Simi Bot (Similarity Analyzer)

Applied Project Final Report

By

Wukun Chen

Spring, 2021

A paper submitted in partial fulfillment of the requirements for the degree of
Master of Science in Management and Systems
at the
Division of Programs in Business
School of Professional Studies
New York University

Table of Contents

Declaration

I, Wukun Chen, declare that this project report submitted by me to the School of Professional Studies, New York University in partial fulfillment of the requirement for the award of the degree of Master of Science in Management and Systems is a record of project work carried out by me under the guidance of Dr. Andres Fortino, NYU Clinical Assistant Professor of Management and Systems. I grant powers of discretion to the Division of Programs in Business, School of Professional Studies, and New York University to allow this report to be copied in part or in full without further reference to me. The permission covers only copies made for study purposes or inclusion in the Division of Programs in Business, School of Professional Studies, and New York University research publications, subject to normal conditions of acknowledgment. I further declare that the work reported in this project has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Acknowledgments

I sincerely thank Dr. Andres Fortino for his contribution as the sponsor of this project and as my mentor during this project. I also want to thank all the instructors in the Management and Systems program who I have taken courses with and learned a great deal.

Abstract

The purpose of this project is to develop an application to perform TF-IDF text similarity scoring analysis for NYU School of Professional Studies and the Management and Systems program (MASY). This application is programmed in the R programming language and hosted on the shinyapp.io server. This text data mining application is featuring topic clustering, keyword extraction, machine learning, cloud computing, and Shiny-based user experience. This project allows users to customize unsupervised machine learning hyperparameters and upload files locally. After uploading a .txt file (comparison source) and a .csv file (comparison target), users need to choose the number of clusters for the text cluster analysis (from 2 to 20), the number of most frequent words to display for each text cluster (from 2 to 20), and the level of word combinations (from 1 to 3). When a user clicks the "Analyze Data" button after all the hyperparameters are set, this application will generate two data tables to indicate similarity scores, cluster group, size of each cluster group, and the keyword in each cluster group.

The underlying algorithms of this program are as following: cleanse the text source and target, drop all non-alphabetical characters, eliminate multi-space, and lemmatize all the words; apply TF-IDF transformation, compute similarity-score against the source to each target; cluster with hierarchical method; calculate the mean similarity scoring by the group to determine the cluster of the max mean; output data table with cluster group, size of each cluster group, and the keyword in each cluster group.

With this new tool, students studying data analysis and machine learning would have an easy-to-use R tool to perform TF-IDF text similarity scoring analysis, which works for both Windows-based PCs and Apple Macs. Scenarios that we can put into use include matching resumes to occupations, matching a syllabus to occupations, matching resumes to program syllabi to discover gaps, and recommend courses. Templates, samples, and comprehensive tutorials are provided in the application.

URL of Simi Bot: https://simibot.shinyapps.io/SimiBot/
GitHub Repository: https://github.com/ericchen1785/SimiBot/

Abbreviations and Definitions

R: A programming language and free software environment for statistical computing and graphics

MASY: Master of Science in Management and Systems is a master's program offered through the Management & Technology department within the Division of Programs in Business at the School of Professional Studies at New York University.

NYU: New York University, a private research university based in New York City

SPS: School of Professional Studies at New York University

Text Mining: The process of deriving high-quality information from text

Introduction

Background information

NYU School of Professional Studies and the Management and Systems program (MASY) appeals to develop a tool to perform TF-IDF text similarity scoring analysis in the R program. I will build on earlier work at NYU done in Python with an easy-to-use user interface. I will program the tool in R language and deliver the text-similarity scoring function. With this new tool, students studying data analysis and machine learning would have an easy-to-use R tool to perform TF-IDF text similarity scoring analysis, which works for both Windows-based PCs and Apple Macs. Users can upload a text file (comparison source) and a .csv file (comparison target). After execution, the tool will generate a new .csv file with a new column indicating the similarity scores. The similarity scores aid user in decision-making.

Company Name

New York University (NYU) is a private research university based in New York City. The university locates at 7 East 12th Street, New York, NY 10003. The MASY program is based on a unique curriculum that provides students with experiential learning opportunities to develop strong management and leadership skills and gain a comprehensive knowledge of current information technologies.

Sponsor Information

Dr. Andres Fortino is the Clinical Associate Professor and MASY ACP Leader at NYU. Over the past 40 years, he gained a great deal of experience in higher education as a faculty member, as an administrator, as well as an executive in many colleges and universities. He is currently a Clinical Associate Professor of Management and Systems at the NYU School of

Professional Studies and has taught in the Management and Systems (MASY) master's program since 2012. Previously, he held the Campus Provost and Dean of Academic Affairs position at DeVry College of New York. Formerly (2006-10). He was on the faculty and held an Associate Provost's position at the Polytechnic Institute of NY, where he managed two campuses and numerous graduate programs, among other duties. Before that (2004-06), he was Dean of the Marist College School of Management, where he successfully initiated many new graduate programs. And he spent six years before that (1998-06) on the faculty and as Associate Dean at George Mason University School of Management, where he managed their MBA and other master's programs.

Problem Description/Opportunity

NYU MASY wishes to develop a tool to perform TF-IDF text similarity scoring analysis in the R program. The consultant will build on earlier work at NYU done in Python with an easy-to-use user interface. Students studying data analysis and machine learning currently do not have an easy-to-use R tool to perform TF-IDF text similarity scoring analysis. The current tool is built on Python code with a Heroku web interface. We wish to develop a more transparent and easier to build on tool in R. NYU MASY now wishes to complete the development of text analytic tools and make sure the tool works for both Windows-based PCs and Apple Macs.

Deliverables of this project include:

1. Become familiar with TF-IDF text analytics in R as well as how to develop shiny apps for R.
2. Become familiar with the previous Python/Heroku tool for text analytics.
3. Build an R-based tool to emulate the functionality of the earlier Python tool. Exemplar and target text file configuration and ingestion. For an input text file, a resume, for example, and target text file, texts of many job descriptions, the algorithm should return a ranked file of the jobs based on similarity scores. The output file should be viewable and downloadable.
4. Build a shiny app easy-to-use interface to the tool.
5. Test the validity of the algorithm using client-defined data sets.
6. Deliver a ready-to-install add-on tool to R, with clear user instructions for installation and use.
7. The final deliverable must be in working code with a shiny app interface with installation and operating instructions.

8. Final project files and supporting documentation to be delivered via an accessible GitHub repository

The opportunity of this program is offering students at NYU or ever the public a free and powerful similarity scoring application. Besides, this project adds value to document clustering, which makes it a more useful tool for text data mining.

