Capturing a firm's diversity and inclusion culture from employee communication
A pilot study for a semi-supervised machine learning approach to textual analyses

Peter Schäfer and Philipp Richter

TUD Dresden University of Technology
Chair of Business Administration, esp. Management Accounting and Control
peter.schaefer@tu-dresden.de

August 2024

1. Introduction and goal

Firms increasingly care about diversity and inclusion. Are all employees, their ideas, and contributions treated and valued equally? Extensive research tries to understand the antecedents and effects of diversity and inclusion. Surprisingly, most research captures the strength of a firm's diversity and inclusion by demographic diversity. However, demographics ignore many dimensions of a diverse and inclusive corporate culture. Minorities may be represented but still not have the opportunity to promote their ideas, or firms may ignore the ideas of some employees even if they do not differ from the majority in demographic factors.

To foster research on diversity and inclusion with a more comprehensive perspective, we suggest a measure that uses a machine-learning algorithm for text analysis and applies it to employee reviews and management communications. Applying this measure, we will provide a database that includes the diversity and inclusion score calculated by the proposed measure for thousands of firms worldwide over multiple years. We base the measure on three dimensions. First, we capture employees' perspectives on diversity and inclusion. To this end, we collect employee reviews from the employer rating portal Glassdoor and utilize a semi-supervised machine-learning algorithm to learn about employees' perspectives on a firm's diversity and inclusion culture. Second, we capture how strongly employee satisfaction (captured by the employee ratings on Glassdoor) depends on their background and position within the firm. Third, we capture the tone of the top management about diversity and inclusion. To this end, we apply the above algorithm to management communication during earnings calls. We will calculate our measure for a panel of worldwide firms from 2009 to 2023.

In this pilot study, we test the most data- and processing-intense part of the proposed approach for feasibility and assess its performance, that is, scraping employee reviews and analyzing them with natural-language processing (NLP) and a semi-supervised machine-learning algorithm. First, we scrape employee reviews from the Glassdoor website to forecast the time and cost required. Second, we test and assess the performance of the NLP tools to prepare the Glassdoor reviews for the semi-supervised

learning algorithm. Finally, we run the machine-learning algorithm using the employee reviews and validate the calculated diversity and inclusion scores.

The results from our pilot study provide a general proof of principle. We implemented all required steps successfully in Python, and the procedure has provided meaningful results. For ten firms in our pilot sample, we calculate the suggested diversity and inclusion score from 2008 to 2024. The measure has considerable variance across firms and time, i.e., the average score is 2.11 with a standard deviation of 0.68 and a maximum (minimum) of 3.71 (1.00). As a validity check, we calculated the correlation with diversity and inclusion ratings from Glassdoor, which have been available since 2020. Both measures are significantly correlated, with a correlation coefficient of 0.61. These results lend some credibility to our procedure, even though we note that the sample for this test is still small and further validation checks will be required in the final sample.

The total time required for the scraping process was approximately 4.2 hours (scraping 5,000 pages with an average of twelve seconds per page, scraping four pages in parallel). For the final project, we expect 5m pages to be scraped. We expect the total process to take 420 hours (scraping 100 pages in parallel, with an increased duration of 30 seconds per page; 4.2 hours * 30 sec per page / 12 sec per page * 5m pages / 5k pages * 4 / 100 pages in parallel). We expect costs of $3,720 and plan six months for the full scraping process, simultaneously scraping multiple pages (see detailed calculations in Section 3 below). Regarding the computational requirements for the NLP and machine-learning algorithm, we found that running the NLP steps will not be feasible on a personal computer (8 hours for the 10 MB of text that we have processed in the pilot) but forecast only about an hour when we utilize the TU Dresden high-performance computing service.

In the remainder of this pilot study, we first describe the key steps of the initially defined test procedure (Section 2). Then, in Section 3, we summarize the results of this test procedure and describe the implications for the final study.

