_Recall) ) / ( (Macroaveraged_Precision) + (Macroaveraged_Recall) ) )
```

In [39]:
```
print('Macroaveraged_Precision:',Macroaveraged_Precision)
print('Macroaveraged_Recall:',Macroaveraged_Recall)
print('F1 Score:', f1)
```

```
Macroaveraged_Precision: 0.5019924402384506
Macroaveraged_Recall: 0.5934951473585859
F1 Score: 0.5439223240232746
```

*Figure 20: Evaluation Metrics Calculation - Naive Bayes*

# Appendix B: Topic Modelling - Choosing optimal number of topics

```
def optimal_topics(corpus):
    model_list = []
    coherence_values = []
    model_topics = []

    for num_topics in range(2, 10):
        stop_list = nltk.corpus.stopwords.words('english')
        stop_list.extend(["company","ibm","oracle","apple","microsoft","google"])

        fids = corpus.fileids()
        docs1 = []
        for fid in fids:
            doc_raw = corpus.raw(fid)
            doc = nltk.word_tokenize(doc_raw)
            docs1.append(doc)
        nouns = []
        pos_tag = [nltk.pos_tag(doc) for doc in docs1]
        for doc in pos_tag:
            document = []
            for w in doc:
                if w[1] == 'NN' or w[1] == "NNS":
                    document.append(w[0])
            nouns.append(document)
        docs2 = [[w.lower() for w in doc] for doc in nouns]
        docs3 = [[w for w in doc if re.search('^[a-z]+$', w)] for doc in docs2]
        reviews_docs = [[w for w in doc if w not in stop_list] for doc in docs3]

        dictionary = gensim.corpora.Dictionary(reviews_docs)
        vecs1 = [dictionary.doc2bow(doc) for doc in reviews_docs]
        tfidf = gensim.models.TfidfModel(vecs1)
        reviews_vecs = [tfidf[vec] for vec in vecs1]

        reviews_lda_x = gensim.models.ldamodel.LdaModel(corpus=reviews_vecs, id2word=dictionary, num_topics=num_topics)
        coherencemodel = CoherenceModel(model=reviews_lda_x, texts=reviews_docs, dictionary=dictionary, coherence='c_v')
        model_topics.append(num_topics)
        model_list.append(reviews_lda_x)
        coherence_values.append(coherencemodel.get_coherence())
        # print("#Topics: " + str(num_topics) + " Score: " + str(coherencemodel.get_coherence()))

    print(datetime.datetime.now())

    return coherence_values
```

*Figure 1: Function to find optimal Number of topics by using Coherent Scores*

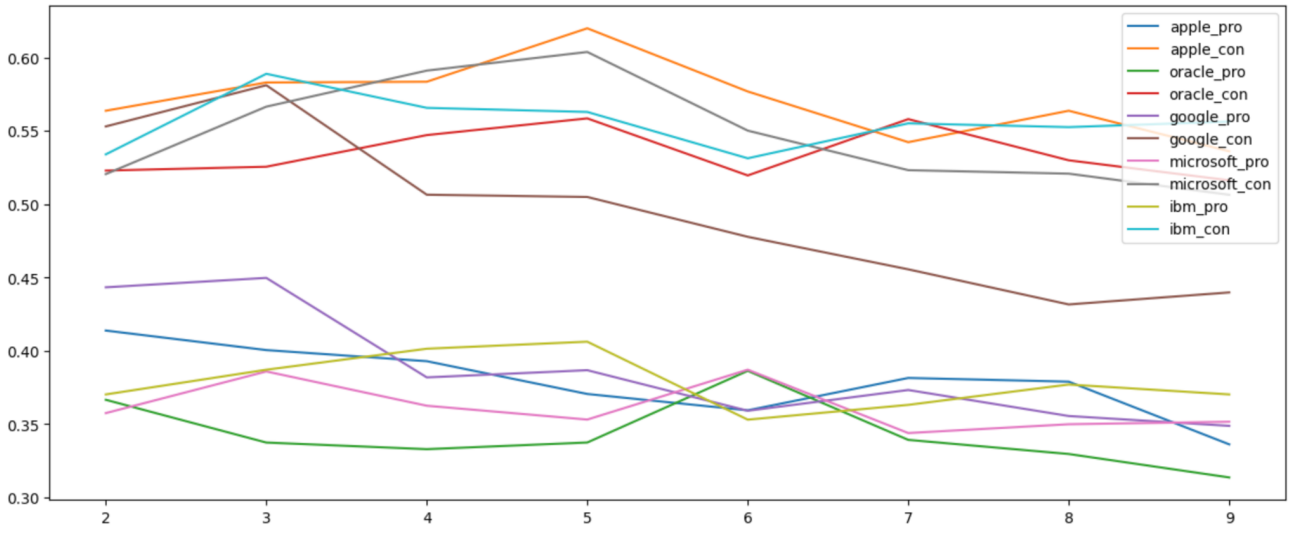


*Figure 2: Plot of Coherence Scores against Number of Topics for each pro and con*

# Appendix C: Topic Modelling - LDA Model Code

```
def topic_modelling(corpus):
    stop_list = nltk.corpus.stopwords.words('english')
    stop_list.extend(["company","ibm","oracle","apple","microsoft","google", 'pros', 'cons'])

    fids = corpus.fileids()
    docs1 = []
    for fid in fids:
        doc_raw = corpus.raw(fid)
        doc = nltk.word_tokenize(doc_raw)
        docs1.append(doc)
    nouns = []
    pos_tag = [nltk.pos_tag(doc) for doc in docs1]
    for doc in pos_tag:
        document = []
        for w in doc:
            if w[1] == 'NN' or w[1] == "NNS":
                document.append(w[0])
        nouns.append(document)
    docs2 = [[w.lower() for w in doc] for doc in nouns]
    docs3 = [[w for w in doc if re.search('^[a-z]+$', w)] for doc in docs2]
    reviews_docs = [[w for w in doc if w not in stop_list] for doc in docs3]


    dictionary = gensim.corpora.Dictionary(reviews_docs)
    vecs1 = [dictionary.doc2bow(doc) for doc in reviews_docs]
    tfidf = gensim.models.TfidfModel(vecs1)
    reviews_vecs = [tfidf[vec] for vec in vecs1]


    reviews_lda = gensim.models.ldamodel.LdaModel(corpus=reviews_vecs, id2word=dictionary, num_topics=3)
    topics = reviews_lda.show_topics(3, 20)

    perplex= reviews_lda.log_perplexity(reviews_vecs, total_docs=None)
    coherence_model_lda = CoherenceModel(model=reviews_lda, texts=reviews_docs, dictionary=dictionary, coherence='c_v')
    coherence_lda = coherence_model_lda.get_coherence()

    topics_split = []
    for topic in range(0,3):
        split = topics[topic][1].split(" + ")
        topics_split.append(split)

    topic_split_df = pd.DataFrame(topics_split).T
    for i in range(3):
        word_prob_df = topic_split_df[i].str.split("*", n = 1, expand = True)
        topic_split_df["probability-" + str(i)]= word_prob_df[0]
        topic_split_df["word-" + str(i)]= word_prob_df[1]
    return topic_split_df, coherence_lda, perplex
```

*Figure 1: Getting topics and their respective coherence and perplexity scores*

```
def topic_modelling_bigram(corpus):
    stop_list = nltk.corpus.stopwords.words('english')
    stop_list.extend(["company","ibm","oracle","apple","microsoft","google"])

    fids = corpus.fileids()
    docs1 = []
    for fid in fids:
        doc_raw = corpus.raw(fid)
        doc = nltk.word_tokenize(doc_raw)
        docs1.append(doc)
    nouns = []
    pos_tag = [nltk.pos_tag(doc) for doc in docs1]
    for doc in pos_tag:
        document = []
        for w in doc:
            if w[1] == 'NN' or w[1] == "NNS":
                document.append(w[0])
        nouns.append(document)
    docs2 = [[w.lower() for w in doc] for doc in nouns]
    docs3 = [[w for w in doc if re.search('^[a-z]+$', w)] for doc in docs2]
    reviews_docs = [[w for w in doc if w not in stop_list] for doc in docs3]

    bigram = gensim.models.Phrases(reviews_docs)
    for idx in range(len(reviews_docs)):
        for token in bigram[reviews_docs[idx]]:
            if '_' in token:
                # Token is a bigram, add to document.
                reviews_docs[idx].append(token)


    dictionary = gensim.corpora.Dictionary(reviews_docs)
    vecs1 = [dictionary.doc2bow(doc) for doc in reviews_docs]
    tfidf = gensim.models.TfidfModel(vecs1)
    reviews_vecs = [tfidf[vec] for vec in vecs1]

    reviews_lda = gensim.models.ldamodel.LdaModel(corpus=reviews_vecs, id2word=dictionary, num_topics=3)
    topics = reviews_lda.show_topics(3, 20)

    perplex= reviews_lda.log_perplexity(reviews_vecs, total_docs=None)
    coherence_model_lda = CoherenceModel(model=reviews_lda, texts=reviews_docs, dictionary=dictionary, coherence='c_v')
    coherence_lda = coherence_model_lda.get_coherence()

    topics_split = []
    for topic in range(0,3):
        split = topics[topic][1].split(" + ")
        topics_split.append(split)

    topic_split_df = pd.DataFrame(topics_split).T
    for i in range(3):
        word_prob_df = topic_split_df[i].str.split("*", n = 1, expand = True)
        topic_split_df["probability-" + str(i)]= word_prob_df[0]
        topic_split_df["word-" + str(i)]= word_prob_df[1]

    return topic_split_df, coherence_lda, perplex
```

*Figure 2: Bigrams for each topic, and their respective coherence and perplexity scores*

# Appendix D: Topic Modelling - LSA Model Code

## LSA model for Pros