Importance of the project

The earlier work at NYU is built on Python code with a Heroku web interface. That program is sensitive to data type and has a high requirement for data format. Also, the program running time is too long according to the user experience. The importance of this project is to offer students studying data analysis and machine learning an easy-to-use R tool to perform TF-IDF text similarity scoring analysis.

Further than that, this project greatly reduces the repetitive work of text comparing and similarity scoring. Also, it does not require users to learn about any coding language. Users only need to know how to copy and paste their text contents into the .txt and .csv templates. Finally, this project is purely non-profit and education-oriented, so the interface is ad-free, which provides a better user experience than other web applications.

Alternate Solutions Evaluated

Here are several alternative solutions for the NYU MASY program to develop a similarity scoring application.

1. Improve the previous code based on Python and Heroku. The problems of former work include discontinuous front-end user experience, the inefficiency of the code, and tedious running time.

   Pros:

   First, the continued development of the Python program is free. It would be free for students and users. Second, Python is a very popular programming language. Abundant free resources and packages are available for use. Third, this application would be easy to use and revise. Since we develop and admin the application, we have full copyright and access to make any change in the future.

   Cons:

   First, there could be some unexpected difficulty to integrate the Python code with the Heroku front-end. Second, the cloud computing resource on the Heroku server is limited, which results in long running time. This problem can not be migrated on the software development side.

2. Outsource the program to other companies. The problem with this alternate solution is, we have a limited budget for this program. Also, when this program is created, it would be open-source and free to the public as a non-profit program.

   Pros:

   First, professional software development companies which featuring software as a service could deliver very mature and sophisticate product. The final product will be robust, responsive, user-friendly, and heavy-duty. Such a program could be published on a public server for commercial use.

Cons:

Outsourcing a program is expensive while we have a limited budget. Also, when the contract ends and the outsourcing company delivers the project, we are responsible to maintain the software and debug it if any deficiency occurs. Since we are not engaged in the software development phase, it would be a headache to take over the project and understand how the application works. It would be extremely time-consuming and may end up solving the problem by outsourcing the task to another company.

3. Build the program based on R and host on GitHub. This alternative solution is using R directly to analyze the data and conduct similarity scoring. It could be the simplest way to develop a program based on an R script. But the problem is, this solution requires a deep understanding of R and the ability to customize the code, which is not user-friendly. Plus, the source code could be vulnerable and may most perform if there any error in the code.

   Pros:

   First, developing an R program is free. This product would be free for students and users. Second, R is a very popular programming language. Abundant free resources and packages are available for use. There should be no problem to deliver a program to conduce similarity scoring. Third, since we develop and admin the program, we have full copyright and access to make any change in the future.

   Cons:

   First, users are required to install the R program on their computer and know how to customize the R script. Second, users are required to know how to read and write the file with the path on their computer. Third, if any error occurs, there would be no instruction or error handling for help. Users would see the original error message in the R console and need to troubleshoot by themselves.

Solution Evaluation Criteria

Here are several solution evaluation criteria for the NYU MASY program to evaluate solutions and select the final one.

1. Whether the solution meets NYU MASY's objectives of creating a similarity scoring application.
2. Whether the solution meets NYU MASY's exception on time, good quality, and within budget.
3. Whether the client has access to admin the program, and how hard to maintain the program after the project closing.
4. Whether the solution has extra maintenance cost after the project closing.

Selection Rationale

Based on the four solution evaluation criteria listed above, we could compare three alternative solutions with our plan.

First, from the perspective of building an "application", writing an R script is not qualified. What we need is a mature tool that will benefit people who have no technology background. So, a user-friendly interface is a must, either the backend is based on R or Python.

Second, from the overall feasibility perspective, either of the alternative solutions might be finished on time with good quality. However, the outsourcing solutions are not free, thus they will not be delivered within budget.

Third, the project's continuous development after project closing is critical. If we program the application base on R or Python, then we have no problem doing any continuous development. While if we outsource the project, it probably would not be easy to take over and maintain the program.

Forth, if we program the application base on R or Python, we anticipate the program staying free to maintain and for use. If our application is further developed and attracts more traffic, we might need to upgrade to premium service on the server and expand the cloud computing availability. The related estimated cost is $9 - $39 per month. While if we outsource the project, the estimated cost is much farther away more than the former.

Approach and Methodology

Work Breakdown Structure, Industry Analysis, Stakeholder Analysis, Porter's Five-Force Analysis, Project Plan, Gantt Chart, and Resource Loading are used in the project.

Work Breakdown Structure

1.1 Initiation

1.1.1 Evaluation & Recommendations

1.1.2 Develop Project Proposal

1.1.3 Develop Project Charter

1.1.4 Deliverable: Submit Project Charter

1.1.5 Project Sponsor Reviews Project Charter

1.1.6 Project Charter Signed/Approved

1.2 Planning

1.2.1 Create Preliminary Scope Statement

1.2.2 Determine Project Team

1.2.3 Project Team Kickoff Meeting

1.2.4 Develop Project Plan

1.2.5 Submit Project Plan

1.2.6 Milestone: Project Plan Approval

1.3 Execution

1.3.1 Verify & Validate User Requirements

1.3.2 Design System

1.3.3 Install Development System

1.3.4 Testing Phase

1.3.5 Install Live System

1.3.6 User Training

1.3.7 Go Live

1.4 Control

1.4.1 Project Management

1.4.2 Project Status Meetings

1.4.3 Risk Management

1.4.4 Update Project Management Plan

1.5 Closeout

1.5.1 Audit Procurement

1.5.2 Document Lessons Learned

1.5.3 Update Files/Records

1.5.4 Gain Formal Acceptance

1.5.5 Archive Files/Documents

Industry Analysis

According to IBIS World, the software publishing industry in the US could be defined as "Software publishers disseminate licenses to customers for the right to execute software on their computers. Operators in this industry market and distribute software products and may also design the software, produce support materials and provide support services"(Cook, 2020). Software as a service (SaaS) is "a model of software deployment in which a provider licenses an application to customers for use as a service on demand"(Cook, 2020).

This industry reported 293.1-billion-dollar revenue in 2020 and is expected to have 2.4% annual revenue growth in 2020-2025. Profit was reported at $83.2 billion with 5.2% average growth in the past 5 years. The average profit margin in the last 5 years was 28.4% with an outlook of 1.77pp downside(Cook, 2020).

In general, this industry is still in the growth section of the industry life cycle. Capital intensity and regulation are light, which gives us chance to break in as a newbie. However, industry assistance is low and would be steady. The technology change, industrial globalization, and competition are high and would be increasing.