2. Procedure

Our test focuses on the most resource-intensive process: scraping and processing the employee reviews. It includes a test of the proposed machine-learning algorithm, Word2Vec, and the planned procedure to calculate the first component of the diversity and inclusion measure, that is, employees' assessment in their reviews on the employer rating portal Glassdoor. We implement all data collection and processing steps that will be required to obtain the first component of the proposed measure for a test sample of ten firms. Specifically, we perform the following steps:

- Step 1: *Select ten sample firms from different countries and industries (see Appendix 1)*

  To test the proposed procedure, we select ten firms from different countries and various industries. For these firms, we run the full procedure to obtain the first component of the diversity and inclusion measure.

- Step 2: *Implement the scraping tool in Python and scrape the first 500 employee-review pages for the chosen firms*

  First, we compare different web scraping platforms that provide APIs allowing users to automate web scraping tasks, including handling dynamic content, browser automation, and collecting large amounts of data from the web efficiently and legally. For cost and performance reasons, we chose to proceed with the API by ScrapFly.

  Second, we implement Python code to collect employee reviews from Glassdoor. The input is a list of firm names and links to the corresponding Glassdoor review pages. The output is a folder with one .json file per firm that includes all employee review texts and ratings of the first 500 review pages.

  Our program shall be able to scrape multiple pages simultaneously (due to capacity restrictions of the standard ScrapFly subscription, a maximum of four pages simultaneously). In order to forecast the total time required for the final project, the program shall calculate the average time it took to scrape one page.

- Step 3: *Implement an NLP tool in Python to prepare the reviews for the machine-learning algorithm*

  We implement Python code to extract employee reviews from the .json files and collect all review texts in one text file. The code shall then prepare the texts for machine learning using NLP. To parse and prepare the texts, we implement the Stanza package for NLP (see https://stanfordnlp.github.io/stanza/).

  Specifically, we first use sentence segmentation, tokenization (the Word2Vec model operates at the sentence levels and requires tokenization), lemmatization to return words to their base forms, named entity recognition to substitute named entities (firms, places, persons, etc.) by prespecified tags, and apply dependency parsing to learn grammatical relationships within a sentence and identify multi-word expressions as well as compounds. To assess differences in the performance of different NLP packages, we also implement *Stacy* as an alternative to the *Stanza* package for NLP.

- Step 4: *Implement the semi-supervised machine-learning algorithm in Python (Word2Vec, word embedding)*

  We implement the Word2Vec model (Mikolov et al., 2013) in Python and choose seed words to run the algorithm. The inputs to this step are the preprocessed employee review texts (from Step 3) and the list of seed words. The output is a list of key terms that the algorithm calculates to be similar to the seed words.

- Step 5: *Generate the diversity and inclusion dictionary and calculate the diversity and inclusion scores*

  We manually go through the suggested word lists and choose the words that we consider to capture best the strength of a firm's diversity and inclusion culture to create a diversity and inclusion dictionary. We implement a Python algorithm that counts the relative frequency of the words from the dictionary in the employee reviews for each firm year. This frequency will be the first component of our diversity and inclusion measure.

3. Results and learnings

Steps 1 & 2: *Scraping Glassdoor reviews*

Process: We have chosen ten firms from different countries and industries for the pilot study. Appendix 1 provides an overview. For these firms, we have collected the links to the corresponding Glassdoor review pages. As an API for scraping, we chose ScrapFly because it is said to work reliably and comes at reasonable costs.

We have then implemented and run a scraping program in Python. The code collects four pages simultaneously and provides users with the average time it takes to collect one page. Also, it is robust to errors in the scraping processes (e.g., blocking or lost connections) and dynamically pauses the process after multiple failure returns. The file 01_scraping.py includes our scraping program.

Observations: Our scraping algorithm utilizing the ScrapFly API successfully scraped 99.5 percent of the first 500 pages of employee reviews for the ten firms. The average time taken was approximately twelve seconds per scraped page of employee reviews, using four searches concurrently. The algorithm uses, on average, 7 API credits per scraped page.