```
#alt LSA code ( FINAL)
import gensim
from gensim import corpora, models
from gensim.corpora import Dictionary
from gensim.models.coherencemodel import CoherenceModel
from gensim.matutils import Sparse2Corpus
from prettytable import PrettyTable



# create a dictionary and bag-of-words representation of the reviews
pro_dictionary = corpora.Dictionary(pro_adj_removed_list)
pro_corpus = [pro_dictionary.doc2bow(review) for review in pro_adj_removed_list]

# build the LSA model
pro_lsa_model = models.LsiModel(pro_corpus, num_topics=3, id2word=pro_dictionary)
pro_coherence_model_lsa = CoherenceModel(model=pro_lsa_model, texts=pro_adj_removed_list, dictionary=pro_dictionary, coheren
pro_coherence_lsa = pro_coherence_model_lsa.get_coherence()


#code to display words
# get the top words for each topic
pro_topic_words = []
for i, topic in enumerate(pro_lsa_model.show_topics()):
    pro_topic_words.append([word for word, _ in pro_lsa_model.show_topic(i, topn=20)])

# create a dictionary to store the top words for each topic
pro_word_dict = {}
for i in range(len(pro_topic_words)):
    for j, word in enumerate(pro_topic_words[i]):
        if j in pro_word_dict:
            pro_word_dict[j].append(word)
        else:
            pro_word_dict[j] = [word]

# create a PrettyTable and add the topic words as rows
pro_table = PrettyTable(['Topic ' + str(i+1) for i in range(len(pro_topic_words))])
for j in range(len(pro_topic_words[0])):
    row = []
    for i in range(len(pro_topic_words)):
        row.append(pro_word_dict[j][i])
    pro_table.add_row(row)

print(pro_table)
```

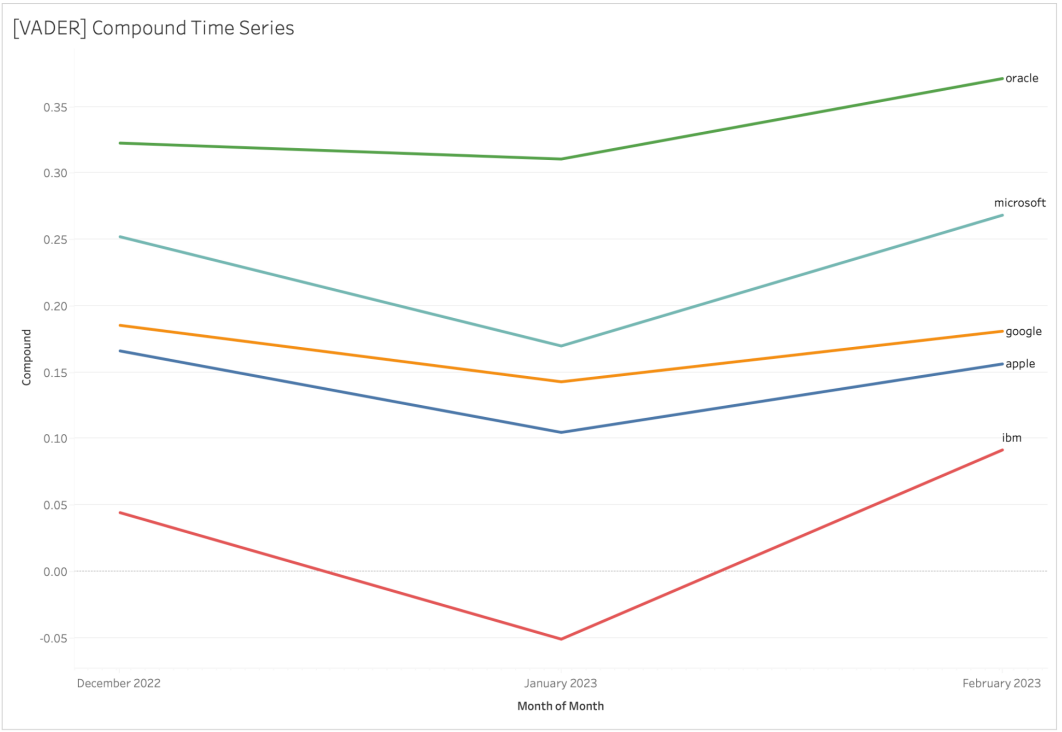
*Figure 1: LSA model code for Pros data with calculation of coherence scores*

### LSA model for Cons

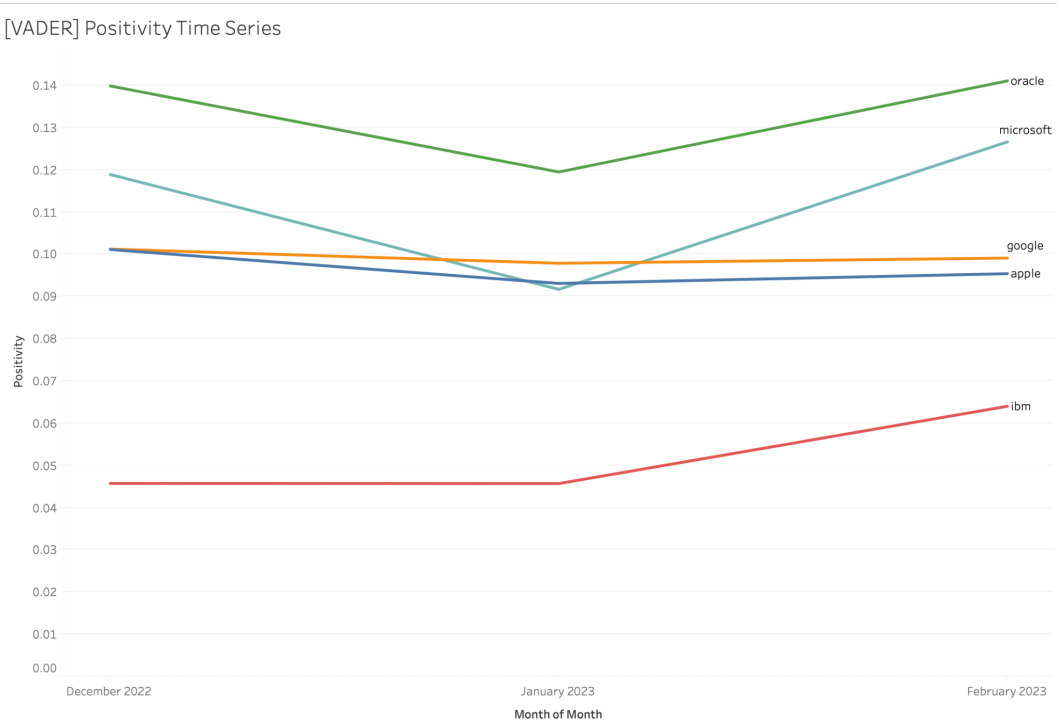
```
#alt LSA code ( FINAL)
import gensim
from gensim import corpora, models
from gensim.corpora import Dictionary
from gensim.models.coherencemodel import CoherenceModel
from gensim.matutils import Sparse2Corpus
from prettytable import PrettyTable



# create a dictionary and bag-of-words representation of the reviews
con_dictionary = corpora.Dictionary(con_adj_removed_list)
con_corpus = [con_dictionary.doc2bow(review) for review in con_adj_removed_list]

# build the LSA model
con_lsa_model = models.LsiModel(con_corpus, num_topics=3, id2word=con_dictionary)
con_coherence_model_lsa = CoherenceModel(model=con_lsa_model, texts=con_adj_removed_list, dictionary=con_dictionary, coheren
con_coherence_lsa = con_coherence_model_lsa.get_coherence()


#code to display words
# get the top words for each topic
con_topic_words = []
for i, topic in enumerate(con_lsa_model.show_topics()):
    con_topic_words.append([word for word, _ in con_lsa_model.show_topic(i, topn=20)])

# create a dictionary to store the top words for each topic
con_word_dict = {}
for i in range(len(con_topic_words)):
    for j, word in enumerate(con_topic_words[i]):
        if j in con_word_dict:
            con_word_dict[j].append(word)
        else:
            con_word_dict[j] = [word]

# create a PrettyTable and add the topic words as rows
con_table = PrettyTable(['Topic ' + str(i+1) for i in range(len(con_topic_words))])
for j in range(len(con_topic_words[0])):
    row = []
    for i in range(len(con_topic_words)):
        row.append(con_word_dict[j][i])
    con_table.add_row(row)

