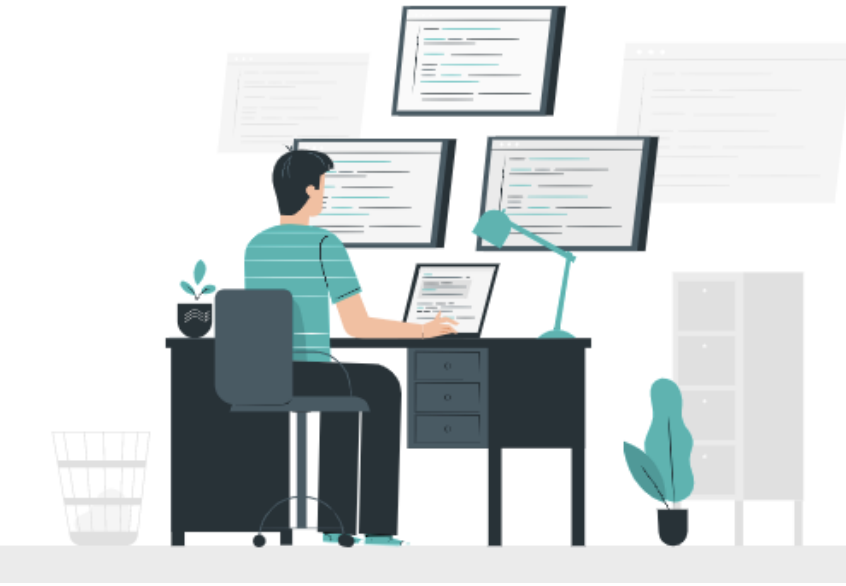
The Discovery Study of Company Culture through Indeed



Fordham's Graduate Gabelli School of Business
Text Analytics (BYGB - 7977 V01)
Spring 2021

Yuge Ma, Muhetaer Mayila, Juliann Negron,
Nicholas Sansone, and Yohabet Tejeida Caballero

Executive Summary

In the competitive culture of the modern-day job market, a company's public image is crucial in attracting, recruiting, and retaining potential employees. Internal reputation parallels a company's public image, with both being factored heavily in one's decision of a job- the better reviews and reputation a company has, the more skilled and driven the workers are that apply to that enterprise. Our team wanted to explore this relationship and how company culture affects their engagement and retention ability.

Due to the magnitude of importance it holds within the employment market, *Indeed* is an excellent source for data in analyzing company culture- it is a renowned business leader in connecting potential employees with job opportunities through its functionality supporting both job searches and company reviews, with traffic from millions of users daily. Not only that, but the structure of the website allows for us to parse reviews for any company, given by current and former employees. This further allowed us to create a robust dataset containing review information from a large array of predetermined companies.

TABLE OF CONTENTS

Business Goal Analysis

As of 2016, A Glassdoor report [1] uncovered that more than two-thirds of candidates check a company's reviews before accepting a job offer. When it comes to finding the right job, applicants are not just looking for the company with the most awards or best compensations and benefits; they prioritize other things such as gender equality, diversity, transparency, and overall company culture.

This project aims to analyze the overall sentiment of over 131,940 Indeed reviews of six of the most influential financial companies on the Fortune 500 list by using different text classification algorithms.

Below we describe the steps of the project, the theory behind it, and we finish by taking a look at some key findings from our analysis.

Dataset Description

First, we scraped a list of financial companies from Fortunes 500. Using selenium allowed us to create a virtual webpage and mimicked human interaction by automatically entering the company’s name, clicking on the top search result, directing the review page, and scraping the review page by page by clicking on the ‘Next’ button. At last, we got 287,742 reviews from Indeed of 122 companies. Because of the different number of reviews of each company, in order to create an unbiased environment to analyze a company's culture, we sampled the dataset by using the reviews of the following companies in a total of 131,940 rows of data.

The figure below shows all the elements we scraped from the review section:

Figure 1: Example of an Indeed Review

The dataset includes variables such as star, title, content, pros, cons, author_title, author_status, location, time, and company.