The strength of this industry would be the growth life cycle stage, low imports, low capital requirements. The weaknesses are the low and steady level of assistance, high competition, high customer class concentration, high product and service concentration, and low revenue per employee. Potential opportunities are high revenue growth and investor uncertainty. At last, the threats are low outlier growth, low-performance drivers, and private investment in computers and software.

Current competitors in the text-similarity analysis software industry include twinword inc. (Amazon AWS), Grammarly, Turnitin, Copyleaks, Prepostseo, Duplichecker. These web service suppliers already offer mature text similarity analysis products. Potential competitors include Microsoft Corporation, International Business Machines Corporation, Apple Inc., Oracle Corporation, etc. These technology giants are extremely capable and wealthy. They are likely to create disruptive software with sufficient funds.

Stakeholder Analysis

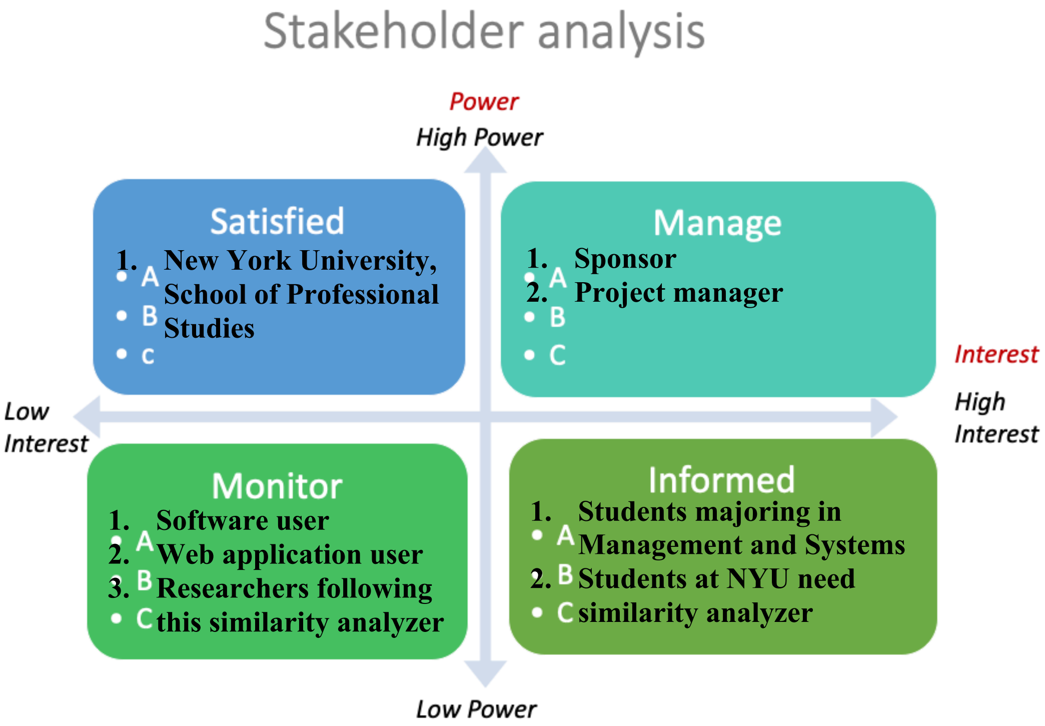
List of all stakeholders :

1. Sponsor, Dr. Andres Fortino
2. Project manager, Wukun Chen
3. New York University, School of Professional Studies
4. Teammates
5. Colleagues
6. Students majoring in Management and Systems
7. Students at NYU need a similarity analyzer
8. Software user
9. Web application user
10. Researchers following this similarity analyzer

For internal stakeholders (1-3), their opinion will directly influence the business need and scope of this project. For external stakeholders (4-10), we will regard their opinion as feedback and keep improving this product.

Figure 1

Stakeholder Analysis



Note. Stakeholder Analysis of Simi Bot

Porter's Five-Force's Analysis

The industry of my project belongs to Software computer, packaged, publishers (NACIS code 511210).

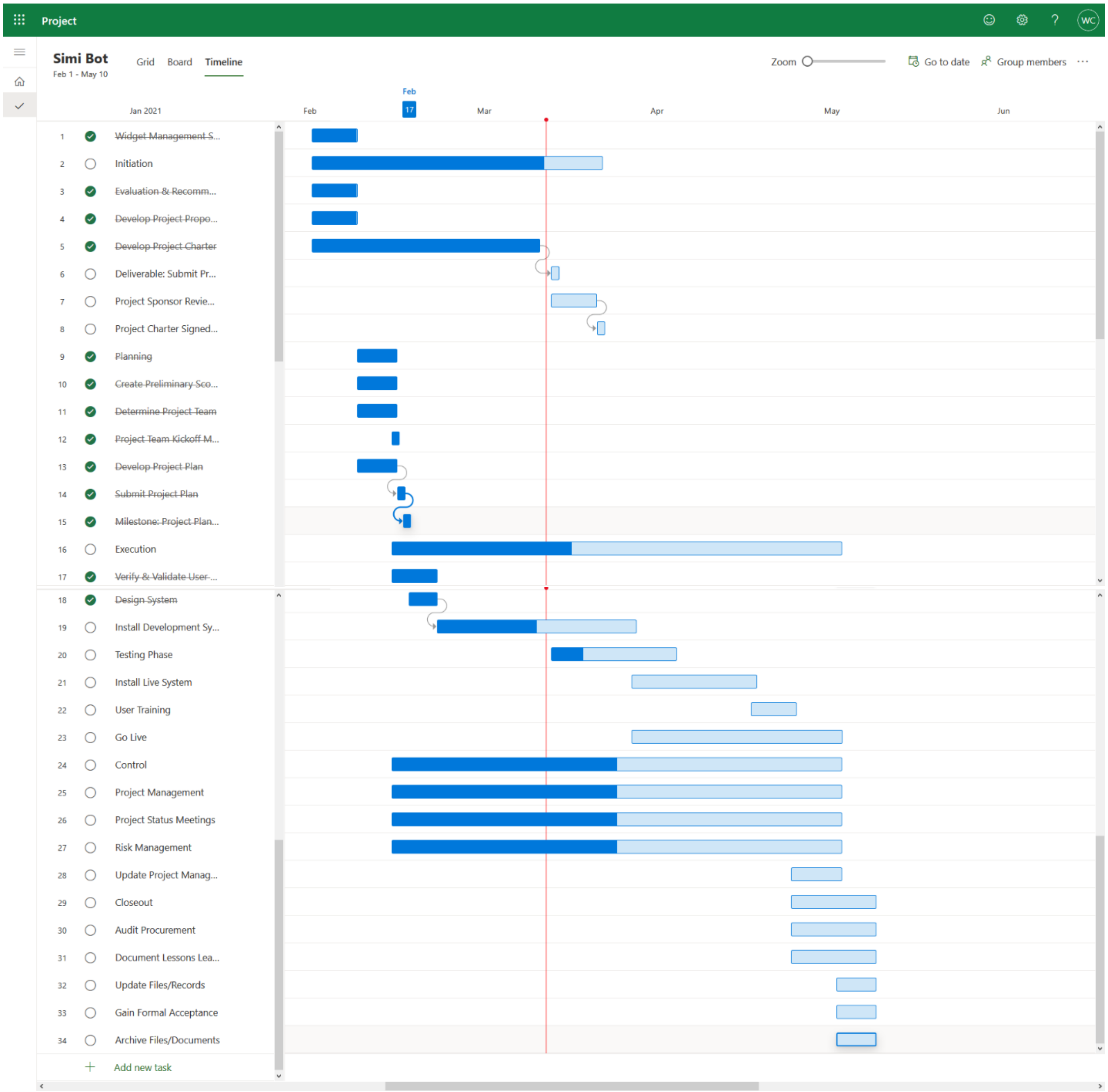
Competitors against whom the business competes include twinword inc., Grammarly, Turnitin, Copyleaks, Prepostseo, Duplichecker.

