International Journal of Science and Research (IJSR)
ISSN: 2319-7064
SJIF (2022): 7.942

Classification of Glassdoor Pros and Cons into Pre-Defined Categories

Mahak[1], Aditya Raj Gupta[2], Deepti Buriya[3]

[1, 2]Indian Institute of Technology, Delhi, India
[3]Netaji Subhash University of Technology

Abstract: ***Glassdoor have huge amount of data related to an organisation ranging from job listing to employee reviews. It can contain several thousands of reviews for a sizeable company. It becomes imperative to understand and classify the humungous information into relevant categories to be able to drive meaningful insights. It is therefore of utmost importance to companies to keep analysing their employee reviews on Glassdoor.***

Keywords: Natural Language Processing, Glassdoor, Web Scraping, Multinomial Bayes Theorem

1. Introduction

A lot of job candidates turn to Glassdoor to get a better sense of the company culture and salary ranges. Glassdoor allows reviews to be posted anonymously by either Current or Former employees and thus encourages honesty and participation. More positive reviews and feedbacks from employees, invited greater number of applications from prospective employees. A candidate who is well informed is more likely to submit a relevant application and integrate well into your company. A recent survey conducted by Glassdoor showed that: "77% of adults would consider a company's culture before applying to work there". With this in mind, it makes perfect sense to really sell your business to people. It can also be hard for employers to attract high-caliber candidates if it has a low average score due to poor or critical Glassdoor reviews.

However, despite storing humungous amount of data about a company, there is a need to devise a mechanism to understand that valuable information and drive analysis useful for the company. There do exist a number of studies which focuses on driving results from the textual information. However, none of them so far focus on classification of reviews into broader categories. It can be primarily attributed to the unavailability to manually annotated reviews.

2. Literature Review

A comprehensive study done by Sanford institute describes a comprehensive methodology to predict the overall ratings by performing a Natural Language Processing Model on the Glassdoor reviews. They have used a Bi-LSTM model; they tried to predict the sentiment of the employees. Another research focused on performing empirical analysis of company culture using Glassdoor Data to estimate the impact of culture and employee satisfaction on performance.

Lastly, a research focused on modelling organizational culture with workplace experiences by performing an empirical analysis.

3. Methodology

3.1. Web Scraping

A typical page in Glassdoor review is illustrated in ref (fig: typical-review) Glassdoor website is rendered in JavaScript. Glassdoor provides restricted number of data points. It doesn't offer scraping of reviews. Thus, a typical web crawler was constructed.

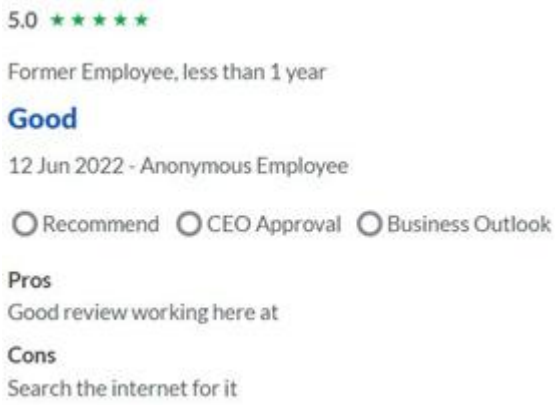

Figure 1: Typical Glassdoor review

Typically on the Glassdoor website, reviews are stored over multiple pages ranging up to few hundreds. Selenium library was used to scrape the reviews by building a python scripts which mimics human usage. The end product of the Python script is to obtain a data frame comprising of columns:

Date
Employee Status
Location
Job Title
Overall Ratings
Pros
Cons
Advice to Management

The idea is to use only the Pros and Cons columns which contain textual information. However, it can be extended to find correlation between other columns like Job Title, Location, etc. to the categories into which Pros and Cons can be classified to give a deeper analysis which can be useful got businesses.

The python script goes to the specified Glassdoor page which can belong to any organisation. It looks for reviews based on their similar fixed pattern and scrapes all the listing for the given page. It then follows the next button onto the next page of reviews and performs the same function. The process is repeated until the specified number of data points is met or the scraper reaches the end of reviews.

Since all the reviews for all organisation has a same structure. They belong to the same hierarchy of web page elements. We need to specify the address of the reviews from the hierarchy. To identify the address, we have to right-click and choose the option "Inspect Element". It opens a small window with the html content of the web page, with your element highlighted. The highlighted region represents the address of the element to be scraped. We need to scrape the entire review thus; we click on title of the review.

If the specified address is not present, in that case, for the data fields, it is a good practice to assign a "not found" value let's say-1.

A major challenge is bypassing the sign up prompt, since as soon as you click anywhere on the screen it opens the sign up window. This sign up windows forbids Selenium from accessing the web elements or clicking anywhere. This is the case every time selenium does a new search or go to the next page of job listings. In order to avoid this from happening, we click the X button to close it right after clicking on the topmost job review on the website. The chosen element is given the class name "selected". IT can be identified and clicked on by find element by class name ("selected").