The details of each variable are listed below:

Variables Names	**Scale**	**Type**	**Explanations**
star	Ratio	Integer	The stars each reviewer gave to the company: ● 1-2 stars indicate employees are dissatisfied ● 3 stars indicate employees feel average ● 4-5 stars indicate employees are satisfied
title	Nominal	String	A subject title for the review content
content	Nominal	String	The main review content, and also the main element we focus on
pros	Nominal	String	Advantages of the companies that reviewers recommend; null value is accepted in this section.
cons	Nominal	String	Disadvantages of the companies that reviewers do not recommend; null value is accepted in this section.
author_title	Nominal	String	Title or positions of the reviewer.

author_status	Nominal	String	Flag variable of a former employee and current employee.
location	Nominal	String	The location of the company that reviewers work or worked.
time	Nominal	String	The date of the review
company	Nominal	String	Company's name

Figure 2: Table of Variables

Data Visualizations

The following visualizations were created to better understand project's dataset:

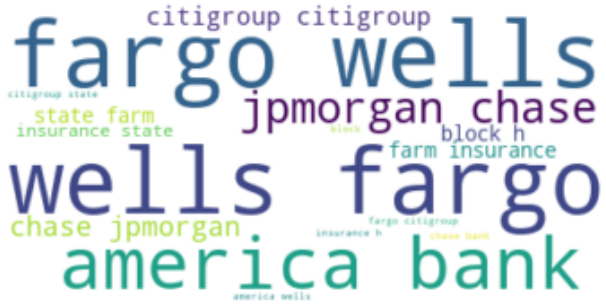


Figure 1: Word Cloud of Companies names

Figure 1 is a word cloud of the six company names we analyzed in this project, which contained Wells Fargo, Bank of America, JP Morgan Chase, Citi Group, State Farm Insurance, and H&R Block.

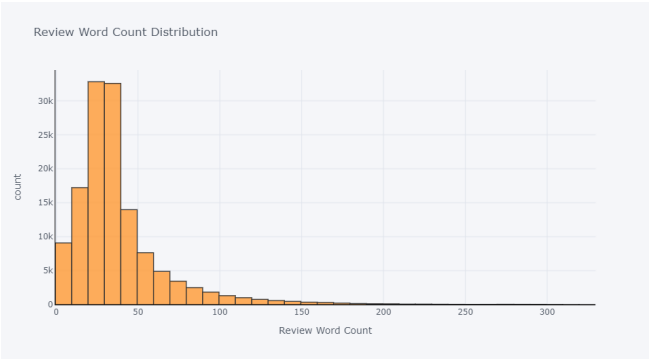


Figure 2: Histogram of the Review Word Count Distribution

Figure 2 is a histogram of the word count distribution of the reviews in our dataset. As you can see, it shows that over 60,000 reviews have between 30 and 40 words per review, and overall more than 15,000 reviews with more than 50 words each.

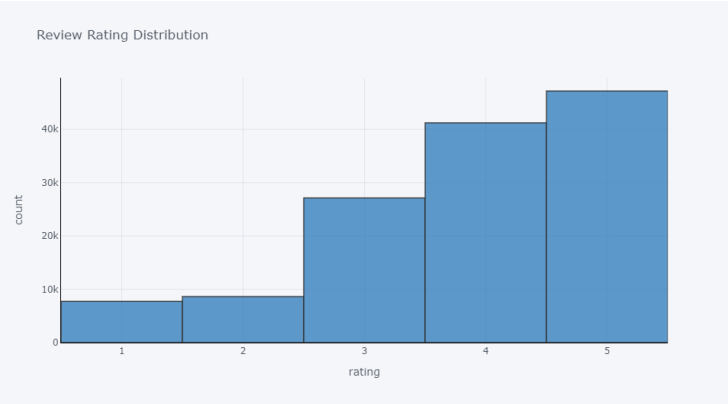


Figure 3: Histogram of Review Rating Distribution

Figure 3 is a histogram of the rating distribution of the reviews in our dataset. As you can see, it shows that a little under 40,000 reviews have less than three stars, and over 80,000 reviews have more than three-star reviews. As a result, we aim to find a model that has higher specificity, or more false negatives.