Our business cultures include user-centered, benefited from open-source, and give back to open-source, relentless optimization.

Gantt Chart (Microsoft Project)

Figure 2

Gantt Chart



Simi Bot
Feb 1 - May 10

1	✓	~~Widget Management S...~~
2	○	Initiation
3	✓	~~Evaluation & Recomm...~~
4	✓	~~Develop Project Propo...~~
5	✓	~~Develop Project Charter~~
6	○	Deliverable: Submit Pr...
7	○	Project Sponsor Revie...
8	○	Project Charter Signed...
9	✓	~~Planning~~
10	✓	~~Create Preliminary Sco...~~
11	✓	~~Determine Project Team~~
12	✓	~~Project Team Kickoff M...~~
13	✓	~~Develop Project Plan~~
14	✓	~~Submit Project Plan~~
15	✓	~~Milestone: Project Plan...~~
16	○	Execution
17	✓	~~Verify & Validate User...~~
18	✓	~~Design System~~
19	○	Install Development Sy...
20	○	Testing Phase
21	○	Install Live System
22	○	User Training
23	○	Go Live
24	○	Control
25	○	Project Management
26	○	Project Status Meetings
27	○	Risk Management
28	○	Update Project Manag...
29	○	Closeout
30	○	Audit Procurement
31	○	Document Lessons Lea...
32	○	Update Files/Records
33	○	Gain Formal Acceptance
34	○	Archive Files/Documents

Note. Gantt Chart of Simi Bot

Project Objectives and Metrics

The goal of the project

The goal of the project is to create a computer-based tool to perform TF-IDF text similarity scoring in R with an easy-to-use user interface. We came across lots of information arrangements based on text similarity, such as matching one resume to multiple job descriptions or matching one faculty profile to multiple course descriptions. NYU MASY wishes to develop a tool to perform TF-IDF text similarity scoring analysis in the R program. They have earlier work done in Python with an easy-to-use user interface with Heroku. The problems of the previous work include long webpage loading time, incompatible file type, relying on web connection, etc. We now wish to complete the development of text analytic tools and make sure the tool works for both Windows-based PCs and Apple Macs.

After this project, my client will have a more transparent and easier build on text analytic tools in R based on Python code with a Heroku web interface. It will be a powerful tool in a general text comparison. Scenarios that we can put into use include matching resumes to occupations, matching a syllabus to occupations, matching resumes to program syllabi to discover gaps and recommend courses, etc. We might have some add-ons to extend the function. For instance, we can add a text box to receive the source keyword so people can copy and paste instead of creating a new text file. Or we allow users to input masked words, so we will ignore that wording in text-similarity scoring.

As the person conducting the project, I will convert an earlier Python-based web application to a new R-based local application. I am accounted for the project throughout its lifecycle. I will engineer and deliver the application on my own.

As the project manager of my project, I will coordinate and communicate with the sponsor of the project. I am accounted for initiating, planning, executing, monitoring, controlling, and closing the project. I will determine the business needs, make sure every objective is completed within the time frame, and keep the guideline align with the sponsor's requirement.

The project was started on January 25th, 2021, and the final version will be delivered on May 17th,2021. My proposed duration is 280 hours in the time frame of 14 weeks (February 8th to May 14th). The weekly expected work hour is 20 (Monday to Friday, 8 PM to 12 AM).

Project Deliverables and Metrics

Project Objective 1 – Build a functional requirement definition document.

Metric: Have a document with functional specification

Project Objective 2 – Build an R-based tool to emulate the functionality of the earlier Python tool.

Metric: Deliver the R script by end of the semester

Project Objective 3 – Build a shiny app with an easy-to-use interface.

Metric: Deliver a working shiny interface delivered by end of the semester

Project Objective 4 – Develop a tutorial for installation and use and lab manual. Develop and populate a GitHub repository of project deliverables by end of the semester

Metric: Deliver by May 5th, 2021 to pass a customer test based on customer's data

Risk Analysis

I identified five major potential risks, while none of them happened:

1. Cannot deliver on time

   I fast-tracked the project and left enough cushion for debugging and troubleshooting. In the end, I deliver the problem one week ahead of the schedule.

2. Bugs and Error in the Code

   My original contingency plan is to reach out for help from coworkers or other people who are proficient in R. However, I solved most of the problems by myself through online searching.

3. Computer Deficiency

   My original contingency plan is to maintain a backup code on the cloud or purchase a new computer to continue the program. In practice, I backed up my code on Google drive weekly to prevent computer deficiency.

4. Client requests to change the project

   If my client request to change the scope, I would negotiate about new business needs and opportunities to continue the project.

5. Health issue

   Normal health issues would not influence the progress of this project. During this project, I took care of my wellness, got vaccinated for Covid-19, and had a balanced diet to ensure my health conditions.

Issues Encountered

While working on the project, I encountered some issues. All of the issues I faced are minor issues that do not have a major impact on the project. All issues were solved immediately once indicated so that the project was able to finish on time with high quality. Here is all type of issues project I faced in the duration of the project.

The first issue I faced was adapting R code to shiny. Shiny is a very powerful tool to realize the functionality of R code with a user-friendly front-end. However, the running sequence and logic are different within the Shiny framework. For instance, code in Shiny would be rendered only when users open that page/tab. I educated myself with LinkedIn Learning and Stack Overflow to optimize most of the problems. By the end of the semester, I delivered a web application with a seamless user experience with responsive front-end design.

