# Reading Assistant: Project Progress Report

Christopher Rock (cmrock2) (Captain)
Zichun Xu (zichunx2)
Kevin Ros (kjros2)

Due 29 November 2020

## 1 Progress

We began by planning a road-map for our project, which consisted of a set of goals and group meeting checkpoints to evaluate progress. Additionally, we created a GitHub repository which holds our project code.

Following this, we implemented the basic structure for our reading assistant. On start-up, text documents (directory path provided by user) are loaded by the assistant. During loading, the documents are processed and added to an inverted index. These documents are considered to be the previously-read documents by the user. Once the loading is complete, the assistant waits for a path to a text file (unseen document). Given this path, the assistant ranks the previously-read documents using the unseen document and returns the most similar read document names to the user. We also provided methods for a user to add and remove previously-read documents.

Currently, the assistant calculates similarity using the Okapi BM25 ranking function along with various optimization techniques, including an inverted index. To heuristically gauge the effectiveness of this method, each team member collected approximately 8-10 documents. These documents were loaded as the previously-read documents, and additional documents were provided as the unseen documents. From the preliminary examination, the results seem promising.

The code is written in a modular fashion, so that we can easily extend the assistant to use different similarity/difference measures and methods.

In addition to document-level BM-25, we have implemented paragraph-level BM25 which allows more detailed evaluation of unseen documents compared to seen documents. We have also used the external library **gensim** to include Latent Semantic Indexing at a document level, with document-level similarity. This is currently a separate script and will be integrated for the final project. Our planned extensions are discussed in the following section.

## 2 Remaining Tasks

Our first remaining task is to add more fine-grained similarity and difference measurement techniques. Regarding the ranking function itself, we are considering adding pre-trained word embeddings and cosine similarity to effectively assess similarity and differences on a word and sentence level. We will combine this with our Okapi BM25 calculations to compare seen and unseen documents on a paragraph granularity.

Our second remaining task is to create a user-friendly command line interface. This will allow the user to easily add and remove documents, and view the similarities and differences between

the seen and unseen documents. Ideally, we plan to output a detailed summary that describes the relationship between the documents.

## 3 Challenges and Issues

One particular challenge that we've encountered is the evaluation of our reading assistant's effectiveness. Because we haven't encountered a data set that exactly fits our needs, we plan to address this issue by incorporating a feedback mechanism in the terminal. This way, users can provide real-time feedback that we can dynamically incorporate into the reading assistant.