System Design

According to the major steps this project implemented, it could be concluded as the following flowchart:

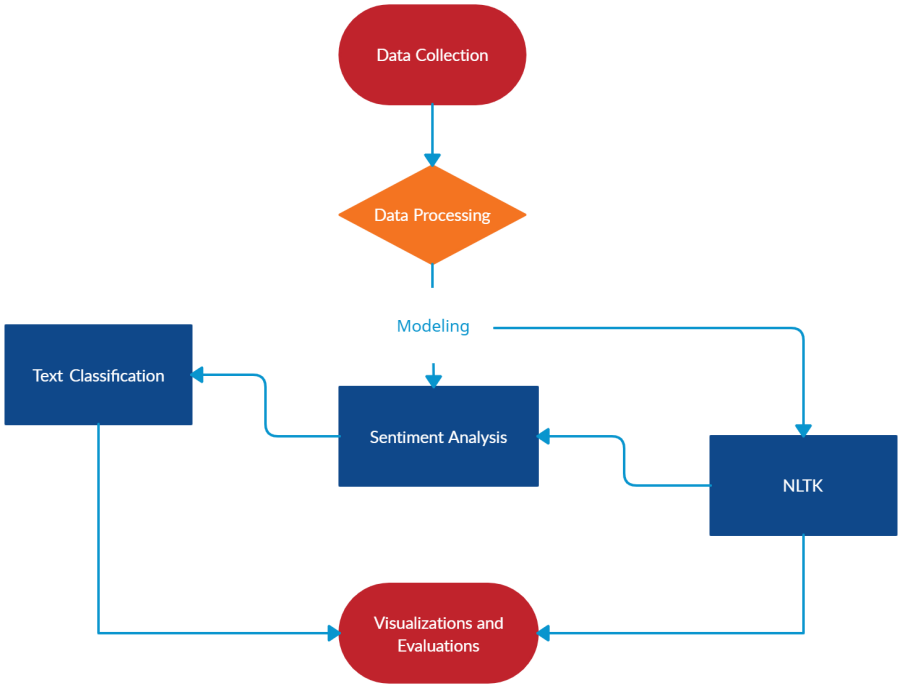


Figure 4: System Design Flow

1. **Data Collection**

   Using Selenium to scrape Indeed.com, we collected a total of 287,742 reviews from 122 companies in the Fortune 500 list.

2. **Data Processing**

   Picked a total of 6 companies to make our analysis, which is in a total of 131,940 rows of data- each company had over 10,000 different reviews within our dataset—applied data cleaning process on Jupyter Notebook and Google Colab.

3. **Sentiment Analysis - NLTK**

   Remove non-alphabets, token alphabets-only list, assign Porter Stemmer to a lambda function to run on each line of value, assign lemmatizer to a lambda function to run on each line of value.

4. **Sentiment Analysis - Bag Of Words**

   Apply bigram frequency and trigram frequency on six company's "content" columns, "pros" columns, "cons" columns separately.

5. **Sentiment Classification - Dictionary Approach**

   Apply BingLiu's Dictionary, LM Dictionary, TextBlob, Vader, and compare their accuracy, precision score, recall score, and f1-score.

6. **Visualizations and Evaluations**

   Analyze the most frequent word in the reviews with visualizations to value the company's possible culture.

System Implementation

Data collection & Data Processing

By using Selenium to scrape Indeed.com, we collected a total of 287,742 reviews from 122 companies in the Fortune 500 list. In order to create an unbiased environment to analyze a specific industry's company culture, we sampled our dataset by creating a list of the companies with reviews greater than 10,000. In the end, we picked a total of 6 companies to make our analysis: "JPMorgan Chase" - 18,721 reviews, "Bank of America" - 29,876 reviews, "Wells Fargo" - 40,360 reviews, "Citigroup" - 18,041 reviews, "State Farm Insurance" - 12,241 reviews, and "H&R Block" - 12,701 reviews. This totals 131,940 rows of data.