The second issue I faced was the running time of the program. I was very concerned about it when I ran the code on my local personal computer, which took 40 seconds to render 2 output data tables with the sample files. I tried to eliminate the redundant code and streamlined logic within my code to reduce the running time. After my improvement, the running time on my local personal computer reduced 35% (average 26 seconds). After the application was published on the shinyapp.io server, the running time was not a problem anymore. Thanks to the powerful computing resource of the cloud, the average running time is less than 5 seconds, which is very satisfying for most of the users.

The third issue I faced was the change of objectives. I need to work closely with my client and follow up with the latest business need. My client roused lots of creative ideas and requests, I need to evaluate the feasibility and give timely feedback. Some of the requests are hard to fulfill due to the scope, time, or budget, others are reasonable and meaningful. For the latter ones, I accepted the challenge and made the shift to the project.

Project Chronology and Critique

The room for improvement in this project is the responsiveness of the front-end, the running time of the program, and the algorithm of the clustering. I could work on these areas to improve the quality of my project.

First, for the responsiveness of the front-end, the current experience could be more consistent. For instance, when the users click “Analyze Data”, the program should automatically redirect the frond-end page to the result tabs. Another example is the Shiny App only renders the code related to the current tab. So, when users click and open the result tabs, they still need to wait for the code to run and render the output. These actions should be activated when users click “Analyze Data” too. In the end, there are lots of responsive fold, hover, highlight actions that could be added to this application using JavaScript.

Second, the running time of the program is a subtle area that we could improve. There are some chunks of code that we can integrate and streamline to make the code more concise. It would not be a significant improvement on the server (a few seconds), but it would be a distinct difference on the localhost. I used my computer (8 core 16 threads 2.90 GHz CPU, 16 GB RAM) to run the sample files (666 words in the sample .txt file, 38416 words within 1100 samples in the sample .csv file) as a baseline test. When I tested the application with my localhost and sample files, the running time is around 26 seconds, comparing with 40 seconds before. There are some threads of code that have not been integrated. We could assume that the running time could be shorter after we solve that problem.

Third, the algorithm of the clustering may not be the most efficient one. Simi Bot is based on hierarchy clustering, which is a basic method using TF-IDF and cosine distance. There are other ways to cluster documents, such as K-means clustering. Given the time limit of this project, I did not compare different clustering methods and selected the most efficient one. There might be some room for improvement in the overall clustering efficiency and so as the running time.

In short, this project meets the requirements of the client and is delivered within the time limit, but there is room for continuous research and improvement. For the details of the limitations and deficiencies, please refer to the *Limitations, Recommendations and Scope for Future Work* section.

Lessons Learned

The whole project was able to deliver as planned with expected quality and in time and this could not have been done without the contribution and help from my sponsor. During the whole project implementation, I have learned how to manage a project and develop software. Also, I became familiar with software development life cycle, project management, and risk management. After this project, I have more confidence to take ownership of sophisticated software development tasks.

First, my programming skills and text mining skills are improved and honed during the project. I learned how to cleanse the text source and target, drop all non-alphabetical characters, eliminate multi-space, and lemmatize all the words; apply TF-IDF transformation, compute similarity-score against the source to each target; cluster with hierarchical method; calculate the mean similarity scoring by the group to determine the cluster of the max mean; output data table with cluster group, size of each cluster group, and the keyword in each cluster group. I built a comprehensive tool to help students studying data analysis and machine learning. Also, this tool is compatible with generic text comparing scenarios including matching resume to occupations, matching a syllabus to occupations, matching resume to program syllabi to discover gaps, and recommend courses.

Second, as a project manager for this project, I have learned the skills to schedule, design, and implement a project. I always fast-tracked my progress and left enough cushion for any changes. I worked closely with my client and kept up with the latest business need. Also, I adapted the agile methodology to my project for time-sensitive feedback and update. For instance, I published Simi Bot to shinyapp.io server once I finished the scratch code. In this way, I kept my client posted with a unified link. During our weekly touch base meeting, we can quickly jump to the next phase. Last, I gained experience in developing documentations. The process of rationale exhaustively extends my vision of project management. When I was developing the contingency plan, I challenged my imagination and critical thinking to include all risks that might happen. I appreciate this opportunity to lead this project and complete it with my wonderful client.

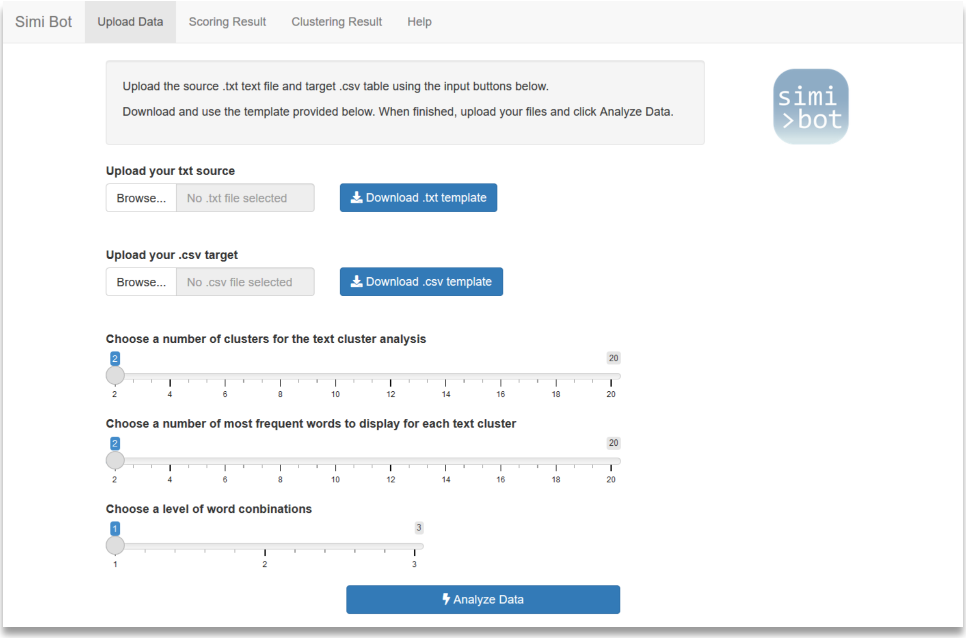
Third, this project improved my communication and collaboration skills. I kept in touch with my client through weekly meetings; I worked closely with other colleagues who were also developing text mining programs; I learned how to follow up and follow through a project. I believe that after this project, I would be a comfortable team player to work with.

