Predicting Company Ratings through Glassdoor Reviews

Fabian Frederik Frank
Department of Management Science and Engineering
Stanford University
Stanford, CA 94305
fabfrank@stanford.edu

Tyler Emerson Whittle
Department of Management Science and Engineering
Stanford University
Stanford, CA 94305
twhittle@stanford.edu

Abstract

Managing employees is perhaps the most critical task within an organization. Recent work in economics and finance has shown that employee perceptions of culture, managerial integrity, and other intangible factors are correlated with financial performance. Thus, it is critical that managers develop and curate skills to effectively interface employees. In this paper, we explore models that predict employee sentiment based on text in employee review data from a popular career website, Glassdoor.com. By developing a high-fidelity model that can accurately predict employee sentiment from text, we provide organizations with a new avenue to examine unstructured text generated by their employees, for example internal quarterly reviews.

1 Introduction

Reviews are a very common way of giving feedback to entities and helps the reader to get a sense of the quality. Such reviews are used in various contexts: For companies, books, films, etc. However, the qualitative nature of a review oftentimes leave an unsatisfying feeling of not being able to identify how to compare different entities just based on the review. A translation of a qualitative review into a quantitative metric can close this gap. Websites like Glassdoor.com already offer the option to also add a quantitative (star ratings out of 5) rating. The goal of this project is to predict those quantitative ratings by applying an advanced sentiment analysis to the qualitative reviews based on reviews and ratings from Glassdoor. Such a model can then potentially also be applied to review platforms where there is no quantitative component and therefore enrich the review and enable comparisons between different reviews.

Example Input-Output behavior in Figure 1.

2 Background / Related Work

Understanding employee perception of the firm is a critical task for managers (cite). Research in strategy and finance has started to look at the importance that employee satisfaction can have on firm level outcomes. Erhard, Jensen, and Zaffron (2007) show that maintaining a culture of integrity

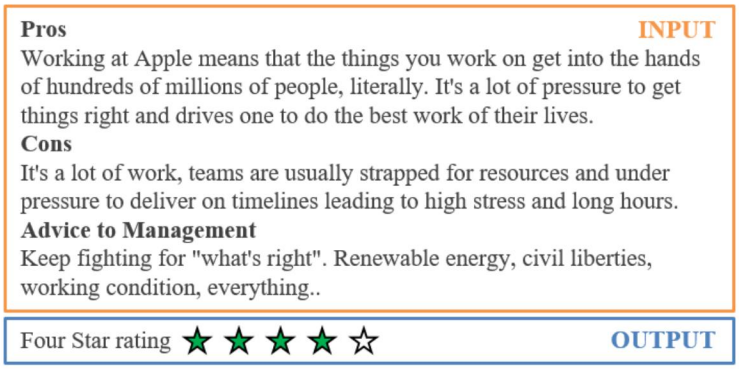

Figure 1: Example input-output behavior

can have short-term costs, but long-term benefits to a firm. In another study, Edmans (2012) looks at whether the stock market fully values intangibles. Using the "100 Companies to Work for in America", Edmans finds that that a portfolio of these companies earned an annual four-factor alpha of 3.5% from 1984 to 2009, 2.1% above the industry benchmark. Lastly, Guiso, Sapienza, and Zingales (2015) study which elements of corporate culture are related to a firm's performance. They use data from the Greatest Places to Work Institute and find that managerial integrity and managerial ethics as rated by employees both have a positive correlation with Tobin's Q.

A common problem in Natural Language Processing (NLP) is predicting the sentiment of a paragraph. In this report, we will look at employee reviews from Glassdoor.com. Employees write about their employer and then provide a star rating from one to five. This star rating will serve as a good proxy for the sentiment of the review. Glassdoor stands to learn about relevant employee satisfaction metrics after analyzing sentiment across many firms. Employers can also keep a finger on the pulse of the firm by monitoring the sentiment that appears in reviews. By first training models on Glassdoor data, employers could then apply the same models to unstructured reviews and feedback provided by employees. This would allow firms to quantitatively analyze and evaluate text in internal reviews provided by employees.