print(con_table)
```

*Figure 2: LSA model code for Cons data with calculation of coherence scores*

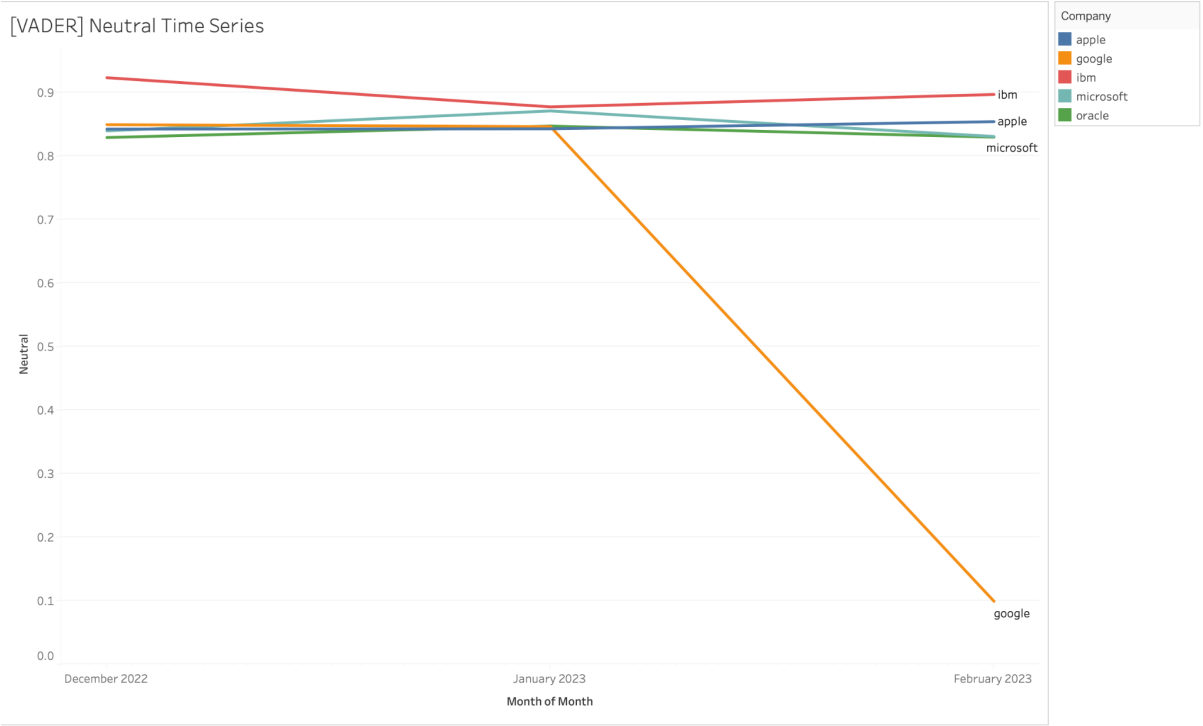
# Appendix E: Sentiment Analysis Time Series



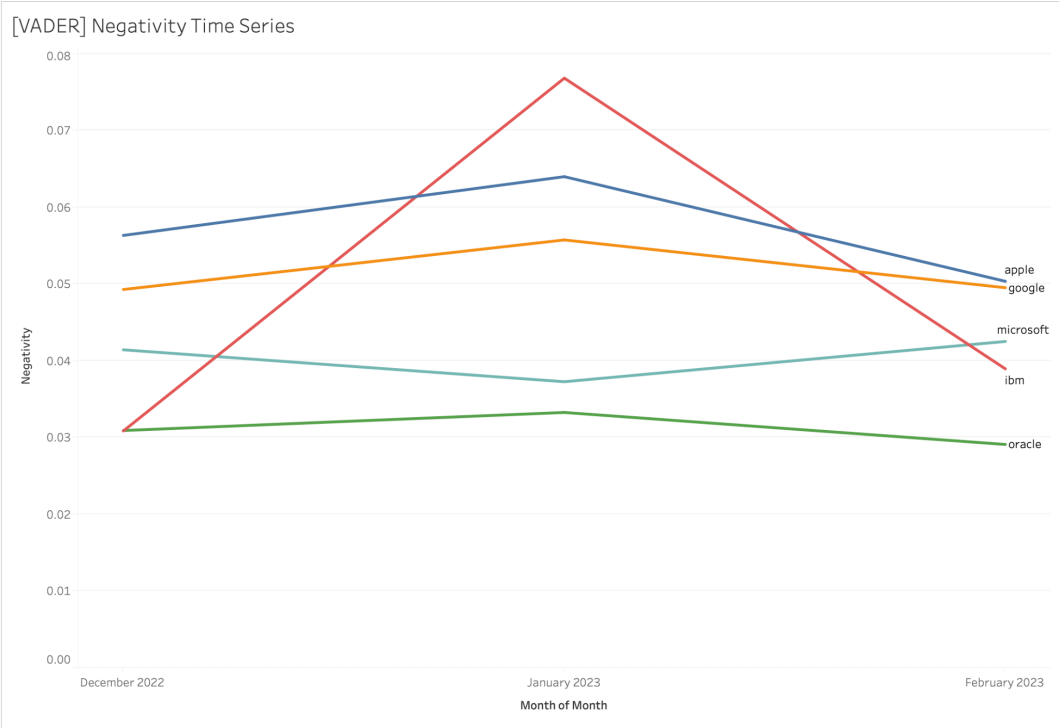

*Time Series 1: VADER Compound Sentiment*



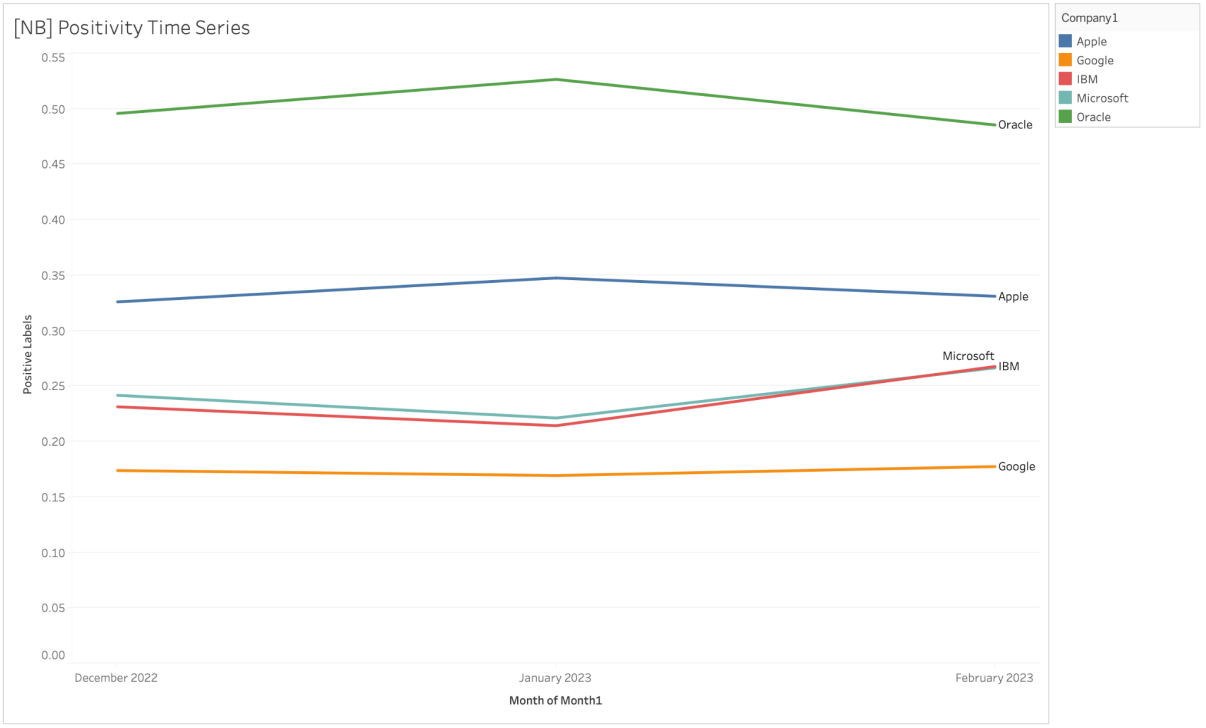

*Time Series 2: VADER Positive Sentiment*



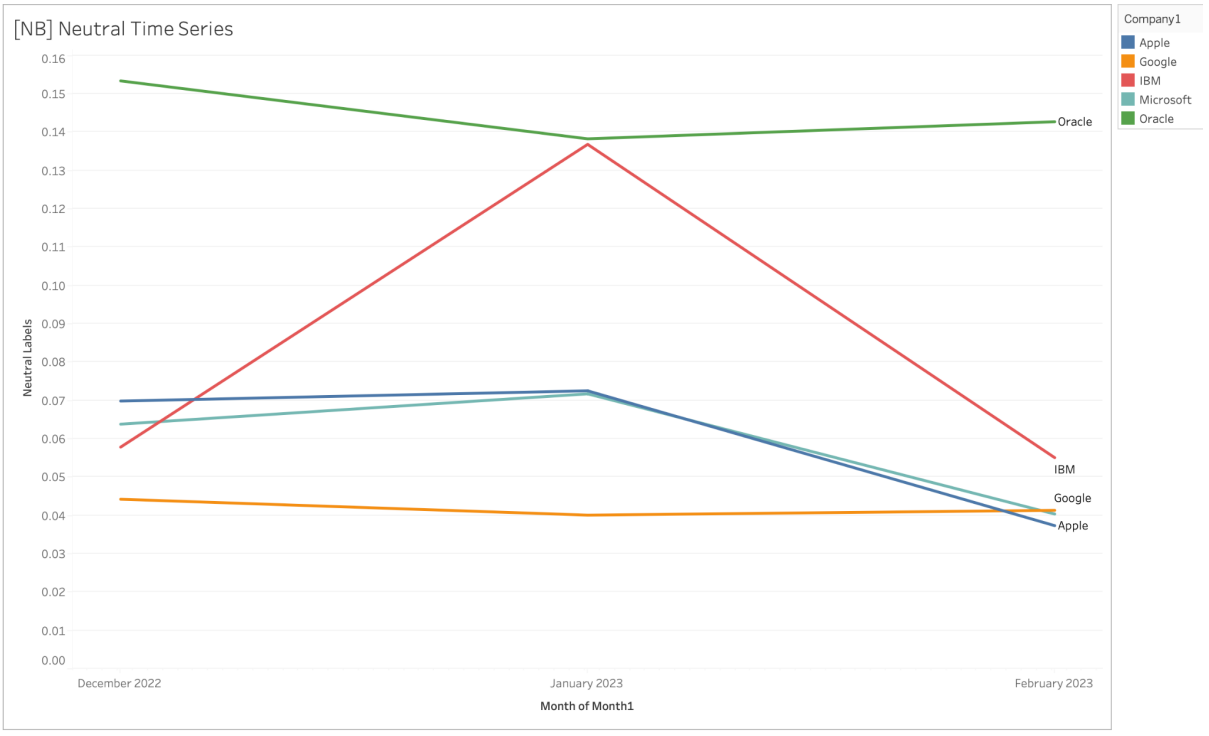

*Time Series 3: VADER Neutral Sentiment*



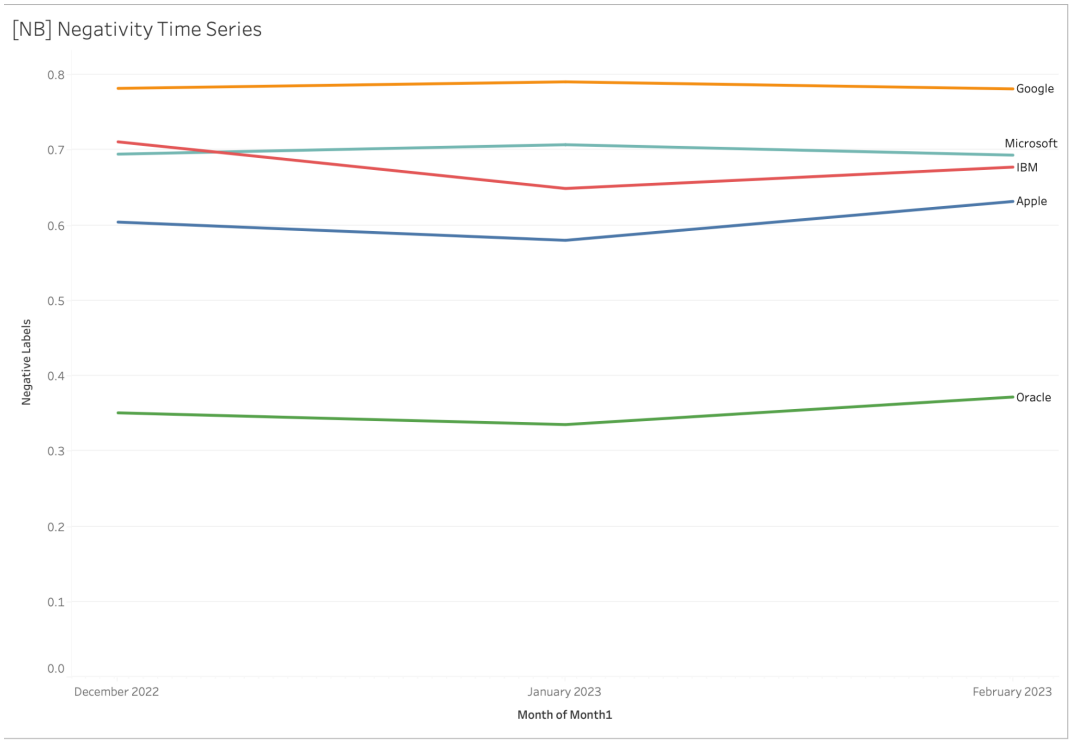

*Time Series 4: VADER Negative Sentiment*




*Time Series 5: Naive Bayes Positive Sentiment*




*Time Series 6: Naive Bayes Neutral Sentiment*




*Time Series 7: Naive Bayes Negative Sentiment*

# Appendix F: Topic Modelling Output

| Topic 1 | Topic 2 | Topic 3 |
|---|---|---|
| benefits | work | products |
| work | products | work |
| people | benefits | discounts |
| employees | people | environment |
| discounts | employees | place |
| products | discounts | life |
| time | customers | team |
| pay | environment | discount |
| opportunities | place | people |
| part | opportunities | managers |
| job | job | home |
| environment | team | hours |
| stock | managers | opportunities |
| employee | things | balance |
| customers | lot | services |
| health | get | friends |
| place | management | fun |
| managers | time | employees |
| lot | everyone | pay |
| team | culture | companies |

*Table 1: LSA Topics for Apple's Pros*

| Topic 1 | Topic 2 | Topic 3 |
|---|---|---|
| employees | hours | customers |
| work | employees | employees |
| customers | managers | work |
| hours | work | hours |
| managers | management | customer |
| people | customers | managers |
| management | people | advisors |
| time | sales | management |
| job | years | things |
| customer | things | sales |
| things | life | life |
| sales | stores | opportunities |
| years | store | hr |
| day | balance | job |
| life | weekends | issues |
| get | products | career |
| team | companies | service |
| products | team | time |
| store | hr | balance |
| manager | customer | expectations |

*Table 2: LSA Topics for Apple's Cons*

| Topic 1 | Topic 2 | Topic 3 |
|---|---|---|
| work | benefits | opportunities |
| life | work | sales |
| balance | sales | products |
| benefits | life | benefits |
| home | opportunities | life |
| opportunities | people | people |
| people | products | balance |
| place | employees | lot |
| culture | balance | software |
| environment | health | lots |
| hours | lot | place |
| team | pay | employees |
| employees | companies | resources |
| lot | resources | product |
| management | job | career |
| sales | lots | companies |
| time | management | job |
| products | career | technology |
| technologies | compensation | management |
| managers | things | team |

*Table 3: LSA Topics for Oracle's Pros*

| Topic 1 | Topic 2 | Topic 3 |
|---|---|---|
| sales | sales | employees |
| managers | employees | managers |
| employees | years | years |
| people | managers | people |
| management | work | work |
| work | people | sales |
| years | management | hikes |
| customers | hikes | management |
| time | companies | things |
| products | hr | manager |
| team | time | team |
| manager | job | months |
| job | promotions | time |
| companies | pay | product |
| things | things | get |
| pay | manager | products |
| customer | employee | companies |
| reps | hours | lot |
| year | get | job |
| lot | team | promotions |

*Table 4: LSA Topics for Oracle's Cons*

| Topic 1 | Topic 2 | Topic 3 |
|---|---|---|
| work | benefits | perks |
| benefits | work | work |
| people | place | employees |
| perks | life | people |
| employees | people | benefits |
| place | balance | place |
| life | environment | life |
| projects | employees | opportunities |
| opportunities | projects | products |
| environment | things | offers |
| things | get | projects |
| culture | lot | food |
| balance | perks | things |
| lot | products | lots |
| food | time | balance |
| products | hours | colleagues |
| lots | colleagues | management |
| colleagues | pay | care |
| time | culture | weeks |
| hours | teams | lot |

*Table 5: LSA Topics for Google's Pros*

| Topic 1 | Topic 2 | Topic 3 |
|---|---|---|
| work | work | things |
| people | managers | managers |
| managers | people | employees |
| things | hours | projects |
| projects | employees | project |
| time | things | people |
| employees | life | work |
| management | projects | manager |
| hours | management | management |
| lot | sales | sales |
| years | balance | hours |
| sales | manager | companies |
| manager | engineers | team |
| engineers | years | folks |
| team | project | hr |
| life | career | years |
| get | team | career |
| job | problems | engineers |
| culture | promotions | culture |
| way | culture | year |

*Table 6: LSA Topics for Google's Cons*

| Topic 1 | Topic 2 | Topic 3 |
|---|---|---|
| work | opportunities | benefits |
| life | work | employees |
| opportunities | life | opportunities |
| balance | benefits | people |
| home | people | life |
| people | career | balance |
| place | employees | projects |
| benefits | lots | work |
| employees | lot | lot |
| environment | projects | years |
| projects | balance | resources |
| culture | skills | companies |
| hours | growth | time |
| lot | job | hours |
| flexibility | things | opportunity |
| opportunity | areas | managers |
| career | resources | pay |
| time | roles | place |
| technologies | home | employee |
| managers | years | management |

*Table 7: LSA Topics for IBM's Pros*

| Topic 1 | Topic 2 | Topic 3 |
|---|---|---|
| employees | employees | managers |
| work | work | work |
| managers | managers | hours |
| years | people | years |
| management | hours | sales |
| people | years | time |
| hours | projects | life |
| time | management | hikes |
| projects | time | balance |
| sales | sales | projects |
| job | things | manager |
| things | manager | line |