Conclusion and Summary

This project is delivered on time with great quality and satisfaction from the client. It allows users to customize unsupervised machine learning hyperparameters and upload files locally. After uploading a .txt file (comparison source) and a .csv file (comparison target), users need to choose the number of clusters for the text cluster analysis (from 2 to 20), the number of most frequent words to display for each text cluster (from 2 to 20), and the level of word combinations (from 1 to 3). When a user clicks the "Analyze Data" button after all the hyperparameters are set, this application will generate two data tables to indicate similarity scores, cluster group, size of each cluster group, and the keyword in each cluster group.

Figure 3

Upload Data Page



Upload the source .txt text file and target .csv table using the input buttons below.

Download and use the template provided below. When finished, upload your files and click Analyze Data.

Upload your txt source

No .txt file selected

Upload your .csv target

No .csv file selected

Choose a number of clusters for the text cluster analysis: 2

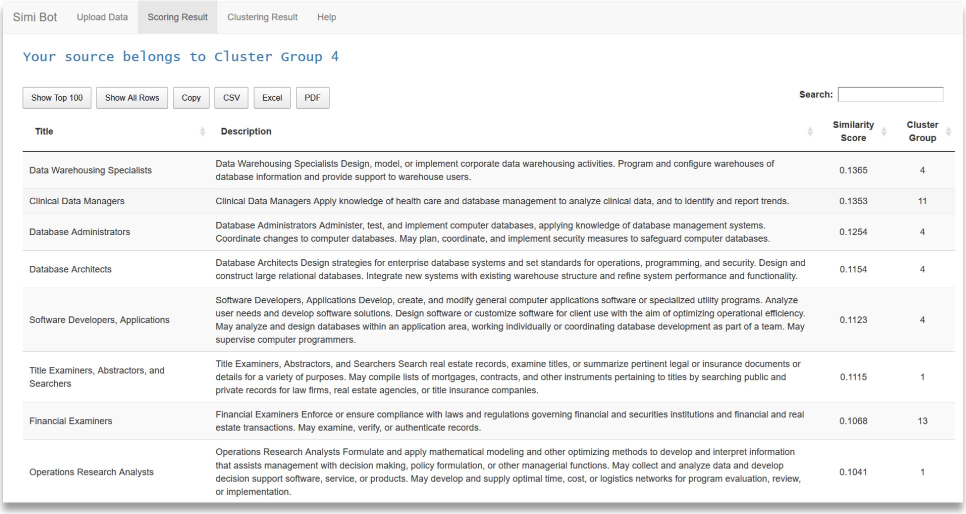
Choose a number of most frequent words to display for each text cluster: 2

Choose a level of word conbinations: 1

Note. *Simi Bot Home Page, screenshot from https://simibot.shinyapps.io/SimiBot/*

Figure 4

Scoring Result Page



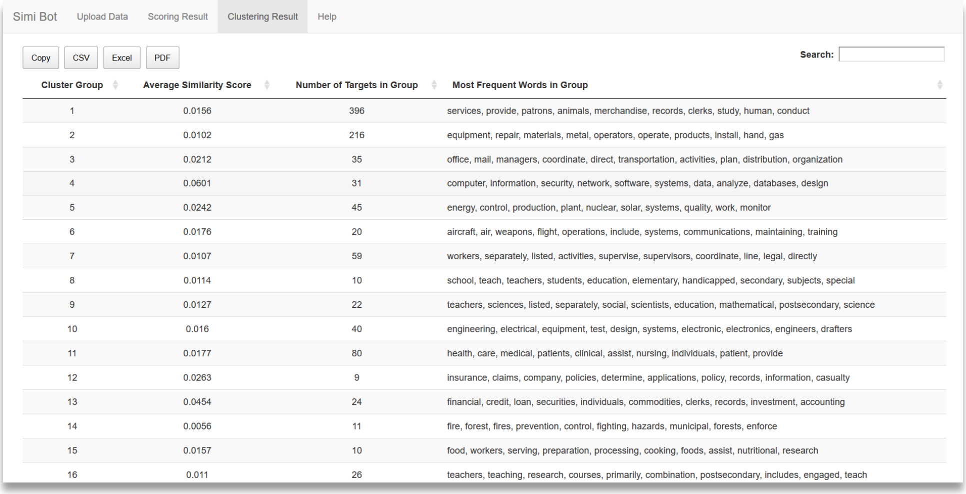
Your source belongs to Cluster Group 4

Title	Description	Similarity Score	Cluster Group
Data Warehousing Specialists	Data Warehousing Specialists Design, model, or implement corporate data warehousing activities. Program and configure warehouses of database information and provide support to warehouse users.	0.1365	4
Clinical Data Managers	Clinical Data Managers Apply knowledge of health care and database management to analyze clinical data, and to identify and report trends.	0.1353	11
Database Administrators	Database Administrators Administer, test, and implement computer databases, applying knowledge of database management systems. Coordinate changes to computer databases. May plan, coordinate, and implement security measures to safeguard computer databases.	0.1254	4
Database Architects	Database Architects Design strategies for enterprise database systems and set standards for operations, programming, and security. Design and construct large relational databases. Integrate new systems with existing warehouse structure and refine system performance and functionality.	0.1154	4
Software Developers, Applications	Software Developers, Applications Develop, create, and modify general computer applications software or specialized utility programs. Analyze user needs and develop software solutions. Design software or customize software for client use with the aim of optimizing operational efficiency. May analyze and design databases within an application area, working individually or coordinating database development as part of a team. May supervise computer programmers.	0.1123	4
Title Examiners, Abstractors, and Searchers	Title Examiners, Abstractors, and Searchers Search real estate records, examine titles, or summarize pertinent legal or insurance documents or details for a variety of purposes. May compile lists of mortgages, contracts, and other instruments pertaining to titles by searching public and private records for law firms, real estate agencies, or title insurance companies.	0.1115	1
Financial Examiners	Financial Examiners Enforce or ensure compliance with laws and regulations governing financial and securities institutions and financial and real estate transactions. May examine, verify, or authenticate records.	0.1068	13
Operations Research Analysts	Operations Research Analysts Formulate and apply mathematical modeling and other optimizing methods to develop and interpret information that assists management with decision making, policy formulation, or other managerial functions. May collect and analyze data and develop decision support software, service, or products. May develop and supply optimal time, cost, or logistics networks for program evaluation, review, or implementation.	0.1041	1

Note. *Scoring result with sample files, screenshot from https://simibot.shinyapps.io/SimiBot/*