Another challenge that can be witnessed is rejection of requests or throttling of connection. It means that sometimes due to multiple consecutive or simultaneous requests to access web elements on the Glassdoor website, it can result in slower connection or blocking the IP address.

One other point about Glassdoor's website is that, if you get too aggressive about get requests, Glassdoor will start rejecting or throttling your connections. In other words, if you were to do a get request to multiple links at once, it is likely that your IP address will get blocked, or you'll have a slow connection. Those things leave us no other option than writing a script Python scripts written imitates human's interaction to avoid Glassdoor's server and at the same time automating the entire process.

Another thing to note is that Selenium opens a new Chrome Driver every time the script is executed. However, we do have a provision to allow Selenium to scrape reviews in the backend without opening a new browser. This type of scraping is called "headless", and it is tricky to perform.

3.2. NLP Modeling

1) Data Cleaning

(a) Remove Stopwords-A lot of words in English language are used for the grammatical correctness but do not add a lot of meaning to the sentences. Such words are used on a day to day basis when humans interact. Additionally, there might be few words which are not required for the business case given in hand. Such words are called stop words and are deleted from the data. The NLTK package has a predefined set of stopwords for multiple languages; however, we will only be dealing with English.

(b) Converting to lower case: This step is performed to ensure uniformity among all the words

(c) Stemming-IT produces morphological variants of a root/base word. It is somewhat a crude method for categorizing related qwords. For example, "boat" would be stem for "boats". "boaters", "boating", etc. Since English is a morphologically rich language, such technique may not function properly. Examples of commonly used stemmers include:

i. Porter Stemmer-It follows Porter's algorithm in which there is five phases of word reduction, each with its own set of mapping.
ii. Snowball Stemmer-It shows slight improvement over the Porter Stemmer in terms of speed and logic. To illustrate their use case, let us take example of the word "fairly", a Porter Stemmer gives the result as "fairly" while a Snowball Stemmer produces "fair"

(d) Lemmatization-It takes into account language's full vocabulary to apply a morphological analysis and just focus on word reduction. It considers the surrounding text to determine a word's part of speech. For example, "ran" (past tense) is reduced to "run" (present tense) Lemmatizers does give better results, since they take into consideration the richness of structure of the language. At the same time, they are more time consuming as words are filtered from a corpus and further their parts of speech are identified. At the same time, stemmer depends on the situation; if speed is prioritize then stemmers are preferred whereas if accuracy is prioritise then we choose lemmatizer.

(e) Removal of regular expressions: Regular expression helps to identify and to remove multiple patterns like emails, hashtag, underscore, etc. which are not required in the text.

(f) Parts-of-Speech (POS) tagging: This is used to identify the parts of speech. Based on the use case, one can decide to omit or keep them.

(g) Named-Entity-Recognition (NER): This is used to categorize the different groups which include names, occupation, places, etc.

2) Tokenization-It is a process of breaking down a sentence or a phase into smaller constituents words called tokens. There are multiple ways to tokenize a word:

(a) Using split () function: Returns list of words after breaking any sentence by the specified separator. By default, a separator is a space.

(b) Using Regular expression: Returns list based on the regular expressions specified

(c) Using NLTK: There are different types of tokenizers under NLTK package like word tokenizer (word tokenize), regex tokenizer (RegexpTokenizer), whitespace tokenizer (WhitespaceTokenizer) etc.

3. Vectorization/ Word embeddings-Since the computers cannot understand English, it is imperative to convert the words into numbers, so that the computer can decipher it. Vectorization helps to map the words to a vector of real numbers, which further helps into predictions. There are many ways in which vectorization can be performed.

   (a) Count Vectorization-It simply counts the number of times a particular words is occurring in the document.

   (b) N gram vectorization-It counts the number of times n consecutive words are occurring. N can take any value from 2 (bi gram), 3 (trigram), etc. Since in many a situation, simply counting the number of times a particular word is occurring doesn't give a wholesome picture of the sentiments being convey, hence n-gram vectorization comes into play.

   (c) TF-IDF-Term frequency Inverse document frequency (TF-IDF) provides an overall weightage of a word in the document. The words appearing less frequent are given higher weights. This is done to ensure words used more frequently are penalized due to the higher occurrences. Each of the vectorization technique is used to give a count or weight to a word or group of words. However, to understand each word's context and to identify the content, one vector per word is much more appropriate. Word2Vec gives a vector for each word from the given document which gives a more contextual understanding. Glove is another pre trained vectorization technique to include the local context which wasn't provided with the Word2Vec form of vectorization. FastText is another popular technique which works for rare words or Out of Vocabulary (OOV).