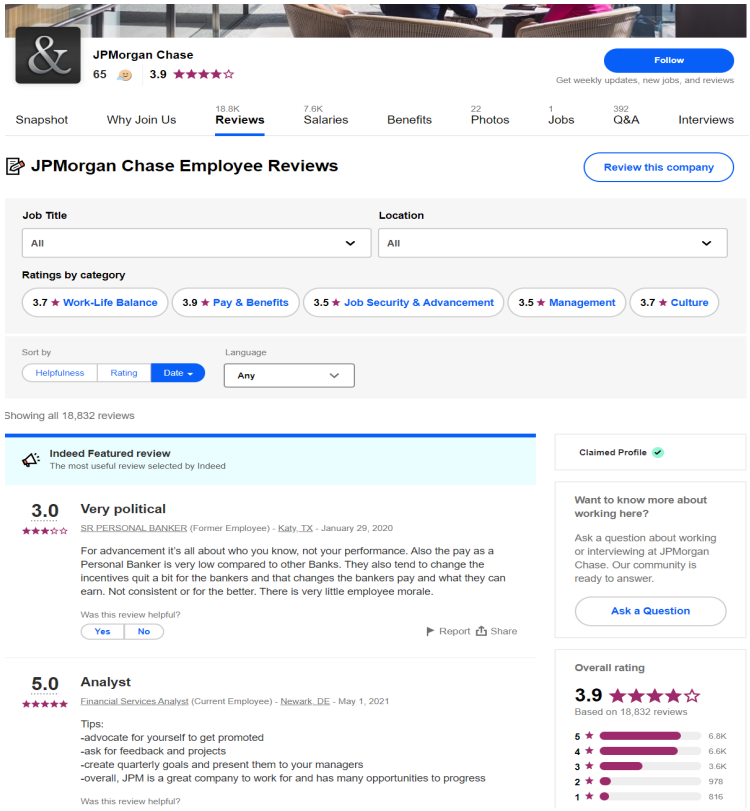


Figure 5: Example of JPMorgan Chase's reviews page on Indeed

The clean process was completed on Jupyter Notebook and Google Colab. For missing values in the dataset, since our NA values are less than 10% of total values, in order to perform a more accurate sentiment analysis later, we removed all the missing values. After going through all the processes, the data was clean, organized, and ready for sentiment analysis to be performed on it.

	star	title	content	pros	cons	author_title	author_status	location	time	company
0	4.0	Very competitive, one must love what they do t...	Conducted meeting where we spoke about how to ...	Flexibility on time	Not much growth	Sr. Personal Banker/Business Specialist	Current Employee	177 Montague St, Brooklyn, NY 11201	April 5, 2017	JPMorgan Chase
1	3.0	Awesome benefits	Worked for JPMC for over 10 years & the time a...	Vacation benefits	Not very great @ showing their appreciation	Fraud Investigator	Former Employee	Columbus, OH	March 24, 2021	JPMorgan Chase
2	2.0	Big corporation.	Manager only cares about bonus. Typical, dead-...	Vacation time, sick time, pay, benefits	Work load, stressful, very high call volume, s...	Lead teller operations specialist	Former Employee	Youngstown, OH	March 24, 2021	JPMorgan Chase
3	1.0	Bad management	Don't expect to get to chase and promote. Unde...	NaN	NaN	Associate	Current Employee	Jacksonville, FL	March 23, 2021	JPMorgan Chase
4	3.0	Great culture	Culture is great but pay is below average. Eve...	NaN	NaN	Corporate analyst development program	Former Employee	Dallas, TX	March 23, 2021	JPMorgan Chase

Figure 6: Example Data Frame

Python Implementation of Sentiment Analysis & Classification

NLTK

Stop-words are words that, from a non-linguistic view, do not carry information- usually, we remove them to help the methods perform better. Stemming provides an inexpensive mechanism to merge words and reduce dimensionality, as words often have different spelling but similar meanings. Therefore, this was an integral part of performing sentiment analysis as different forms of the same word are usually problematic for text analysis.