Figure 5

Clustering Result Page



Cluster Group	Average Similarity Score	Number of Targets in Group	Most Frequent Words in Group
1	0.0156	396	services, provide, patrons, animals, merchandise, records, clerks, study, human, conduct
2	0.0102	216	equipment, repair, materials, metal, operators, operate, products, install, hand, gas
3	0.0212	35	office, mail, managers, coordinate, direct, transportation, activities, plan, distribution, organization
4	0.0601	31	computer, information, security, network, software, systems, data, analyze, databases, design
5	0.0242	45	energy, control, production, plant, nuclear, solar, systems, quality, work, monitor
6	0.0176	20	aircraft, air, weapons, flight, operations, include, systems, communications, maintaining, training
7	0.0107	59	workers, separately, listed, activities, supervise, supervisors, coordinate, line, legal, directly
8	0.0114	10	school, teach, teachers, students, education, elementary, handicapped, secondary, subjects, special
9	0.0127	22	teachers, sciences, listed, separately, social, scientists, education, mathematical, postsecondary, science
10	0.016	40	engineering, electrical, equipment, test, design, systems, electronic, electronics, engineers, drafters
11	0.0177	80	health, care, medical, patients, clinical, assist, nursing, individuals, patient, provide
12	0.0263	9	insurance, claims, company, policies, determine, applications, policy, records, information, casualty
13	0.0454	24	financial, credit, loan, securities, individuals, commodities, clerks, records, investment, accounting
14	0.0056	11	fire, forest, fires, prevention, control, fighting, hazards, municipal, forests, enforce
15	0.0157	10	food, workers, serving, preparation, processing, cooking, foods, assist, nutritional, research
16	0.011	26	teachers, teaching, research, courses, primarily, combination, postsecondary, includes, engaged, teach

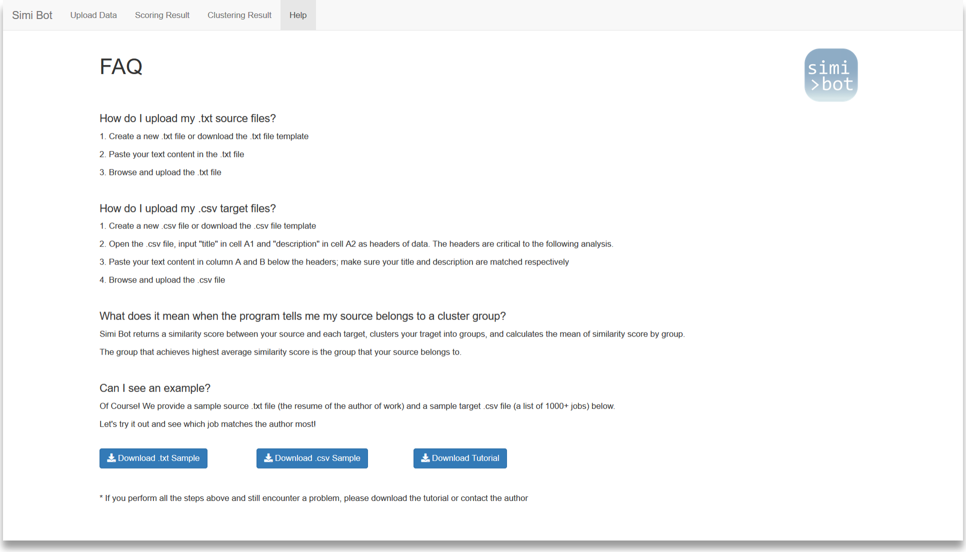
Note. *Clustering result with sample files, screenshot from https://simibot.shinyapps.io/SimiBot/*

The underlying algorithms of this program are as following: cleanse the text source and target, drop all non-alphabetical characters, eliminate multi-space, and lemmatize all the words; apply TF-IDF transformation, compute similarity-score against the source to each target; cluster with hierarchical method; calculate the mean similarity scoring by the group to determine the cluster of the max mean; output data table with cluster group, size of each cluster group, and the keyword in each cluster group.

With this new tool, students studying data analysis and machine learning would have an easy-to-use R tool to perform TF-IDF text similarity scoring analysis. Scenarios that we can put into use include matching resumes to occupations, matching a syllabus to occupations, matching resumes to program syllabi to discover gaps, and recommend courses. Templates, samples, and comprehensive tutorials are provided in the application.

Figure 6

Help Page



FAQ

How do I upload my .txt source files?

1. Create a new .txt file or download the .txt file template
2. Paste your text content in the .txt file
3. Browse and upload the .txt file

How do I upload my .csv target files?

1. Create a new .csv file or download the .csv file template
2. Open the .csv file, input "title" in cell A1 and "description" in cell A2 as headers of data. The headers are critical to the following analysis.
3. Paste your text content in column A and B below the headers; make sure your title and description are matched respectively
4. Browse and upload the .csv file

What does it mean when the program tells me my source belongs to a cluster group?

Simi Bot returns a similarity score between your source and each target, clusters your traget into groups, and calculates the mean of similarity score by group.

The group that achieves highest average similarity score is the group that your source belongs to.

Can I see an example?

Of Course! We provide a sample source .txt file (the resume of the author of work) and a sample target .csv file (a list of 1000+ jobs) below.

Let's try it out and see which job matches the author most!

* If you perform all the steps above and still encounter a problem, please download the tutorial or contact the author

Note. Simi Bot Help Page, screenshot from https://simibot.shinyapps.io/SimiBot/

Limitations, Recommendations and Scope for Future Work

Even this project was able to deliver as excepted, there are still some limitations within this project and some of the limitations may be improved in the future similar projects in NYU MASY.

First of all, this project is developed as a generic text similarity analyzer. It only works for UTF-8 text contents in English. Other languages, non-alphabetical text, or non-ASCII UTF-8 characters would not be analyzed nor get the accurate result. Future work could enable the functionally with other languages, non-alphabetical text, and/or non-ASCII UTF-8 characters. Also, user-defined keywords, stop words, or other customized parameters should be added to enhance the functionality of this application.

Second, the current front-end provides limited responsiveness and consistency. For future work, when users click “Analyze Data”, the program should automatically redirect the frond-end page to the result tabs. Meanwhile, code should be run, and output should be rendered. In the end, there are lots of responsive fold, hover, highlight actions that could be added to this application using JavaScript. Future work should focus on integrating JavaScript with Shiny to provide a better user experience.

Third, the current algorithm of the clustering is a basic method using TF-IDF and cosine distance, which may not be the most efficient one. Future work should try other ways to cluster documents, such as K-means clustering. After comparing different clustering methods with rigid statistic metrics, we should select the most efficient one to implement in this application. The side product of using a better clustering algorithm could be shortening the overall running time.