In order to tackle this problem, we take an iterative approach to try and build a model that can correctly classify sentiment. We begin with a Naïve Bayes classifier that analyzes one-hot encoded reviews. Next, we implement simple 1-ReLU and 2-ReLU networks and classify reviews that are encoded using bag-of-words integer encoding, word2vec, and GloVe. Lastly, we use a Long-Short Term Memory (LSTM) Recurrent Neural Network to build a model based on both the one-hot-encoded and GloVe encoded reviews.

3 Approach

3.1 Data

Glassdoor is a jobs and recruiting site that holds a database of millions of company reviews, CEO approval ratings, salary reports, interview reviews, and more. Before reviews are posted, employees must verify that they currently or previously worked at the listed employer. Along with this, review postings are completely anonymous and either unsolicited or contributed by employees searching for jobs in return for unlimited site access. These aspects of Glassdoor help ensure the authenticity of the reviews and reduce the potential for reviewer bias.

We obtained the dataset for this project by scraping the reviews of all public companies on Glassdoor.com. After extensive work developing the script to scrape Glassdoor, we ran our script for one week to obtain 1,015,163 reviews. Each of these reviews contains text sections for employees to fill in pros, cons, and advice to management. For this paper, we choose to focus only on the pros and cons when predicting the star rating. This is because employees tend to directly describe the state of the firm in the pros and cons, whereas in the advice to management they write how they would like

Table 1: Summary of the Glassdoor.com dataset

ASPECT	QUANTITY
Reviews	1,015,163
Words	100,000,000+
Companies	4,183

the firm to be. Thus, we remove this potentially confounding data from our analyses. Descriptive statistics on the number of companies, reviews, and words in our dataset can be found in Table 1.

3.2 Data Pre-Processing

As we said, we extract only the pros and cons from the Glassdoor reviews. For our analysis, we further limit the pros and cons to 100 words. The average number of words in each section is 67, and the median is 49. Thus, this cutoff will capture the true sentiment of the majority of reviews while keeping the data at a manageable size. We use the Natural Language Toolkit (nltk) package in python to word tokenize the reviews, convert the words to lower case, and make this cut. If the combined text of the pros and cons is less than 200 words, we pad the reviews with a special word, 'PAAAAAD', until they reach this length. We then integer tokenize the words using the top 10,000 words contained in the reviews used for training. Any words that are outside of the top 10,000 words are replaced with a predetermined integer that represents unknown characters. This provides our first encoding of reviews: integer encoding using bag of words. The integer that represents a word corresponds to its position in the overall count of words. Consider the case where 'I' is the 5th most common word, 'am' is the 37th most common word, and 'Sam' is the 7892th most common word. Then the review ['I am Sam'] would be converted to [5, 37, 7892, 0, 0, . . . , 0] with 197 zeros representing the padding character. These integers are then converted to one-hot vectors in our most basic form of embedding.

For many of our models we utilize word to vector dictionaries that convert each word in the review to a vector representation. Specifically, we employ pre-trained word vectors using word2vec and GloVe. The word2vec pre-trained embeddings come from a model trained on a Google News dataset. This dataset has 300-dimnesional vectors for 3 million words and phrases. The GloVe pre-trained embeddings come from a model trained on 42 billion tokens encountered by Common Crawl, a program designed to crawl the web and extract text. This dataset also has 300-dimensional vectors for 1.9 million words and phrases. We then write a function that maps the integer-encoded reviews to their respective word2vec and GloVe embeddings. In the case that the word does not have a pre-trained vector representation, we initialize a vector for unknown words with a small but non-zero uniform vector. This allows the unknown words to effectively be differentiated from the padding vectors (which is just a zero vector). After encoding then, each word will have been mapped to a 300-length vector. A review with word2vec or GloVe vectors thus has a length of 60,000.

As a final step in data preprocessing, we split our reviews into three subsections of training (100k), dev (10k), and test (10k). We select this random subset of the reviews to allow for faster training times, thus allowing us to explore more model configurations.

3.3 Frameworks / Libraries Used

Our implementation of the models that will be discussed in the next section use three different frameworks in python. The baseline Naïve Bayes classifier uses scikit-learn (sklearn) to perform classification between the one-hot-encoded reviews. The simple 1-ReLU and 2-ReLU models use Tensorflow. Finally, the LSTM model uses keras.

3.4 Model Configuration

In this section we will describe the two deep learning models we used to evaluate the data: N-ReLU and LSTM.