Procedures: On "content" column, "pros" column, "cons" column - remove non-alphabet characters, token the remaining word list, assign Porter Stemmer to a lambda function to run on each line of value, and assign lemmatizer to a lambda function to run on each line of value.

```
# remove non alphabets
remove_non_alphabets = lambda x: re.sub(r'[^a-zA-Z]',' ',x)

# tokenn alphabets-only list
tokenize = lambda x: word_tokenize(x)

# assign ps to a lambda function to run on each line of value
ps = PorterStemmer()
stem = lambda w: [ ps.stem(x) for x in w ]

# assign lemmatizer to a lambda function to run on each line of value
lemmatizer = WordNetLemmatizer()
leammtizer = lambda x: [ lemmatizer.lemmatize(word) for word in x ]
```

```
# apply all above methods to the column ''
print('Processing : [=', end='')
df['content'] = df['content'].apply(remove_non_alphabets)
print('=', end='')
df['content'] = df['content'].apply(tokenize)
print('=', end='')
df['content'] = df['content'].apply(stem)
print('=', end='')
df['content'] = df['content'].apply(leammtizer)
print('=', end='')
df['content'] = df['content'].apply(lambda x: ' '.join(x))
print('] : Completed', end='')
df.head()
```

```
Processing : [=====] : Completed
```

```
print('Processing : [=', end='')
df2['pros'] = df2['pros'].apply(remove_non_alphabets)
print('=', end='')
df2['pros'] = df2['pros'].apply(tokenize)
print('=', end='')
df2['pros'] = df2['pros'].apply(stem)
print('=', end='')
df2['pros'] = df2['pros'].apply(leammtizer)
print('=', end='')
df2['pros'] = df2['pros'].apply(lambda x: ' '.join(x))
print('] : Completed', end='')
df2.head()
```

```
Processing : [=====] : Completed
```

```
print('Processing : [=', end='')
df3['cons'] = df3['cons'].apply(remove_non_alphabets)
print('=', end='')
df3['cons'] = df3['cons'].apply(tokenize)
print('=', end='')
df3['cons'] = df3['cons'].apply(stem)
print('=', end='')
df3['cons'] = df3['cons'].apply(leammtizer)
print('=', end='')
df3['cons'] = df3['cons'].apply(lambda x: ' '.join(x))
print('] : Completed', end='')
df3.head()
```

```
Processing : [=====] : Completed
```

Figure 7: Column's NLTK Performance

Bag Of Words - Frequency Bigram & Trigram

Word frequencies within the text have a power distribution, often represented as either a small number of very frequent words or a big number of low-frequency words.