Learnings: For the final project, the total sample will comprise 5,000 firms. We forecast an average of 1,000 pages of reviews to be scraped. From the experience in the pilot study, we have forecast the following costs and time requirements for the final project:

- Costs: Utilizing the ScrapFly Enterprise subscription (currently $500 per month) includes 5.5m API credits. We plan six months for data collection such that the subscription costs a total of $3,000 and includes 33m API credits. Then, we need an additional 1m API credits per month, costing $120, such that total scraping costs are projected to be $3,720.
- Required processing time: To finish the scraping in a reasonable time, we extend the number of concurrent requests. However, this will increase the block rate and the time per page. We will utilize 100 concurrent requests (maximum no. of concurrent requests with the ScrapFly Enterprise subscription) and forecast an average scraping duration per page of 30 seconds. Thus,

the total scraping time will be approximately 420 hours, which we can manage in six months, including code adjustments, preparation, and potential rework.

Step 3: *Applying NLP to prepare the texts for the machine-learning algorithm*

Process: We have implemented a Python program that extracts the individual scraped reviews from the .json files. The corresponding code is in 02_process_json_files.py. We have then implemented the preprocessing steps using NLP in a Python program. The file 03_create_di_measure_gdreviews.py includes the preprocessing code.

Observations: We have tested two packages for NLP (Stanza and Spacy). Stanza provides more advanced language processing, but Spacy is much quicker. All processed employee reviews (from 5,000 employee review pages) amount to 10 MB of text. Stanza requires approximately 8 hours to process the full text on a standard laptop (Intel vPro i7, 32 GB RAM), while Spacy requires only about one hour. With both packages, the results were sufficiently good to be entered into the Word2Vec model in the next steps. However, the suggested dictionaries appeared better applicable when we used Stanza for NLP.

Learnings: The complexity of the applied models is approximately linear in the text length. Given that 5,000 employee review pages amount to 10 MB, we will have to process 10 GB of text for the full sample. Preprocessing this amount of text will not be possible on a standard laptop. We will access the high-performance computing facilities of TU Dresden. The *Barnard* high-performance computer by Atos/Eviden provides 104 cores and 512 GP RAM per core. After consultation with the TU Dresden Center for Information Services and High-Performance Computing, the high-performance computer should perform the full job with 10 GB in about an hour.

Step 4 & 5: *Applying the semi-supervised learning algorithm and calculating the diversity and inclusion score*

Process: We have implemented a Python program that utilizes the Word2Vec model to suggest a list of key terms for a diversity and inclusion dictionary. The procedure starts with seed words. For our pilot study, we choose the following list of seed words: “diversity,” “inclusion,” “equal_opportunity,” “belonging,” “engagement,” “representation,” “inclusive_culture,” and “diverse_skills.” The word embedding algorithm then captures the meaning of words in the employee reviews using a numeric vector. The algorithm calculates the distance between the vectors of two words to assess the relationship between these words and suggests a list of words that are similar to the seed words. For our test run, we follow the recommendation by Li et al. (2021) and use the following parameter settings: Window size = 5; dimension of the word vector = 300; number of iterations over the corpus = 20; min word count = 5; training method = negative sampling. The file 03_create_di_measure_gdreviews.py includes the

program that performs these steps. It applies the Word2Vec model and provides a list of keywords from this algorithm. The file 03_create_di_measure_gdreviews.py includes the code that runs the machine learning algorithm. The file 04_processed_gd_reviews / 03_text_processing_steps / 06_di_dictionary.txt includes the list of words that the algorithm considers similar to our seed words.

Next, we manually review the list of words to generate the diversity and inclusion dictionary (saved in 04_processed_gd_reviews/03_text_processing_steps/06_di_dictionary_revised.txt). We then implement a Python program that counts the relative frequencies of the words from the dictionary in the employee reviews and calculates the firm-year measure of diversity and inclusion culture from this word count. The file 07_di_measures.csv includes the firm-year diversity and inclusion measure generated through the whole process.