The 1-ReLU and 2-ReLU models are built on the same principle. Below we provide an illustration of the 2-ReLU network. We begin by feeding the integer encoded reviews into the embedding layer. If word2vec or GloVe is being used, then we apply the embeddings in this step. Otherwise we convert the reviews to one-hot vectors. Next, we feed the embedded words into a ReLU with a hidden layer size of 200. Nodes are then dropped with a 10% probability. This process is then repeated once again in the second ReLU layer to generate the probability distributions across the five scores. The output from the second ReLU layer is then fed into a softmax function which calculates the cross-entropy loss of the 2-layer neural network. An outline of the 2-ReLU model can be seen in Figure 2.

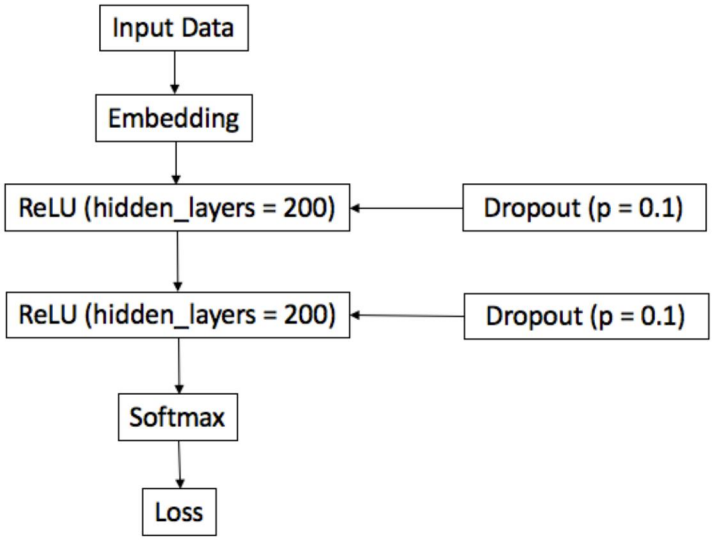

Figure 2: Model configuration of the models with 2-ReLU

The LSTM model starts the same as the ReLU model. It first takes the integer-encoded reviews and then converts them to a word embedding. These embeddings are then fed into a LSTM with 64 units. The LSTM used is bidirectional, to ensure that both directions are considered. Therefore, the overall LSTM layer has 128 units. Afterwards,a dropout is added, through which nodes are dropped with a 10% probability. The model is trained on categorical cross-entropy loss via a softmax layer, similar to above. Adam is used as the optimizer. An outline of the LSTM model can be seen in Figure 3.

4 Experiments

In this section, we will cover the experiments we ran using the aforementioned Glassdoor data and models. We will first discuss the evaluation metrics we used, and then cover the performance of our model on the dev set.

4.1 Evaluation Metric

To analyze the performance of our models, we use two different performance metrics: accuracy and macro-f1 score. We define accuracy as the percentage of reviews that are correctly predicted across the entire sample. The macro-f1 involves calculating the f1 scores at the label level. For each label (Glassdoor score from one to five), we calculate the precision as tp/(tp + fp) and the recall as tp/(tp + fn). The label-level f1 score is then calculated by 2*p*r/(p+r). Finally, the macro-f1 score is calculated by an unweighted average of the label-level f1 scores. We chose to use the macro-f1 score over the micro-f1 score because scores are clustered around 4 stars. Since we want our model to be able to accurately predict the 1 and 2-star reviews, the macro-f1 score did not give additional bias to correctly predicting 4-star reviews even though they appeared most frequently.

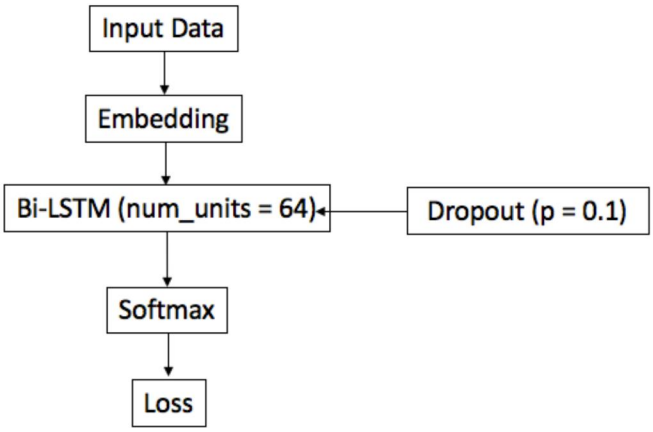

Figure 3: Model configuration of the models with BiLSTM

4.2 Results

Below, we cover the best results obtained on our dev models and the parameters that went into the respective models.