| manager | get | jobs |
| benefits | life | skills |
| jobs | project | management |
| employee | job | layoffs |
| pay | team | career |
| year | lot | week |
| business | pay | hr |
| companies | software | benefits |

*Table 8: LSA Topics for IBM's Cons*

| Topic 1 | Topic 2 | Topic 3 |
|---|---|---|
| benefits | work | opportunities |
| work | benefits | work |
| people | life | products |
| opportunities | people | people |
| life | balance | benefits |
| products | products | life |
| balance | opportunities | lots |
| employees | place | career |
| software | software | employees |
| place | culture | software |
| lot | environment | balance |
| lots | lot | lot |
| culture | things | resources |
| things | projects | growth |
| projects | opportunity | things |
| pay | employees | technologies |
| opportunity | teams | teams |
| companies | lots | tons |
| years | get | roles |
| environment | hours | job |

*Table 9: LSA Topics for Microsoft's Pros*

| Topic 1 | Topic 2 | Topic 3 |
|---|---|---|
| managers | managers | employees |
| work | work | work |
| people | people | managers |
| employees | things | years |
| management | hours | life |
| years | life | hours |
| things | time | management |
| time | balance | things |
| teams | employees | sales |
| team | teams | balance |
| review | lot | people |
| hours | products | products |
| lot | management | system |
| manager | years | review |
| sales | team | benefits |
| job | software | teams |
| life | get | ms |
| companies | job | hr |
| culture | projects | companies |
| ms | culture | changes |

*Table 10: LSA Topics for Microsoft's Cons*

| probability-0 | word-0 | probability-1 | word-1 | probability-2 | word-2 |
|---|---|---|---|---|---|
| 0.040 | "culture" | 0.071 | "people" | 0.046 | "pay" |
| 0.033 | "team" | 0.056 | "benefits" | 0.025 | "job" |
| 0.022 | "time" | 0.053 | "environment" | 0.025 | "benefits" |
| 0.020 | "products" | 0.045 | "work" | 0.024 | "discounts" |
| 0.016 | "part" | 0.036 | "place" | 0.018 | "management" |
| 0.015 | "opportunities" | 0.025 | "employees" | 0.015 | "experience" |
| 0.013 | "customers" | 0.019 | "coworkers" | 0.014 | "lots" |
| 0.013 | "staff" | 0.019 | "home" | 0.012 | "products" |
| 0.012 | "everyone" | 0.018 | "lot" | 0.012 | "perks" |
| 0.012 | "family" | 0.018 | "fun" | 0.012 | "technology" |
| 0.012 | "day" | 0.015 | "life" | 0.012 | "world" |
| 0.011 | "colleagues" | 0.014 | "products" | 0.011 | "training" |
| 0.011 | "things" | 0.014 | "balance" | 0.011 | "health" |
| 0.011 | "people" | 0.012 | "hours" | 0.011 | "stock" |
| 0.011 | "benefits" | 0.011 | "managers" | 0.011 | "people" |
| 0.010 | "leadership" | 0.010 | "growth" | 0.011 | "schedule" |
| 0.010 | "customer" | 0.008 | "development" | 0.010 | "opportunity" |
| 0.010 | "resume" | 0.008 | "care" | 0.010 | "options" |
| 0.009 | "something" | 0.007 | "values" | 0.010 | "compensation" |
| 0.008 | "food" | 0.006 | "career" | 0.009 | "salary" |

*Table 11: LDA Topics for Apple's Pros (without bigrams)*

| probability-0 | word-0 | probability-1 | word-1 | probability-2 | word-2 |
|---|---|---|---|---|---|
| 0.024 | "job" | 0.043 | "hours" | 0.024 | "none" |
| 0.023 | "people" | 0.034 | "work" | 0.019 | "times" |
| 0.021 | "management" | 0.029 | "balance" | 0.011 | "customer" |
| 0.019 | "customers" | 0.025 | "life" | 0.010 | "position" |
| 0.018 | "nothing" | 0.018 | "time" | 0.010 | "managers" |
| 0.017 | "lot" | 0.016 | "pay" | 0.010 | "management" |
| 0.012 | "employees" | 0.016 | "schedule" | 0.010 | "experience" |
| 0.012 | "culture" | 0.016 | "growth" | 0.009 | "customers" |
| 0.011 | "time" | 0.015 | "environment" | 0.009 | "everything" |
| 0.011 | "store" | 0.013 | "weekends" | 0.008 | "promotion" |
| 0.009 | "positions" | 0.011 | "room" | 0.008 | "opportunity" |
| 0.008 | "part" | 0.011 | "holidays" | 0.007 | "everyone" |
| 0.008 | "home" | 0.010 | "career" | 0.007 | "communication" |
| 0.007 | "things" | 0.010 | "place" | 0.007 | "metrics" |
| 0.007 | "training" | 0.010 | "advancement" | 0.006 | "team" |
| 0.007 | "mobility" | 0.009 | "politics" | 0.006 | "shift" |
| 0.007 | "progression" | 0.009 | "opportunities" | 0.006 | "employees" |
| 0.006 | "lots" | 0.008 | "days" | 0.006 | "store" |
| 0.006 | "managers" | 0.008 | "day" | 0.006 | "day" |
| 0.006 | "sales" | 0.007 | "changes" | 0.006 | "people" |

*Table 12: LDA Topics for Apple's Cons (without bigrams)*

| probability-0 | word-0 | probability-1 | word-1 | probability-2 | word-2 |
|---|---|---|---|---|---|
| 0.045 | "place" | 0.052 | "environment" | 0.080 | "balance" |
| 0.029 | "people" | 0.039 | "benefits" | 0.071 | "life" |
| 0.024 | "hours" | 0.026 | "team" | 0.068 | "work" |
| 0.023 | "pay" | 0.023 | "work" | 0.045 | "culture" |
| 0.020 | "job" | 0.022 | "opportunity" | 0.021 | "products" |
| 0.018 | "benefits" | 0.020 | "people" | 0.017 | "opportunities" |
| 0.016 | "sales" | 0.020 | "technology" | 0.017 | "technologies" |
| 0.015 | "salary" | 0.020 | "flexibility" | 0.016 | "home" |
| 0.013 | "lot" | 0.017 | "lots" | 0.016 | "product" |
| 0.012 | "working" | 0.016 | "management" | 0.015 | "timings" |
| 0.009 | "companies" | 0.015 | "experience" | 0.014 | "name" |
| 0.009 | "resume" | 0.015 | "opportunities" | 0.012 | "exposure" |
| 0.008 | "teams" | 0.015 | "compensation" | 0.012 | "time" |
| 0.008 | "managers" | 0.014 | "learning" | 0.011 | "people" |
| 0.008 | "security" | 0.014 | "pressure" | 0.011 | "growth" |
| 0.007 | "years" | 0.012 | "projects" | 0.011 | "brand" |
| 0.007 | "package" | 0.011 | "training" | 0.010 | "colleagues" |
| 0.007 | "health" | 0.011 | "employees" | 0.009 | "management" |
| 0.007 | "home" | 0.011 | "things" | 0.008 | "development" |
| 0.007 | "location" | 0.010 | "lot" | 0.007 | "career" |

*Table 13: LDA Topics for Oracle's Pros (without bigrams)*

| probability-0 | word-0 | probability-1 | word-1 | probability-2 | word-2 |
|---|---|---|---|---|---|
| 0.032 | "hike" | 0.020 | "nothing" | 0.035 | "hikes" |