Observations: The running time of the Word2Vec model is negligible (less than a minute). Even though the training text is only 0.1 % of the length of the text in the main study, the output from the machine-learning algorithm (i.e., the list of proposed words for the diversity and inclusion dictionary that the algorithm detects as similar to the seed words) already includes many terms that we consider beneficial to capture the strength of a firm's diversity and inclusion culture from employee reviews. It includes many key terms that we believe should be included in the final dictionary (e.g., "open door policy," "embrace diversity," "different perspectives," "family-friendly," "respectful environment," "collaborative atmosphere"). From the proposed keywords, we create a dictionary that comprises 158 words, and the Python program successfully counts the relative frequencies of these keywords in the individual employee reviews. The average frequencies per firm and year provide the firm-year diversity and inclusion score. Figure 1 shows the relative frequencies of the diversity and inclusion scores. It reveals that there is considerable variance in the measure (mean = 2.11, std. dev. = 0.68). Finally, we validate the procedure by comparing our diversity and inclusion scores with diversity and inclusion ratings that employees can provide on Glassdoor in the most recent years. Figure 2 shows a corresponding scatter plot. The correlation coefficient is 0.61. This validation check suggests that our diversity rating captures a construct related to diversity and inclusion as employees understand it but includes additional dimensions not captured by this measure.

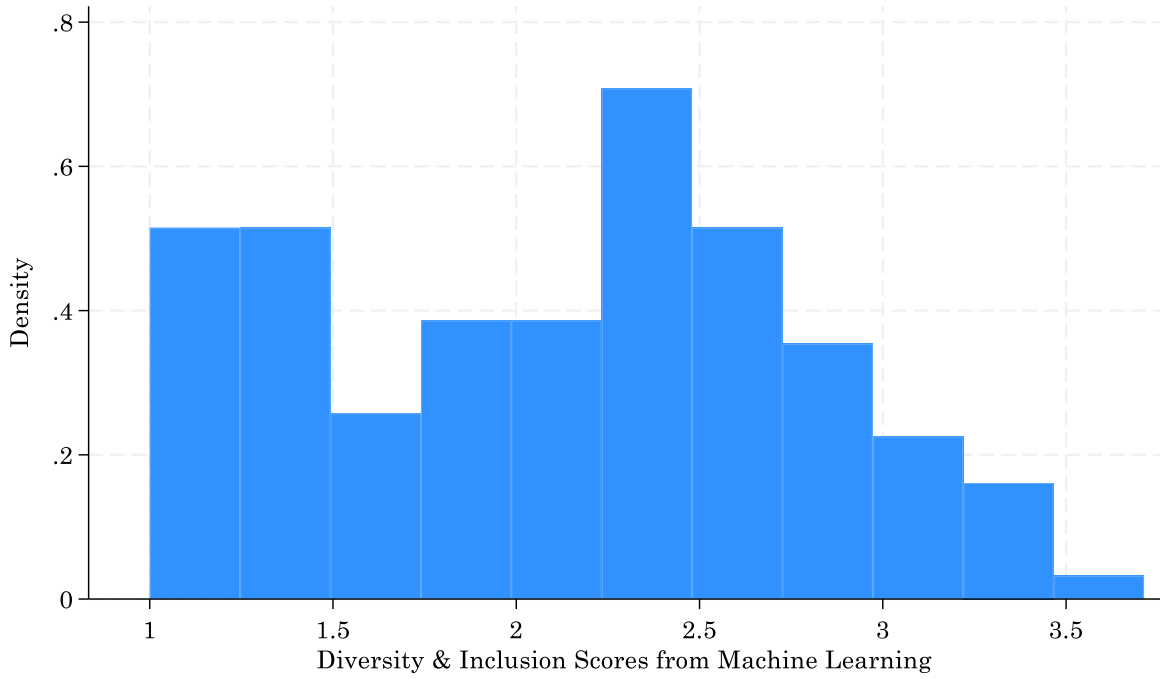

Figure 1. Histogram of the diversity and inclusion scores calculated through the machine-learning algorithm for textual analysis of employee reviews.