4. Model Development The aim is to produce a count based or weight based (TF-IDF) matrix and the dependent variable (label) to develop the model. We need to split the data into train and test set. Any classification models including logistic regression, random forest, support vector machines or any deep learning models like RNN, LSTM or state-of-art model like BERT, GPT3 can be used to predict the label. To estimate the results, we can use accuracy ROC, Recall, F1-score and validate the results.

Preparing Dataset

The dataset was created of 30336 reviews by the Web Scraping method. Each of the review has a Pros and Cons sections and were assigned a category from Pros and Cons respectively. The number of reviews in each of Pros category included:

People	3439
Salary	2111
Environment	1030
Culture	533
Work Life Balance	191

The number of reviews in each of the Cons category included:

Work Life Balance	3515
Management	2521
Salary	581
Stress	402
Bureaucracy	285

In the NLP modeling part, the same steps described in the section were followed

4.1 Multinomial Naive Bayes

After processing the NLP modeling on the data, we used the multinomial Na¨ıve Bayes which is a specialized version of Naive Bayes designed more for text documents were used. It guesses the tag/ category of a text, using the Bayes theorem. It calculates each tag's likelihood for a given sample and outputs the tag with the greatest chance. The Naïve Bayes classifier is made up of a number of algorithms with each algorithm classifying each feature which is unrelated to any other feature. Any feature's inclusion or exclusion is unrelated to other features.

Bayes theorem, calculates the probability of an event occurring based on the prior knowledge of occurrence of an event. The formula can be written as:

P (A—B) = P (A) * P (B—A)/P (B)

Where

P (B) = prior probability of B
P (A) = prior probability of class A
P (B—A) = occurrence of predictor B given class A probability

This formula helps in calculating the probability of the tags in the text.

Results and Discussion

The following tables shows the result for Pros Categories: Multinomial Na¨ıve Bayes

	precision	recall	f1-score	support
Culture	0.90	0.12	0.21	157
Environment	0.89	0.53	0.66	296
People	0.73	0.97	0.83	1035
Salary	0.83	0.79	0.81	643
Work Life Balance	0.83	0.08	0.15	61
Accuracy			0.77	2192
Macro avg	0.84	0.50	0.53	2192
Weighted avg	0.80	0.77	0.74	2192

K Neighbours Classifier

	precision	recall	f1-score	support
Culture	0.68	0.58	0.63	157
Environment	0.79	0.80	0.79	296
People	0.84	0.94	0.88	1035
Salary	0.87	0.73	0.80	643
Work Life Balance	0.66	0.72	0.69	61
Accuracy			0.83	2192
Macro avg	0.77	0.75	0.76	2192
Weighted avg	0.83	0.83	0.82	2192

The following table shows the result for Cons Categories: Multinomial Na¨ıve Bayes

	precision	recall	f1-score	support
Bureaucracy	1.00	0.16	0.28	85
Management	0.67	0.87	0.76	765
Salary	0.95	0.18	0.31	190
Stress	0.80	0.13	0.22	123
Work Life Balance	0.76	0.83	0.79	1029
Accuracy			0.73	2192
Macro avg	0.84	0.44	0.47	2192
Weighted avg	0.76	0.73	0.69	2192

K Neighbours Classifier

	precision	recall	f1-score	support
Bureaucracy	0.52	0.16	0.25	85
Management	0.78	0.51	0.62	765
Salary	0.58	0.41	0.48	190
Stress	0.5	0.26	0.34	123
Work Life Balance	0.64	0.92	0.75	1029
Accuracy			0.66	2192
Macro avg	0.6	0.45	0.49	2192
Weighted avg	0.67	0.66	0.64	2192

Conclusion

Natural Language Processing modelling was performed on annotated Glassdoor reviews to determine the precision, recall, F1 score and accuracy across Multinomial Na¨ıve Bayes and K Neighbours Classifier. The same process was repeated for Pros as well as Cons. The best accuracy value obtained was 83% (for Pros by K Neighbours Classifier) and 73% (for Cons by Multinomial Na¨ıve Bayes)

References

[1] https://mersakarya. medium.com/selenium-tutorial-scraping-glassdoor-com-in-10-minutes-3d0915c6d905
[2] https://towardsdatascience.com/basic-concepts-of-natural language-processing-nlp-models-and-python-mplementation-88a589ce1fc0
[3] Predicting Company Ratings through Glassdoor Review, Fabian Frederik Frank, Tyler Emerson Whittle
[4] An Empirical Analysis of Company Culture: Using Glassdoor Data to Measure the Impact of Culture and Employee Satisfaction on Performance, Linnea H. R. Uyeno [2019]
[5] Modeling Organizational Culture with Workplace Experiences Shared on Glassdoor, Vedant Das Swain, 88 Koustuv Saha, Manikanta D. Reddy, Hemang Rajvanshy, Gregory D. Abowd, Munmun De Choudhury [2020]