Procedures: Apply bigram frequency and trigram frequency on “content” column, “pros” column, “cons” column separately to “JPMorgan Chase” - 18721 reviews, “Bank of America” - 29876 reviews, “Wells Fargo” - 40360 reviews, “Citigroup” - 18041 reviews, “State Farm Insurance” - 12241 reviews, and “H&R Block” - 12701 reviews.

```
[(('great', 'benefit'), 333),
 (('free', 'lunch'), 280),
 (('to', 'work'), 221),
 (('good', 'benefit'), 214),
 (('work', 'environ'), 174),
 (('work', 'life'), 173),
 (('life', 'balanc'), 167),
 (('benefit', 'great'), 150),
 (('lot', 'of'), 144),
 (('benefit', 'and'), 134),
 (('co', 'worker'), 132),
 (('good', 'pay'), 121),
 (('work', 'with'), 115),
 (('and', 'benefit'), 114),
 (('great', 'pay'), 98),
 (('benefit', 'good'), 98),
 (('good', 'work'), 98),
 (('pay', 'and'), 95),
 (('place', 'to'), 83),
 (('time', 'off'), 82),
 (('great', 'peopl'), 82),
 (('vacat', 'time'), 75),
 (('excel', 'benefit'), 70),
 (('great', 'work'), 66),
 (('work', 'from'), 65),
 (('from', 'home'), 65),
 (('benefit', 'benefit'), 63),
 (('benefit', 'free'), 62),
 (('health', 'benefit'), 61),
 (('pay', 'benefit'), 59)]
```

```
[(('work', 'life', 'balanc'), 162),
 (('work', 'from', 'home'), 62),
 (('place', 'to', 'work'), 61),
 (('to', 'work', 'with'), 57),
 (('pay', 'and', 'benefit'), 55),
 (('great', 'benefit', 'great'), 37),
 (('good', 'work', 'environ'), 36),
 (('great', 'place', 'to'), 36),
 (('benefit', 'great', 'benefit'), 34),
 (('paid', 'time', 'off'), 34),
 (('great', 'benefit', 'and'), 33),
 (('great', 'benefit', 'good'), 30),
 (('great', 'work', 'environ'), 30),
 (('peopl', 'to', 'work'), 30),
 (('great', 'co', 'worker'), 29),
 (('benefit', 'free', 'lunch'), 28),
 (('great', 'pay', 'and'), 28),
 (('free', 'lunch', 'and'), 26),
 (('good', 'work', 'life'), 26),
 (('to', 'work', 'for'), 26),
 (('good', 'pay', 'and'), 25),
 (('good', 'benefit', 'great'), 25),
 (('close', 'to', 'home'), 24),
 (('benefit', 'work', 'life'), 22),
 (('work', 'environ', 'great'), 22),
 (('free', 'lunch', 'free'), 22),
 (('learn', 'a', 'lot'), 21),
 (('great', 'benefit', 'benefit'), 20),
 (('lot', 'of', 'opportun'), 20),
 (('compani', 'to', 'work'), 20)]
```

```
[(('long', 'hour'), 376),
 (('short', 'break'), 218),
 (('work', 'life'), 209),
 (('life', 'balanc'), 201),
 (('lack', 'of'), 161),
 (('work', 'hour'), 118),
 (('job', 'secur'), 112),
 (('can', 'be'), 93),
 (('to', 'work'), 87),
 (('lot', 'of'), 86),
 (('poor', 'manag'), 84),
 (('of', 'the'), 83),
 (('hard', 'to'), 79),
 (('not', 'enough'), 78),
 (('for', 'advanc'), 78),
 (('long', 'work'), 74),
 (('room', 'for'), 73),
 (('too', 'mani'), 70),
 (('to', 'get'), 69),
 (('in', 'the'), 68),
 (('N', 'a'), 63),
 (('micro', 'manag'), 62),
 (('at', 'time'), 62),
 (('is', 'not'), 58),
 (('due', 'to'), 58),
 (('don', 't'), 54),
 (('to', 'be'), 52),
 (('low', 'pay'), 51),
 (('lay', 'off'), 50),
 (('the', 'compani'), 48)]
```

```
[(('work', 'life', 'balanc'), 195),
 (('long', 'work', 'hour'), 69),
 (('room', 'for', 'advanc'), 50),
 (('a', 'lot', 'of'), 30),
 (('no', 'room', 'for'), 28),
 (('long', 'hour', 'and'), 27),
 (('no', 'job', 'secur'), 26),
 (('hard', 'to', 'get'), 24),
 (('stress', 'at', 'time'), 20),
 (('short', 'break', 'and'), 19),
 (('to', 'move', 'up'), 18),
 (('No', 'work', 'life'), 18),
 (('work', 'from', 'home'), 18),
 (('lack', 'of', 'advanc'), 18),
 (('no', 'work', 'life'), 18),
 (('hour', 'short', 'break'), 17),
 (('work', 'long', 'hour'), 17),
 (('be', 'abl', 'to'), 16),
 (('hard', 'to', 'advanc'), 16),
 (('lack', 'of', 'commun'), 16),
 (('could', 'be', 'better'), 16),
 (('manag', 'long', 'hour'), 16),
 (('hour', 'long', 'hour'), 16),
 (('life', 'balanc', 'is'), 15),
 (('need', 'to', 'be'), 15),
 (('break', 'and', 'lunch'), 15),
 (('not', 'be', 'abl'), 14),
 (('can', 'be', 'stress'), 14),
 (('mani', 'to', 'list'), 14),
 (('poor', 'work', 'life'), 14)]
```

Figure 8: Example of bigram and trigram performance on JPMorgan Chase

Dictionaries Approach

Words are used to determine the sentiment (positive and negative) of a document according to the application. In sentiment classification, opinion/sentiment words are the most important; through the use of several dictionary approaches, it can be inferred that these words contain an opinion on one main object expressed by the author of the document.

Procedures: Apply BingLiu's Dictionary, LM Dictionary, TextBlob, and Vader- compare their accuracy, precision score, recall score, and f1-score. Finally, use the ensemble method to combine the two highest performance measures. In this case, we ensembled the LM & Vader techniques.