| 0.020 | "work" | 0.017 | "sales" | 0.028 | "growth" |
| 0.015 | "people" | 0.015 | "politics" | 0.021 | "management" |
| 0.014 | "process" | 0.013 | "work" | 0.018 | "career" |
| 0.014 | "team" | 0.013 | "time" | 0.016 | "opportunities" |
| 0.013 | "employees" | 0.012 | "environment" | 0.015 | "salary" |
| 0.013 | "benefits" | 0.011 | "job" | 0.015 | "promotions" |
| 0.013 | "none" | 0.011 | "lot" | 0.015 | "compensation" |
| 0.011 | "increase" | 0.011 | "balance" | 0.015 | "bonus" |
| 0.011 | "place" | 0.010 | "life" | 0.014 | "pay" |
| 0.011 | "years" | 0.009 | "product" | 0.014 | "culture" |
| 0.011 | "market" | 0.009 | "hours" | 0.012 | "raises" |
| 0.010 | "pressure" | 0.008 | "increases" | 0.012 | "bonuses" |
| 0.010 | "employee" | 0.008 | "changes" | 0.011 | "years" |
| 0.010 | "salaries" | 0.008 | "people" | 0.011 | "things" |
| 0.009 | "projects" | 0.007 | "lots" | 0.011 | "promotion" |
| 0.009 | "management" | 0.007 | "management" | 0.010 | "raise" |
| 0.008 | "everything" | 0.007 | "products" | 0.009 | "organization" |
| 0.007 | "way" | 0.007 | "managers" | 0.009 | "companies" |
| 0.007 | "tape" | 0.006 | "year" | 0.009 | "technologies" |

*Table 14: LDA Topics for Oracle's Cons (without bigrams)*

| probability-0 | word-0 | probability-1 | word-1 | probability-2 | word-2 |
|---|---|---|---|---|---|
| 0.073 | "environment" | 0.065 | "people" | 0.038 | "balance" |
| 0.046 | "work" | 0.059 | "place" | 0.032 | "life" |
| 0.030 | "lots" | 0.043 | "culture" | 0.031 | "benefits" |
| 0.025 | "people" | 0.036 | "perks" | 0.027 | "work" |
| 0.019 | "lot" | 0.033 | "food" | 0.022 | "team" |
| 0.018 | "fun" | 0.026 | "benefits" | 0.020 | "pay" |
| 0.018 | "coworkers" | 0.025 | "opportunities" | 0.019 | "time" |
| 0.017 | "benefits" | 0.019 | "job" | 0.019 | "everything" |
| 0.017 | "perks" | 0.018 | "salary" | 0.018 | "colleagues" |
| 0.016 | "food" | 0.016 | "employees" | 0.017 | "experience" |
| 0.015 | "pay" | 0.015 | "world" | 0.015 | "projects" |
| 0.013 | "management" | 0.014 | "compensation" | 0.012 | "things" |
| 0.013 | "culture" | 0.014 | "growth" | 0.011 | "technology" |
| 0.012 | "opportunity" | 0.010 | "office" | 0.011 | "perks" |
| 0.011 | "everyone" | 0.009 | "career" | 0.010 | "impact" |
| 0.009 | "hours" | 0.009 | "flexibility" | 0.010 | "culture" |
| 0.009 | "workplace" | 0.008 | "gym" | 0.009 | "people" |
| 0.009 | "resources" | 0.008 | "products" | 0.008 | "working" |
| 0.009 | "employee" | 0.007 | "peers" | 0.008 | "food" |
| 0.009 | "money" | 0.006 | "projects" | 0.007 | "companies" |

*Table 15: LDA Topics for Google's Pros (without bigrams)*

| probability-0 | word-0 | probability-1 | word-1 | probability-2 | word-2 |
|---|---|---|---|---|---|
| 0.048 | "none" | 0.051 | "nothing" | 0.025 | "time" |
| 0.026 | "balance" | 0.040 | "work" | 0.019 | "things" |
| 0.022 | "people" | 0.030 | "hours" | 0.017 | "politics" |
| 0.022 | "life" | 0.019 | "lot" | 0.017 | "job" |
| 0.017 | "work" | 0.017 | "place" | 0.016 | "management" |
| 0.015 | "growth" | 0.016 | "pressure" | 0.016 | "times" |
| 0.013 | "team" | 0.016 | "projects" | 0.012 | "lots" |
| 0.011 | "project" | 0.014 | "environment" | 0.011 | "culture" |
| 0.010 | "process" | 0.013 | "bit" | 0.010 | "managers" |
| 0.010 | "salary" | 0.013 | "everything" | 0.010 | "people" |
| 0.009 | "bureaucracy" | 0.012 | "impact" | 0.010 | "office" |
| 0.008 | "day" | 0.010 | "pay" | 0.009 | "career" |
| 0.008 | "employees" | 0.010 | "anything" | 0.009 | "everyone" |
| 0.008 | "stress" | 0.009 | "people" | 0.008 | "experience" |
| 0.008 | "competition" | 0.006 | "management" | 0.008 | "opportunities" |
| 0.007 | "promotion" | 0.006 | "thing" | 0.008 | "companies" |
| 0.007 | "benefits" | 0.006 | "leadership" | 0.007 | "organization" |
| 0.007 | "teams" | 0.006 | "con" | 0.007 | "way" |
| 0.006 | "workload" | 0.006 | "culture" | 0.007 | "food" |
| 0.006 | "days" | 0.005 | "sales" | 0.006 | "manager" |

*Table 16: LDA Topics for Google's Cons (without bigrams)*

| probability-0 | word-0 | probability-1 | word-1 | probability-2 | word-2 |
|---|---|---|---|---|---|
| 0.066 | "work" | 0.025 | "lots" | 0.065 | "balance" |
| 0.055 | "environment" | 0.016 | "technology" | 0.056 | "life" |
| 0.045 | "home" | 0.014 | "salary" | 0.051 | "people" |
| 0.040 | "benefits" | 0.013 | "working" | 0.041 | "place" |
| 0.040 | "culture" | 0.013 | "name" | 0.037 | "opportunities" |
| 0.033 | "flexibility" | 0.012 | "things" | 0.029 | "work" |
| 0.022 | "hours" | 0.012 | "resources" | 0.024 | "career" |
| 0.022 | "pay" | 0.011 | "brand" | 0.020 | "projects" |
| 0.017 | "team" | 0.010 | "people" | 0.018 | "growth" |
| 0.015 | "time" | 0.010 | "opportunities" | 0.017 | "lot" |
| 0.014 | "employees" | 0.010 | "resume" | 0.017 | "experience" |
| 0.012 | "training" | 0.009 | "ability" | 0.017 | "opportunity" |
| 0.010 | "management" | 0.009 | "part" | 0.013 | "learning" |
| 0.010 | "people" | 0.009 | "knowledge" | 0.012 | "skills" |
| 0.010 | "schedule" | 0.009 | "products" | 0.011 | "management" |
| 0.010 | "timings" | 0.008 | "clients" | 0.011 | "technologies" |
| 0.010 | "office" | 0.008 | "areas" | 0.011 | "colleagues" |
| 0.010 | "manager" | 0.008 | "job" | 0.011 | "exposure" |
| 0.009 | "employee" | 0.008 | "training" | 0.010 | "world" |
| 0.009 | "options" | 0.007 | "business" | 0.009 | "culture" |

*Table 17: LDA Topics for IBM's Pros (without bigrams)*

| probability-0 | word-0 | probability-1 | word-1 | probability-2 | word-2 |
|---|---|---|---|---|---|
| 0.024 | "pay" | 0.028 | "nothing" | 0.025 | "management" |
| 0.019 | "compensation" | 0.025 | "hike" | 0.021 | "work" |
| 0.017 | "benefits" | 0.018 | "people" | 0.014 | "hours" |
| 0.016 | "career" | 0.018 | "processes" | 0.012 | "balance" |
| 0.016 | "market" | 0.017 | "growth" | 0.011 | "life" |
| 0.015 | "process" | 0.016 | "none" | 0.011 | "politics" |
| 0.014 | "time" | 0.014 | "opportunities" | 0.011 | "business" |
| 0.014 | "hikes" | 0.014 | "salary" | 0.009 | "culture" |
