Computerized Text Analysis of Student Reflections in the First Year Engineering Curriculum

Elizabeth Strehl, Robin Fowler

Department of Technical Communication, University of Michigan

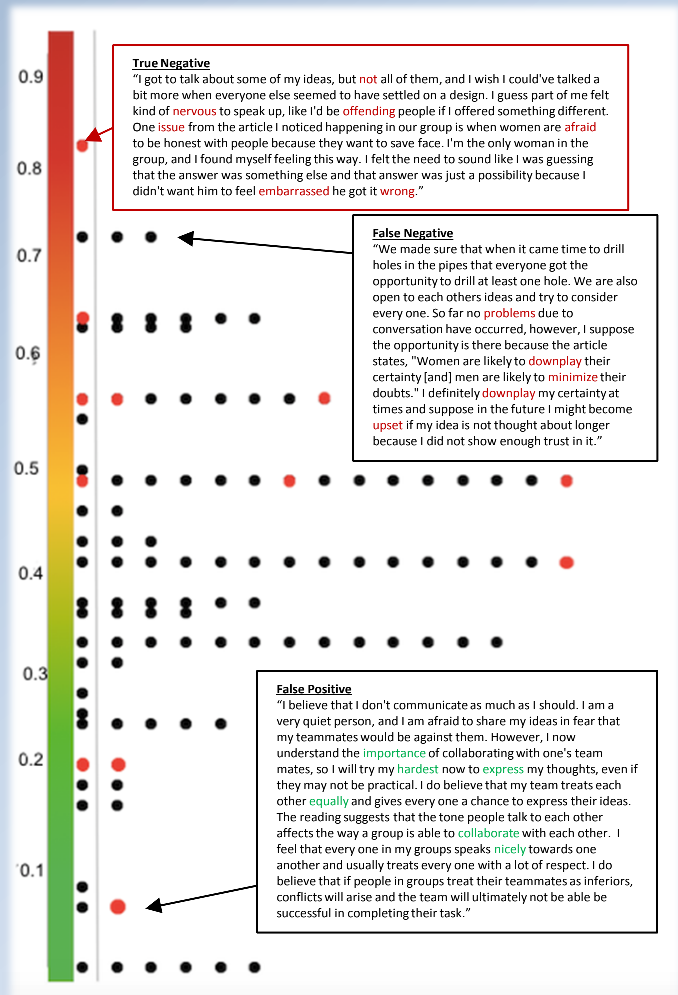
Abstract/Introduction

- Sentiment analysis is the computational process of identifying and categorizing opinions expressed in text in order to determine the writer's attitude towards a certain topic
- It could prove to be a useful tool in an academic setting, especially for collegiate instructors of large classes looking to gain better insight on students' progress individually and within teams
- The purpose of this research is **to investigate whether or not sentiment analysis programs can be used to flag particular student responses to help faculty identify struggling individuals and student groups sooner than without the program**
- Example student responses from an Engineering 100 course were analyzed by both a faculty member and a sentiment analysis program

Materials/Methods

- Data was collected from first-year engineering students enrolled in the Engineering 100: Underwater Vehicle Design course (N= 114). Students answered questions based on a reading or personal experience on teams.
- Each response was run through the **Python NLTK Module** and the neutrality-polarity and positive-negative ratios were recorded. The percent polarized multiplied by the percent negative gave a portion negative value.
- A faculty member analyzed each of the student responses and categorized them as neutral/positive or "concerning," indicating teams/situations she should watch more closely.
- True positives and true negatives (hits) and false positives and false negatives (misses) were analyzed further to determine what language characteristics caused difficulty for the program.

Results



Conclusion

The faculty member flagged 11% of student responses as "concerning". Of these, 75% were correctly ranked as negative by the Python NLTK Sentiment Module. However, of the remaining ("non-concerning") student responses, an additional 31% were also ranked negative by the software. Additionally, the software missed (ranked as positive) 27% of the responses the faculty considered concerning.

The software may have falsely ranked a negative comment as positive or neutral for the following reasons:

- The use of positive words at the end of the response to signify optimism for the future (Ex: "I hope that from now on we will learn how to communicate better")
- Neutral or positive language throughout the response in regards to the assigned pre-laboratory reading (Ex: "The article encouraged diversity in teams to improve problem-solving skills within the group.")
- The use of a typically positive adjective to describe a negative occurrence (Ex: "It was a great disappointment")

On the other hand, the program may have falsely ranked a positive or neutral statement as negative for the following reasons:

- The use of typically negative words to describe a neutral or positive statement (Ex: "I hate that we couldn't meet for longer during class")
- Negative language throughout the response in regards to the assigned pre-laboratory reading (Ex: "The article discussed the disadvantages of a lack of diversity on teams and how it affects problem-solving")

While the Python NLTK Sentiment Analysis Module did not accurately identify struggling student groups in this course, there are many possibilities to refine the program specifications or student writing assignment for future research. Further program development to better distinguish between truly negative and truly positive statements would help eliminate the false software rulings. By limiting the student response to only the area of concern (i.e. diversity/communication on your team) instead of requiring discussion about a pre-laboratory article, the Python software would be able to better analyze the relevant content of response.