4.2.1 Baseline: Naive Bayes Classifier

The baseline with a Naive Bayes classifier was created to generate baseline values for accuracy and macro-f1 that came from a method outside of deep learning. In order to do this, we utilized python's nltk package. The preprocessing was the same as the deep learning models. The Naive Bayes classifier then uses Bayes Law to calculate which of the five star categories the review belongs to. Because this was a baseline calculation, we did not tamper significantly with the model to try to improve performance.

4.2.2 Models with ReLU

For the experiments using the N-ReLU, we used three different embeddings. We started with an one-hot embedding of the integer representation of the words, and continued with more advanced models, namely a word2vec and a GloVe approach.
From experimentation, it turned out that the number of features per word performed best with only 1 feature per word. Also, we decided to cut the review length to 200 words, since there were only a few very long reviews, which did not add significant value in the sense of their length.
For the ReLUs, we used 200 hidden units in both layers, and a batch size of 2048 and a learning rate of 0.001. The embedding size decision was made based on GloVe and therefore was 300 and the dropout is 0.9.
We also experimented with different dropout rates, hidden layer sizes and learning rates. The results are robust to all changes.

4.2.3 Models with Bi-LSTM

In the models with the Bi-LSTM, we experimented with the number of units in the LSTM, with the directionality of the LSTM, with the numbers of additional ReLUs, the dropout, the loss functions, the optimizers. We found that adding additional ReLU functions and dropouts only made the model more complex and therefore less interpretable, but did not contribute to the performance of the model. Therefore we decided to go with a very simple Bi-LSTM model, since the bi-directionality was proving to be effective. 64 units proved to be a good approach. While a mean squared error loss function is generally applicable, since the difference between the star ratings are interpretable, we found that a categorical cross-entropy loss works better. This is due to the fact that we really

Table 2: Experiment Results on Dev Dataset (size: 10,000 reviews)

Model	Word Embedding	Accuracy	Macro-F1
Naive Bayes	one-hot	34.4 %	0.33
1-ReLU	one-hot	22.7 %	0.22
2-ReLU	one-hot	21.4 %	0.20
1-ReLU	word2vec	28.7 %	0.28
2-ReLU	word2vec	29.1 %	0.28
1-ReLU	GloVe	28.0 %	0.27
2-ReLU	GloVe	29.3 %	0.28
Bi-LSTM	one-hot	40.5 %	0.40
Bi-LSTM	GloVe	**46.4 %**	**0.44**

care about getting the review right, and we do not value a two star prediction on a one star review more than a five star prediction. Regarding optimizers, we tried SGD with various learning rates and momentum, but found that Adam optimizer worked the best.

4.3 Error Analysis

Besides the quantitative results presented in Section 4.2, we also took a look at individual reviews and the predictions we made on those to learn from mistakes and also from good examples. We looked through around 50 reviews and their predictions and the ground truth to figure out what was going wrong and what was going well.
Generally, longer reviews were harder. The reviewer usually provided very rich information in those, however, the reviewer might be biased by the point which is the most important to him or her in the quantitative star-rating. Even though the overall review might read very positively, if there was a small negative thing which really bothered the reviewer, the reviewer might have still given only a three star rating, which the model would then not pick up properly.

On the other hand, for short reviews, the models usually got the right sense, but were off by one star, since it was hard to get the fine-tuned sentiment from just a few words. For example, a review with only a few positive words might be a four or five star rating.
For some examples, see Figure 4.