| 0.014 | "project" | 0.013 | "managers" | 0.009 | "layoffs" |
| 0.013 | "growth" | 0.012 | "employees" | 0.008 | "employees" |
| 0.012 | "promotion" | 0.012 | "lot" | 0.008 | "lack" |
| 0.011 | "bonus" | 0.012 | "companies" | 0.008 | "team" |
| 0.010 | "place" | 0.012 | "projects" | 0.007 | "people" |
| 0.010 | "salary" | 0.012 | "organization" | 0.007 | "lots" |
| 0.010 | "opportunity" | 0.011 | "things" | 0.007 | "focus" |
| 0.009 | "years" | 0.010 | "bureaucracy" | 0.006 | "employee" |
| 0.009 | "promotions" | 0.010 | "everything" | 0.006 | "changes" |
| 0.008 | "development" | 0.010 | "manager" | 0.006 | "job" |
| 0.008 | "structure" | 0.009 | "times" | 0.006 | "cost" |
| 0.008 | "industry" | 0.009 | "experience" | 0.005 | "leadership" |

*Table 18: LDA Topics for IBM's Cons (without bigrams)*

| probability-0 | word-0 | probability-1 | word-1 | probability-2 | word-2 |
|---|---|---|---|---|---|
| 0.051 | "benefits" | 0.022 | "technology" | 0.044 | "environment" |
| 0.050 | "balance" | 0.020 | "products" | 0.037 | "opportunities" |
| 0.048 | "people" | 0.016 | "culture" | 0.024 | "work" |
| 0.042 | "work" | 0.014 | "people" | 0.020 | "compensation" |
| 0.039 | "life" | 0.014 | "experience" | 0.020 | "career" |
| 0.038 | "place" | 0.013 | "technologies" | 0.019 | "benefits" |
| 0.031 | "pay" | 0.013 | "world" | 0.019 | "people" |
| 0.020 | "salary" | 0.009 | "benefits" | 0.019 | "lot" |
| 0.015 | "job" | 0.009 | "areas" | 0.018 | "projects" |
| 0.013 | "culture" | 0.008 | "product" | 0.017 | "lots" |
| 0.013 | "perks" | 0.008 | "problems" | 0.016 | "team" |
| 0.011 | "opportunity" | 0.008 | "fun" | 0.016 | "growth" |
| 0.009 | "things" | 0.008 | "years" | 0.013 | "culture" |
| 0.009 | "health" | 0.008 | "impact" | 0.013 | "hours" |
| 0.009 | "lots" | 0.008 | "edge" | 0.012 | "colleagues" |
| 0.009 | "variety" | 0.007 | "exposure" | 0.011 | "management" |
| 0.008 | "benefit" | 0.007 | "companies" | 0.009 | "employees" |
| 0.008 | "care" | 0.007 | "customers" | 0.009 | "campus" |
| 0.007 | "coworkers" | 0.007 | "ability" | 0.008 | "flexibility" |
| 0.007 | "employee" | 0.006 | "business" | 0.008 | "industry" |

*Table 19: LDA Topics for Microsoft's Pros (without bigrams)*

| probability-0 | word-0 | probability-1 | word-1 | probability-2 | word-2 |
|---|---|---|---|---|---|
| 0.027 | "work" | 0.018 | "management" | 0.016 | "nothing" |
| 0.025 | "balance" | 0.018 | "time" | 0.015 | "growth" |
| 0.018 | "people" | 0.015 | "politics" | 0.013 | "review" |
| 0.018 | "life" | 0.015 | "hours" | 0.013 | "companies" |
| 0.014 | "things" | 0.014 | "team" | 0.011 | "bit" |
| 0.012 | "times" | 0.014 | "culture" | 0.010 | "compensation" |
| 0.011 | "lot" | 0.014 | "none" | 0.010 | "pay" |
| 0.011 | "place" | 0.012 | "environment" | 0.010 | "career" |
| 0.010 | "teams" | 0.010 | "employees" | 0.010 | "processes" |
| 0.010 | "organization" | 0.010 | "opportunities" | 0.009 | "system" |
| 0.009 | "management" | 0.010 | "manager" | 0.009 | "politics" |
| 0.008 | "competition" | 0.009 | "bureaucracy" | 0.009 | "teams" |
| 0.008 | "groups" | 0.009 | "changes" | 0.009 | "process" |
| 0.008 | "projects" | 0.008 | "managers" | 0.008 | "benefits" |
| 0.007 | "experience" | 0.008 | "work" | 0.008 | "market" |
| 0.007 | "politics" | 0.007 | "everything" | 0.008 | "lots" |
| 0.006 | "managers" | 0.007 | "pressure" | 0.007 | "office" |
| 0.006 | "decisions" | 0.007 | "people" | 0.007 | "industry" |
| 0.006 | "others" | 0.007 | "way" | 0.006 | "structure" |
| 0.006 | "meetings" | 0.007 | "change" | 0.006 | "technologies" |

*Table 20: LDA Topics for Microsoft's Cons (without bigrams)*

| probability-0 | word-0 | probability-1 | word-1 | probability-2 | word-2 |
|---|---|---|---|---|---|
| 0.056 | "benefits" | 0.043 | "work" | 0.047 | "culture" |
| 0.054 | "people" | 0.025 | "discounts" | 0.026 | "team" |
| 0.035 | "pay" | 0.017 | "home" | 0.024 | "lot" |
| 0.034 | "environment" | 0.017 | "life" | 0.020 | "experience" |
| 0.028 | "place" | 0.015 | "benefits" | 0.019 | "people" |
| 0.022 | "job" | 0.015 | "environment" | 0.018 | "hours" |
| 0.020 | "products" | 0.015 | "balance" | 0.016 | "perks" |
| 0.019 | "time" | 0.014 | "products" | 0.014 | "staff" |
| 0.018 | "employees" | 0.014 | "managers" | 0.013 | "compensation" |
| 0.017 | "training" | 0.013 | "customers" | 0.012 | "things" |
| 0.016 | "management" | 0.012 | "schedule" | 0.011 | "care" |
| 0.015 | "coworkers" | 0.012 | "discount" | 0.010 | "employees" |
| 0.015 | "fun" | 0.012 | "product" | 0.010 | "benefits" |
| 0.015 | "opportunities" | 0.010 | "people" | 0.010 | "technology" |
| 0.013 | "growth" | 0.010 | "colleagues" | 0.010 | "friends" |
| 0.012 | "part" | 0.009 | "resume" | 0.009 | "everyone" |
| 0.012 | "lots" | 0.009 | "support" | 0.009 | "products" |
| 0.010 | "opportunity" | 0.009 | "pay" | 0.009 | "brand" |
| 0.010 | "work" | 0.008 | "something" | 0.008 | "money" |
| 0.009 | "salary" | 0.008 | "atmosphere" | 0.007 | "companies" |

*Table 21: LDA Topics for Apple's Pros (with bigrams)*

| probability-0 | word-0 | probability-1 | word-1 | probability-2 | word-2 |
|---|---|---|---|---|---|
| 0.021 | "customers" | 0.032 | "hours" | 0.037 | "work" |
| 0.019 | "people" | 0.026 | "none" | 0.033 | "balance" |
| 0.013 | "employees" | 0.022 | "nothing" | 0.030 | "life" |
| 0.013 | "schedule" | 0.021 | "times" | 0.030 | "cons" |
| 0.013 | "pay" | 0.015 | "weekends" | 0.029 | "time" |
| 0.013 | "growth" | 0.011 | "home" | 0.024 | "job" |
| 0.012 | "management" | 0.010 | "everything" | 0.022 | "management" |
| 0.011 | "managers" | 0.010 | "lot" | 0.017 | "environment" |
| 0.010 | "opportunities" | 0.010 | "bit" | 0.016 | "hours" |
| 0.010 | "career" | 0.009 | "experience" | 0.012 | "holidays" |
| 0.009 | "room" | 0.008 | "pressure" | 0.011 | "place" |
| 0.009 | "store" | 0.008 | "customer" | 0.009 | "position" |
| 0.009 | "advancement" | 0.007 | "schedules" | 0.009 | "part" |
| 0.009 | "culture" | 0.007 | "expectations" | 0.009 | "day" |