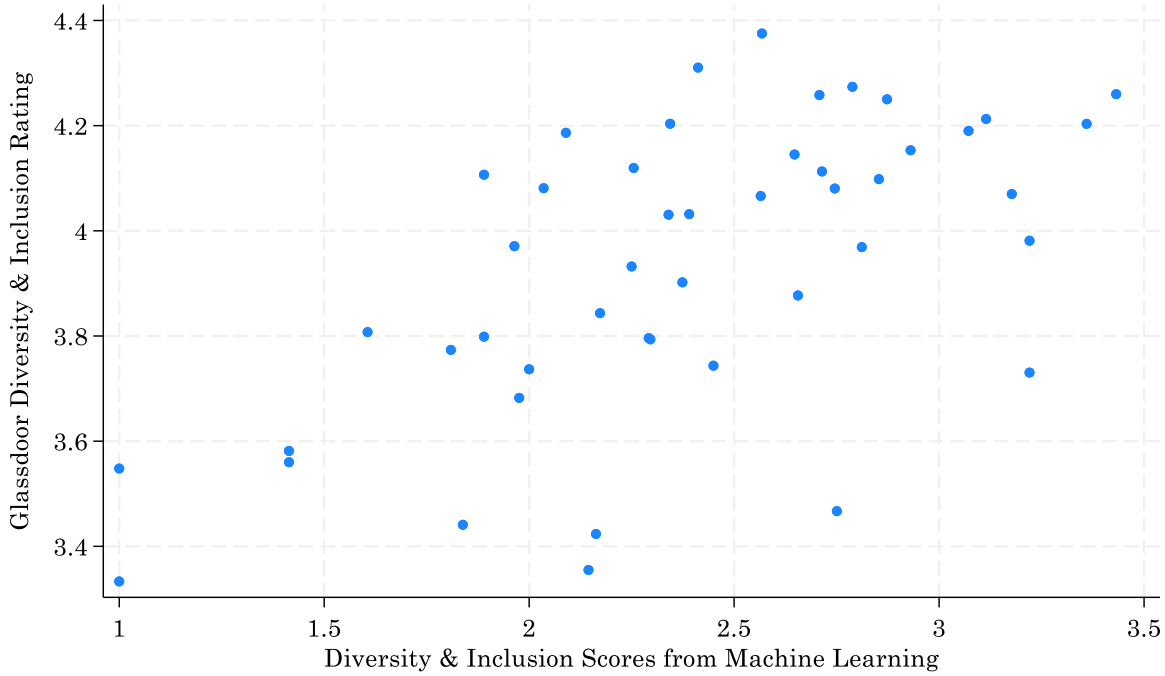

Figure 2. Scatter plot of Glassdoor diversity and inclusion ratings and the scores calculated through the machine-learning algorithm for textual analysis of employee reviews.

References

Li, K., Mai, F., Shen, R., & Yan, X. (2021). Measuring corporate culture using machine learning. *The Review of Financial Studies*, *34*(7), 3265-3315.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems*, *26*.

Appendix

Appendix 1 – Overview of firms for the pilot study

Firm	Link to Glassdoor reviews
Apple	https://www.glassdoor.com/Reviews/Apple-Reviews-E1138.htm
Microsoft	https://www.glassdoor.com/Reviews/Microsoft-Reviews-E1651.htm
General Motors	https://www.glassdoor.com/Reviews/General-Motors-GM-Reviews-E279.htm
Volkswagen	https://www.glassdoor.com/Reviews/Volkswagen-Group-Reviews-E3515.htm
SAP	https://www.glassdoor.com/Reviews/SAP-Reviews-E10471.htm
Shell	https://www.glassdoor.com/Reviews/Shell-Reviews-E5833.htm
Roche Holding AG	https://www.glassdoor.com/Reviews/Roche-Reviews-E3480.htm
L'Oreal	https://www.glassdoor.com/Reviews/L-Or%C3%A9al-Reviews-E3470.htm
Samsung Electronics	https://www.glassdoor.com/Reviews/Samsung-Electronics-Reviews-E3363.htm
Volvo Car	https://www.glassdoor.com/Reviews/Volvo-Cars-Reviews-E37990.htm