Review	Prediction	Ground Truth	Error Analysis
Pros 1. Work from Home 2. Flexible timing option. 3. Employee friendly work environment **Cons** Nothing is there to say as cons.	5	4	This review is solely positive. There is not the slightest negative aspect, so a 5 star-rating would be a reasonable quantitative support of this review. It is unclear why the reviewer only gave 4 stars.
Pros Flexible working place. Nice colleague, supportive and friendly. Great HR team, helpful, competent. Easy to reach agreements with management, ready to negotiate and support **Cons** Low level of skills and business acumen in many employees. Company is changing drastically. Few years ago even a hairdresser could become an IT specialist if they spoke an extra language except English. But nowadays if you want a good job, well paid you need to be very skilled. Low-skilled jobs are declining. If you get to higher band the compensation is beyond competitive. See the upline managers, no one or very few have left the company which there is an indication that higher role are well paid	4	4	This is a very long and elaborate review, and therefore not easy to get the star-rating exactly right. The model is doing a good job and gets the overall sense, taking the information from Pros and Cons in and weighting in properly.
Pros No other organization can be the culture what we have in IBM **Cons** One gets addicted to this culture ☺	3	4	There is a lot of negative language used in this review. Particularly the use of words like "no other organization" and "addicted" pulled the prediction down, even though the review was actually meant positively

Figure 4: Overview of error analysis. These three reviews are representative examples of things that were responsible for predictions being accurate or not accurate.

5 Summary & Challenges

From the results above, we see that the bidirectional-LSTM with GloVe word embeddings performs the best out of all of the models with an accuracy of 46.4% and a macro-f1 score of 0.44. Even without the GloVe embeddings however, the LSTM significantly outperforms the best ReLU architecture with word embeddings. The one-hot-encoded bi-LSTM achieves an accuracy of 40.5% and a macro-f1 score of 0.40 whereas the best 2-ReLU achieves only an accuracy of 29.3% and a macro-f1 score of 0.28. The LSTM has an accuracy that is 17.1% greater than that of the best ReLU architecture. It also performs significantly better than the Naïve Bayes baseline.

This paper presented many challenges. Obtaining the Glassdoor data proved more difficult than expected. It took a significant amount of time to work out bugs and scrape the website for the reviews. Implementing tensorflow code that functioned with our data also proved more difficult than expected. Because our data was raw and unprocessed, we had to spend significant time massaging it to encode it as integers and then create the mapping to word2vec and GloVe. Given time constraints, we chose to use the pre-trained word2vec and GloVe embeddings. However, with more time, we would have liked to update the embeddings to better reflect Glassdoor language tendencies.

6 Conclusion & Future Work

In this paper, we have shown that models can be trained on employee reviews to accurately assess a quantitative sentiment rating based on qualitative text data. While this has been developed before in other contexts, such as movie reviews, this context stands to offer unique insights to managers. Using our Bi-LSTM model, we can accurately predict the sentiment of a review on a 1-5 star scale 46.4% of the time. This model shows initial signs that Glassdoor reviews have the potential to help mangers assess internal reviews. If our model were developed further to increase the accuracy, top managers could use the model to generate a consistent quantitative rating of internal employee reviews.

To improve the usefulness of such a tool, an additional model using attention could be developed. By noting 3-5 words with the highest attention and associating them with the reviews, we could provide additional functionality. These key words could help managers pinpoint specific topics that are either bright spots in good reviews or pain points in bad reviews. In doing so, managers could then determine a strategic plan to bolster elements employees are satisfied with while improving upon elements that leave employees less satisfied.

7 References

[1] Erhard, W, MC Jensen, and S Zaffron. 2007. Integrity: Where Leadership Begins.

[2] Edmans, A. 2012. "The Link Between Job Satisfaction and Firm Value, With Implications for Corporate Social Responsibility." Academy of Management Perspectives 26 (4). Academy of Management: 1–19.

[3] Gers, F. A., Schmidhuber, J., & Cummins, F. (1999). Learning to forget: Continual prediction with LSTM.

[4] Guiso, Luigi, Paola Sapienza, and Luigi Zingales. 2015. "The Value of Corporate Culture." Journal of Financial Economics 117 (1): 60–76.

[5] Pennington, J., Socher, R., & Manning, C. (2014). Glove: Global vectors for word representation. InProceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)(pp. 1532-1543).

[6] Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A., & Potts, C. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. InProceedings of the 2013 conference on empirical methods in natural language processing(pp. 1631-1642).

[7] Sokolova, Marina, and Guy Lapalme. "A systematic analysis of performance measures for classification tasks." Information Processing & Management 45, no. 4 (2009): 427-437.