	star	content	poscnt_BL	negcnt_BL	netcnt_BL	predict
0	1.0	conduct meet where we spoke about how to impro...	2	2	0	0
1	0.0	work for jpmc for over year the time away bala...	4	2	2	1
2	-1.0	manag onli care about bonu typic dead end corp...	2	3	-1	-1
3	-1.0	don t expect to get to chase and promot under ...	1	1	0	0
4	0.0	cultur is great but pay is below averag everyo...	3	0	3	1

	star	content	poscnt_BL	negcnt_BL	netcnt_BL	predict	poscnt_LM	negcnt_LM	netcnt_LM	predictLM
0	1.0	conduct meet where we spoke about how to impro...	2	2	0	0	0	1	-1	-1
1	0.0	work for jpmc for over year the time away bala...	4	2	2	1	1	0	1	1
2	-1.0	manag onli care about bonu typic dead end corp...	2	3	-1	-1	1	1	0	0
3	-1.0	don t expect to get to chase and promot under ...	1	1	0	0	0	0	0	0
4	0.0	cultur is great but pay is below averag everyo...	3	0	3	1	1	1	0	0

star	content	poscnt_BL	negcnt_BL	netcnt_BL	predict	poscnt_LM	negcnt_LM	netcnt_LM	predictLM	score_TextBlob	predictTB	negscore_Vader	neuscore_
1.0	conduct meet where we spoke about how to impro...	2	2	0	0	0	1	-1	-1	0.405714	1	0.000	
0.0	work for jpmc for over year the time away bala...	4	2	2	1	1	0	1	1	0.400000	1	0.036	
-1.0	manag onli care about bonu typic dead end corp...	2	3	-1	-1	1	1	0	0	-0.200000	-1	0.161	
-1.0	don t expect to get to chase and promot under ...	1	1	0	0	0	0	0	0	-0.100000	-1	0.069	
0.0	cultur is great but pay is below averag everyo...	3	0	3	1	1	1	0	0	0.491667	1	0.132	

Figure 9: Dictionary's approach performance

Model Evaluations

The tables below show the performance measures of the models previously described:

Methods	Positive Precision	Neutral Precision	Negative Precision
BL Dict	70.96	23.76	27.05
LM Dict	70.79	23.55	28.17
TextBlob	64.57	21.51	34.69
VADER	66.86	23.08	42.91
Ensemble	72.40	23.75	37.65

Table1: Sentiment Analysis Precision Levels

Methods	Positive Recall	Neutral Recall	Negative Recall
BL Dict	55.63	19.24	55.58
LM Dict	44.31	36.92	45.30
TextBlob	74.81	15.89	26.72
VADER	79.72	12.96	38.05
Ensemble	43.73	62.15	9.89

Table 2: Sentiment Analysis Recall Levels

Table 1 shows the precision score from all methods, while table 2 shows the recall scores. From these tables, we can infer that the highest precision scores for each type of sentiment are as follows: The Ensemble Method for the positive sentiment reviews, Bing Liu's Dictionary for the neutral sentiment reviews, and Vader for the negative sentiment reviews. On the other hand, the models with the highest recall levels are as follows: Vader for the positive sentiment reviews, The ensemble method for the neutral sentiment reviews, and Bing Lui's Dictionary for the negative sentiment reviews. As mentioned earlier, we are interested in identifying false negatives

over false positives, so the Bing Liu's Dictionary approach is favorable with the highest negative recall. Below, the confusion matrices for each method further our analysis:

LM Dict	Predicted -1	Predicted 0	Predicted 1
Actual -1	8594	6388	3990
Actual 0	7470	9521	8797
Actual 1	14442	24515	30994

Bing Liu	Predicted -1	Predicted 0	Predicted 1
Actual -1	10544	3256	5172
Actual 0	10075	4963	10750
Actual 1	18364	12673	38914

VADER	Predicted -1	Predicted 0	Predicted 1
Actual -1	7218	1836	9918
Actual 0	4723	3343	17722
Actual 1	4482	9305	55764

Ensemble	Predicted -1	Predicted 0	Predicted 1
Actual -1	1876	13710	3386
Actual 0	1486	16026	8276
Actual 1	1621	37741	30589

TextBlob	Predicted -1	Predicted 0	Predicted 1
Actual -1	5070	3020	10882
Actual 0	3855	4097	17836
Actual 1	5690	11931	52330

Figure 10: Confusion matrices of all models

The methods with the highest accuracies in the project are VADER (57.8%), TextBlob (53.6%), and Bing Liu (47.4%). Therefore, Bing Liu's dictionary is the superior classification model due to its relatively high accuracy and specificity.