| 0.008 | "politics" | 0.007 | "metrics" | 0.008 | "mobility" |
| 0.007 | "level" | 0.006 | "everyone" | 0.008 | "anything" |
| 0.007 | "lack" | 0.006 | "work" | 0.007 | "stress" |
| 0.007 | "days" | 0.006 | "thing" | 0.006 | "family" |
| 0.007 | "lot" | 0.005 | "numbers" | 0.006 | "store" |
| 0.006 | "benefits" | 0.005 | "competition" | 0.006 | "weekend" |

*Table 22: LDA Topics for Apple's Cons (with bigrams)*

| probability-0 | word-0 | probability-1 | word-1 | probability-2 | word-2 |
|---|---|---|---|---|---|
| 0.032 | "opportunities" | 0.092 | "work" | 0.046 | "place" |
| 0.020 | "lots" | 0.088 | "balance" | 0.036 | "people" |
| 0.020 | "job" | 0.078 | "life" | 0.022 | "products" |
| 0.019 | "technology" | 0.042 | "culture" | 0.018 | "lot" |
| 0.018 | "career" | 0.039 | "environment" | 0.018 | "technologies" |
| 0.017 | "training" | 0.037 | "benefits" | 0.015 | "compensation" |
| 0.015 | "experience" | 0.026 | "team" | 0.012 | "benefits" |
| 0.014 | "name" | 0.023 | "pay" | 0.012 | "office" |
| 0.013 | "growth" | 0.022 | "management" | 0.011 | "home" |
| 0.013 | "product" | 0.021 | "home" | 0.010 | "resources" |
| 0.013 | "sales" | 0.020 | "hours" | 0.010 | "learning" |
| 0.012 | "brand" | 0.018 | "timings" | 0.009 | "employee" |
| 0.011 | "people" | 0.014 | "flexibility" | 0.009 | "schedule" |
| 0.010 | "industry" | 0.013 | "pressure" | 0.009 | "exposure" |
| 0.010 | "benefits" | 0.013 | "people" | 0.008 | "employees" |
| 0.010 | "colleagues" | 0.013 | "salary" | 0.008 | "culture" |
| 0.010 | "projects" | 0.010 | "opportunity" | 0.008 | "things" |
| 0.009 | "opportunity" | 0.008 | "managers" | 0.007 | "package" |
| 0.009 | "development" | 0.007 | "stability" | 0.007 | "customers" |
| 0.009 | "working" | 0.007 | "teams" | 0.007 | "freedom" |

*Table 23: LDA Topics for Oracle's Pros (with bigrams)*

| probability-0 | word-0 | probability-1 | word-1 | probability-2 | word-2 |
|---|---|---|---|---|---|
| 0.017 | "compensation" | 0.033 | "growth" | 0.042 | "hikes" |
| 0.016 | "people" | 0.031 | "work" | 0.032 | "hike" |
| 0.014 | "sales" | 0.022 | "management" | 0.032 | "cons" |
| 0.011 | "market" | 0.022 | "career" | 0.022 | "nothing" |
| 0.010 | "things" | 0.019 | "opportunities" | 0.019 | "bonus" |
| 0.010 | "management" | 0.014 | "salary" | 0.018 | "years" |
| 0.009 | "companies" | 0.012 | "none" | 0.016 | "raises" |
| 0.009 | "processes" | 0.012 | "balance" | 0.015 | "promotions" |
| 0.008 | "times" | 0.012 | "place" | 0.014 | "time" |
| 0.008 | "pay" | 0.012 | "environment" | 0.013 | "bonuses" |
| 0.008 | "pressure" | 0.011 | "life" | 0.013 | "employees" |
| 0.007 | "organization" | 0.010 | "process" | 0.012 | "benefits" |
| 0.007 | "salaries" | 0.010 | "technology" | 0.012 | "promotion" |
| 0.007 | "hours" | 0.010 | "raise" | 0.011 | "technologies" |
| 0.007 | "increase" | 0.010 | "politics" | 0.009 | "team" |
| 0.007 | "lot" | 0.009 | "development" | 0.008 | "opportunity" |
| 0.007 | "training" | 0.009 | "manager" | 0.008 | "culture" |
| 0.007 | "increases" | 0.009 | "managers" | 0.007 | "pay" |
| 0.006 | "employees" | 0.008 | "bureaucracy" | 0.007 | "year" |
| 0.006 | "anything" | 0.008 | "increments" | 0.007 | "increment" |

*Table 24: LDA Topics for Oracle's Cons (with bigrams)*

| probability-0 | word-0 | probability-1 | word-1 | probability-2 | word-2 |
|---|---|---|---|---|---|
| 0.065 | "environment" | 0.059 | "culture" | 0.056 | "benefits" |
| 0.064 | "place" | 0.034 | "perks" | 0.041 | "food" |
| 0.064 | "people" | 0.022 | "people" | 0.037 | "work" |
| 0.030 | "work" | 0.021 | "job" | 0.030 | "balance" |
| 0.022 | "lot" | 0.019 | "everything" | 0.027 | "perks" |
| 0.020 | "salary" | 0.017 | "world" | 0.025 | "life" |
| 0.020 | "time" | 0.017 | "opportunities" | 0.019 | "lots" |
| 0.018 | "pay" | 0.011 | "everyone" | 0.018 | "people" |
| 0.018 | "team" | 0.011 | "technology" | 0.017 | "colleagues" |
| 0.017 | "projects" | 0.010 | "products" | 0.016 | "compensation" |
| 0.014 | "career" | 0.010 | "growth" | 0.016 | "pay" |
| 0.013 | "opportunity" | 0.009 | "projects" | 0.015 | "experience" |
| 0.013 | "opportunities" | 0.009 | "working" | 0.015 | "office" |
| 0.013 | "fun" | 0.007 | "benefits" | 0.014 | "coworkers" |
| 0.012 | "management" | 0.006 | "way" | 0.013 | "employees" |
| 0.012 | "food" | 0.006 | "colleagues" | 0.012 | "things" |
| 0.009 | "problems" | 0.006 | "infrastructure" | 0.010 | "culture" |
| 0.008 | "workplace" | 0.006 | "name" | 0.009 | "flexibility" |
| 0.008 | "benefits" | 0.006 | "scale" | 0.008 | "teams" |
| 0.008 | "hours" | 0.006 | "engineering" | 0.008 | "gym" |

*Table 25: LDA Topics for Google's Pros (with bigrams)*

| probability-0 | word-0 | probability-1 | word-1 | probability-2 | word-2 |
|---|---|---|---|---|---|
| 0.082 | "cons" | 0.037 | "none" | 0.040 | "work" |
| 0.053 | "nothing" | 0.028 | "people" | 0.021 | "balance" |
| 0.026 | "hours" | 0.026 | "time" | 0.019 | "management" |
| 0.023 | "things" | 0.016 | "lot" | 0.018 | "politics" |
| 0.021 | "times" | 0.014 | "culture" | 0.018 | "life" |
| 0.019 | "place" | 0.013 | "bit" | 0.016 | "job" |
| 0.016 | "environment" | 0.009 | "everyone" | 0.011 | "promotion" |
| 0.016 | "pressure" | 0.009 | "opportunities" | 0.010 | "team" |
| 0.013 | "impact" | 0.009 | "experience" | 0.010 | "office" |
| 0.013 | "bureaucracy" | 0.008 | "career" | 0.010 | "process" |
| 0.010 | "anything" | 0.008 | "organization" | 0.010 | "teams" |
| 0.009 | "projects" | 0.008 | "way" | 0.009 | "managers" |
| 0.009 | "work" | 0.008 | "salary" | 0.009 | "pay" |
| 0.008 | "food" | 0.008 | "competition" | 0.009 | "employees" |
| 0.007 | "lot" | 0.007 | "work" | 0.009 | "growth" |
| 0.007 | "processes" | 0.007 | "commute" | 0.008 | "companies" |
| 0.006 | "everything" | 0.006 | "stress" | 0.007 | "lots" |
| 0.006 | "corporation" | 0.006 | "management" | 0.007 | "hours" |
| 0.006 | "structure" | 0.006 | "everything" | 0.007 | "project" |
| 0.006 | "days" | 0.006 | "promotions" | 0.007 | "day" |

*Table 26: LDA Topics for Google's Cons (with bigrams)*

| probability-0 | word-0 | probability-1 | word-1 | probability-2 | word-2 |
|---|---|---|---|---|---|
| 0.059 | "environment" | 0.019 | "experience" | 0.076 | "work" |
| 0.049 | "place" | 0.019 | "benefits" | 0.066 | "balance" |
| 0.026 | "career" | 0.018 | "job" | 0.056 | "life" |
| 0.025 | "projects" | 0.017 | "training" | 0.052 | "people" |
| 0.022 | "time" | 0.013 | "employees" | 0.041 | "culture" |
| 0.021 | "growth" | 0.012 | "resources" | 0.035 | "home" |
| 0.018 | "opportunities" | 0.010 | "business" | 0.027 | "opportunities" |
| 0.016 | "lot" | 0.009 | "technology" | 0.022 | "flexibility" |
| 0.016 | "salary" | 0.009 | "clients" | 0.021 | "hours" |
| 0.014 | "things" | 0.009 | "opportunities" | 0.018 | "benefits" |
| 0.014 | "work" | 0.009 | "resume" | 0.017 | "lots" |
| 0.012 | "skills" | 0.008 | "knowledge" | 0.015 | "pay" |
| 0.012 | "people" | 0.008 | "people" | 0.014 | "colleagues" |
| 0.011 | "part" | 0.008 | "industry" | 0.013 | "opportunity" |
| 0.010 | "management" | 0.007 | "years" | 0.012 | "working" |
| 0.009 | "managers" | 0.007 | "team" | 0.012 | "name" |
| 0.008 | "learning" | 0.007 | "education" | 0.011 | "technologies" |
| 0.008 | "organization" | 0.007 | "support" | 0.011 | "technology" |
| 0.008 | "flexibility" | 0.007 | "management" | 0.011 | "brand" |
| 0.008 | "office" | 0.007 | "teams" | 0.009 | "schedule" |

*Table 27: LDA Topics for IBM's Pros (with bigrams)*

| probability-0 | word-0 | probability-1 | word-1 | probability-2 | word-2 |
|---|---|---|---|---|---|
| 0.034 | "cons" | 0.024 | "management" | 0.031 | "work" |
| 0.021 | "hike" | 0.023 | "salary" | 0.025 | "nothing" |
| 0.016 | "hikes" | 0.017 | "process" | 0.017 | "hours" |
| 0.015 | "processes" | 0.017 | "career" | 0.017 | "compensation" |
| 0.015 | "growth" | 0.015 | "managers" | 0.014 | "balance" |
| 0.013 | "people" | 0.015 | "none" | 0.013 | "politics" |
| 0.012 | "time" | 0.013 | "lot" | 0.013 | "life" |
| 0.011 | "opportunities" | 0.012 | "market" | 0.012 | "pay" |
| 0.011 | "companies" | 0.012 | "lack" | 0.010 | "culture" |
| 0.010 | "projects" | 0.012 | "employees" | 0.010 | "times" |
| 0.009 | "organization" | 0.011 | "place" | 0.009 | "environment" |
| 0.009 | "job" | 0.011 | "manager" | 0.009 | "management" |
| 0.009 | "years" | 0.009 | "everything" | 0.009 | "team" |
| 0.009 | "bureaucracy" | 0.009 | "people" | 0.009 | "salaries" |
| 0.008 | "pay" | 0.008 | "growth" | 0.009 | "opportunity" |
| 0.008 | "things" | 0.008 | "promotions" | 0.009 | "project" |
| 0.008 | "benefits" | 0.008 | "development" | 0.009 | "promotion" |
| 0.008 | "employees" | 0.008 | "anything" | 0.007 | "day" |
| 0.008 | "year" | 0.007 | "structure" | 0.006 | "people" |
| 0.007 | "business" | 0.007 | "tape" | 0.006 | "increment" |

*Table 28: LDA Topics for IBM's Cons (with bigrams)*

| probability-0 | word-0 | probability-1 | word-1 | probability-2 | word-2 |
|---|---|---|---|---|---|
| 0.040 | "place" | 0.039 | "environment" | 0.057 | "people" |
| 0.021 | "team" | 0.035 | "opportunities" | 0.052 | "benefits" |
| 0.021 | "salary" | 0.023 | "lots" | 0.051 | "work" |
| 0.019 | "projects" | 0.020 | "career" | 0.049 | "balance" |
| 0.015 | "job" | 0.018 | "compensation" | 0.040 | "life" |
| 0.015 | "experience" | 0.018 | "benefits" | 0.035 | "culture" |
| 0.013 | "resources" | 0.015 | "growth" | 0.030 | "pay" |
| 0.012 | "colleagues" | 0.014 | "work" | 0.023 | "products" |
| 0.011 | "time" | 0.014 | "people" | 0.021 | "technology" |
| 0.011 | "people" | 0.013 | "opportunity" | 0.015 | "perks" |
| 0.011 | "benefits" | 0.012 | "flexibility" | 0.014 | "lot" |
| 0.009 | "hours" | 0.010 | "employees" | 0.013 | "technologies" |
| 0.008 | "fun" | 0.009 | "impact" | 0.013 | "things" |
| 0.008 | "managers" | 0.008 | "employee" | 0.010 | "world" |
| 0.007 | "health" | 0.008 | "package" | 0.008 | "edge" |
| 0.007 | "lot" | 0.008 | "development" | 0.008 | "problems" |
| 0.007 | "care" | 0.008 | "management" | 0.007 | "exposure" |
| 0.007 | "name" | 0.007 | "campus" | 0.007 | "millions" |
| 0.007 | "learning" | 0.007 | "teams" | 0.006 | "talent" |
| 0.007 | "brand" | 0.007 | "working" | 0.006 | "location" |

*Table 29: LDA Topics for Microsoft's Pros (with bigrams)*

| probability-0 | word-0 | probability-1 | word-1 | probability-2 | word-2 |
|---|---|---|---|---|---|
| 0.023 | "politics" | 0.018 | "people" | 0.032 | "work" |
| 0.021 | "cons" | 0.015 | "hours" | 0.027 | "balance" |
| 0.020 | "management" | 0.015 | "time" | 0.020 | "life" |
| 0.015 | "things" | 0.012 | "environment" | 0.016 | "none" |
| 0.013 | "nothing" | 0.012 | "review" | 0.014 | "times" |
| 0.010 | "growth" | 0.011 | "system" | 0.012 | "culture" |
| 0.010 | "place" | 0.010 | "changes" | 0.011 | "companies" |
| 0.009 | "way" | 0.010 | "teams" | 0.010 | "pay" |
| 0.009 | "processes" | 0.010 | "lot" | 0.010 | "bit" |
| 0.008 | "career" | 0.010 | "team" | 0.010 | "job" |
| 0.008 | "lots" | 0.009 | "organization" | 0.008 | "bureaucracy" |
| 0.008 | "employees" | 0.009 | "opportunities" | 0.008 | "compensation" |
| 0.008 | "people" | 0.008 | "change" | 0.008 | "benefits" |
| 0.008 | "process" | 0.008 | "work" | 0.008 | "managers" |
| 0.007 | "managers" | 0.007 | "everything" | 0.008 | "pressure" |
| 0.007 | "group" | 0.007 | "management" | 0.006 | "system" |
| 0.007 | "lot" | 0.006 | "performance" | 0.006 | "development" |
| 0.007 | "manager" | 0.006 | "years" | 0.006 | "technologies" |
| 0.007 | "leadership" | 0.006 | "year" | 0.005 | "model" |
| 0.007 | "groups" | 0.006 | "technology" | 0.005 | "management" |

*Table 30: LDA Topics for Microsoft's Cons (with bigrams)*

# Appendix G: Topic Modelling Word Cloud


*Word Cloud 1: Apple Pros*


*Word Cloud 2: Apple Cons*


***Word Cloud 3: Oracle Pros***


***Word Cloud 4: Oracle Cons***


*Word Cloud 5: Google Pros*


*Word Cloud 6: Google Cons*


*Word Cloud 7: IBM Pros*


*Word Cloud 8: IBM Cons*


***Word Cloud 9: Microsoft Pros***


***Word Cloud 10: Microsoft Cons***