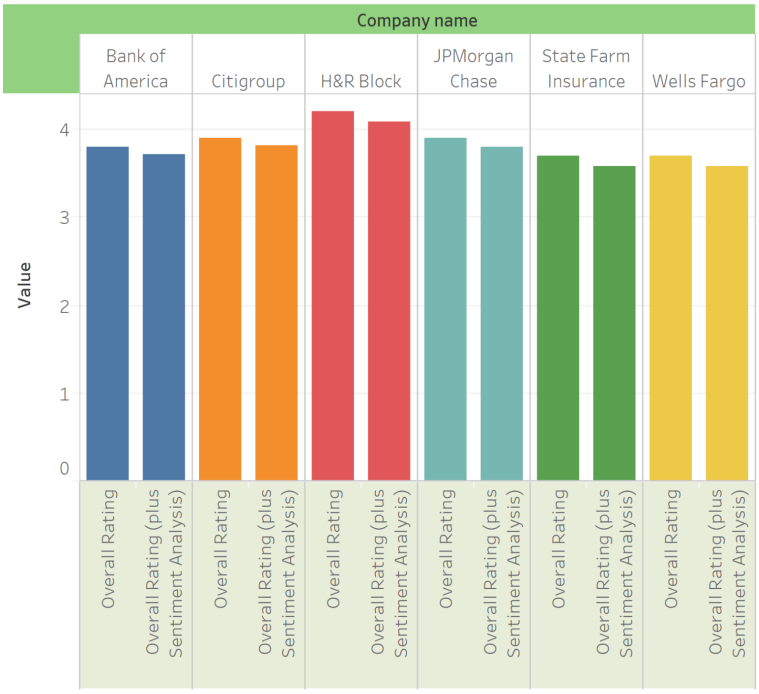


Figure 10: Side-by-Side bar chart of the company's average ratings

Figure 10 is a side-by-side bar chart that displays the average star reviews given to each company by their employees compared to the average star rating given by our sentiment analysis models. As you can see, based on figure 12, H&R Block has the highest ratings while State Farm Insurance, Wells Fargo, and Bank of America have the lowest ratings from the 6 companies. Lastly, it also shows that the star rating given by each employee is higher than what the models estimated for all companies based on the overall sentiment analysis of each review. We want to assume that this is due to the fact that when a person strongly dislikes something, it is easier to convey the message and use the right words in comparison to when people try to convey neutral and positive reviews.

Conclusion and Future Direction

When it comes to finding the right job, applicants are not just looking for the company with the most awards or best compensations and benefits. Nowadays, applicants prioritize other things such as gender equality, diversity, transparency, and the overall company culture.

This project aimed to analyze the overall sentiment of over 131,940 Indeed reviews from six of the most influential financial companies on the Fortune 500 list. We used over four different text classification algorithms to explore how a company's internal reputation parallels its public image. Overall, we discovered that working on any of these six companies offers great compensation and benefits packages, but the work-life balance might be hard to balance and the everyday stress might not be worth the high pay rates.

Moreover, we also found that these companies have a higher employee star rating on Indeed but the overall sentiment of the reviews is lower when compared to each other. Although we did not look for causation in this project, we would like to assume that this might be because people tend to give a higher star rating by itself, but when it comes down to explain the reason behind that rating, it might be harder to put into words.

Lastly, if we were to continue to work on this project, we would like to explore the top 6 companies within different industries to get a better sense of which industry shows the most potential to have the best company culture while providing the same benefits as the most sought after companies. We know finding the right job and company can be tricky, so hopefully by doing these, it might be easier for future applicants to find the right job.

References

1. https://www.glassdoor.com/research/app/uploads/sites/2/2017/01/GD_Report_ReviewsMatterEmployerBrand_FINAL-2.pdf
2. https://www.gobarometer.com/blog/do-glassdoor-reviews-matter
3. https://b2b-assets.glassdoor.com/how-candidates-use-glassdoor.pdf
4. https://www.tutorialspoint.com/count-words-in-a-sentence-in-python-program
5. https://towardsdatascience.com/a-complete-exploratory-data-analysis-and-visualization-for-text-data-29fb1b96fb6a
6. https://towardsdatascience.com/generate-meaningful-word-clouds-in-python-5b85f5668eeb
7. https://towardsdatascience.com/a-complete-exploratory-data-analysis-and-visualization-for-text-data-29fb1b96fb6a
8. https://towardsdatascience.com/evaluating-machine-learning-classification-problems-in-python-5-1-metrics-that-matter-792c6faddf5