Forth, future work should focus on accepting more types of input. The current application only intakes the .txt file as the source and the .csv file as the target. Although the application provides templates on the home page, people might not feel handy to use them. In the future. In the future, at least the source input should be a text box allowing users to paste their text source directly. For the target text, we should add a button asking users what method they would like to upload their target text. Options could be "Paste The Target Text One By One" and "Upload A Target File". For the first option, the applications should allow users to paste the target text into the text box. If users would like to add one more target for comparing, they could click the "Add One More Target Text" button. A blank text box should show up after the click. This option is suitable for users who only have a few targets. For the second option, the applications should allow users to upload a .csv file or other compatible excel file.

Literature Survey

Introduction

This literature review was organized to explain the methodologies and solutions toward text data mining, especially in text-similarity scoring. The concept of text matching and relevance ranking appeared for a long time, even earlier than the invention of the Internet. As we know, text similarity scoring and ranking are crucial to today's decision-making, such as matching proper jobs versus an applicant's resume, matching proper courses with a faculty's resume, matching proper papers with a writer's proposal. We are going to develop a general, multifunctional text similarity scoring tool. This tool receives parameter from users, a text file and a .csv file with a list of text, then score the target list of text with TF-IDF (term frequency and inverse document frequency) method. After processing, this tool will generate a new .csv file with a similarity score following each line of the target text, respectively.

There are different approaches to conduct similarity classification: TF–IDF, LDA, and Doc2Vec (Kim, 2019). According to their findings, these three methods have their pros and cons. They suggest trying all methods and take the result of the best performed one, which is called multi-co-training (MCT), for document classification. We would not choose this approach because the scope of this tool is not for document-length text analytics.

Robinson (2021) has explained what is TF-IDF thoroughly on his website. It also contains an explicit tutorial for how to conduct TF-IDF text comparison with R. The author claimed that TF-IDF allows us to find words that are characteristic for one document within a collection of documents, whether that document is a novel or physics text, or webpage. Also, the *tidytext* package in R enables us to see how different words are important in documents within a collection or corpus of documents.

Our work is based on previous text data mining tools matching student's program curriculums to job descriptions (Fortino, Zhong, Huang, & Lowrance, 2019). This paper demonstrates how a Python-based tool scores different text contents with the TF-IDF method. The goal of my R-based tool is to mimic the functionality of the previous tool. Also, it offers a vision of comparing LSI vs. non-LSI algorithms and concluded that non-LSI algorithms had a higher match rate in the samples (curriculums vs. job descriptions). Other similar papers that conducted a likely process on course-faculty matching showed that this tool can significantly improve assignment effectiveness, hiring effectiveness, and ascertain if the assignments were in the right discipline for each faculty member (Fortino, Zhong, Yeh & Fang, 2020).

Another important reference from Fortino, Zhong, Yeh & Fang (2020) researched the methodology of text scoring with Python program to enhance the literature search process. The target text contents are literature papers in this case. Researchers conducted self-report surveys and blind random tests for the feedback of the tool. They found that the tool-using section of students reported significantly less time to do the literature search, and the quality of their literature review produced had a significantly higher quality (Fortino et al. 2020). This paper offers a good overview of what validation test we need to do for my tool.

Other researchers analyzed different types of text contents. Beel & Gipp (2009) examines the relationship between an article's citation count and its ranking in Google Scholar and found that the ranking result is influenced by the impact of age, search term occurrence in full text, search term frequency in full text. Citation count and keyword in the title are heavily weighted on the ranking in Google Scholar while frequency in terms that occur in the full text would not be considered (Beel & Gipp, 2009). Dai et al. (2019) used literature mining methods to discover that the GNB3, CNR1, MTHFR, and NCAM1 genes were identified and directly or indirectly implicated in the regulation of MI and depression. Inan, E. (2020) used a lexicon-based embedding model which is a new version of the linked open data resource ConceptNet to capture the semantics of word sequences and dependency parse tree information of sentences without using any supervised learning methods. Jayasudha & Christina Esther (2019) claimed that frequent pattern growth algorithm shows the effective result for sequential analyses of the data. Unfortunately, the methods mentioned above do not apply to the work we are going to build in this project.

Conclusions

The existing research around one-to-multiple text similarity scoring is target-specific, such as curriculum to faculties (Fortino, Zhong, Yeh & Fang, 2020), curriculum to job descriptions (Fortino, Zhong, Huang, & Lowrance, 2019), an exemplar to references (Fortino, Zhong, Yeh & Fang, 2020).

To examine the matching rate of the sample data, researchers took different surveys or random tests and analyzed the results on the statistical bias (such as t-test). If the result passes a certain confidence level, it means the tool or the program can elevate the efficiency of searching and matching text contents.

Most of the research reviewed concluded it would be best to try different methods out and choose the most robust result. However, that is not feasible for this program. After rationality, we select TF-IDF as our methodology to program the tool in our project.

References

Beel, J., & Gipp, B. (2009). Google Scholar's ranking algorithm: an introductory overview. In Proceedings of the 12th International Conference on Scientometrics and Informetrics (ISSI'09) (Vol. 1, pp. 230-241).

Dai, Z., Li, Q., Yang, G., Wang, Y., Liu, Y., Zheng, Z., Yu, B. (2019). Using literature-based discovery to identify candidate genes for the interaction between myocardial infarction and depression. BMC Medical Genetics, 20(1). https://doi.org/10.1186/s12881-019-0841-8

Fortino, A., Zhong, Q., Yeh, L., Fang, S. (2020). Selection and Assignment of STEM Adjunct Faculty Using Text Data Mining, IEEE ISEC'20 Conference, Princeton University, NJ, August 1, 2020.

Fortino, A., Zhong, Q., Huang, W., Lowrance, R. (2019). Application of Text Data Mining to STEM Curriculum Selection and Development, Awarded Best Paper Award, IEEE ISEC'19 Conference, Princeton University, NJ, March 2019.

Fortino, A., Zhong, Q., Yeh, L., Fang, S. (2020). Using Text Data Mining to Enhance the Literature Search Process for Novice Researchers, IEEE ISEC'20 Conference, Princeton University, NJ, August 1, 2020.

Inan, E. (2020). SimiT: A Text Similarity Method Using Lexicon and Dependency Representations. New Generation Computing, 38(3), 509–530. https://doi.org/10.1007/s00354-020-00099-8

Jayasudha, J., Christina Esther, A. (2019) Mining Sequential Pattern of Data in Textual Document Using Data Mining Classification Technique. Asian Journal of Computer

Science and Technology (AJCST), Volume 8 No.1 Special Issue: February 2019 pp 41-45. https://www.trp.org.in/issues/mining-sequential-pattern-of-data-in-textual-document-using-data-mining-classification-technique.

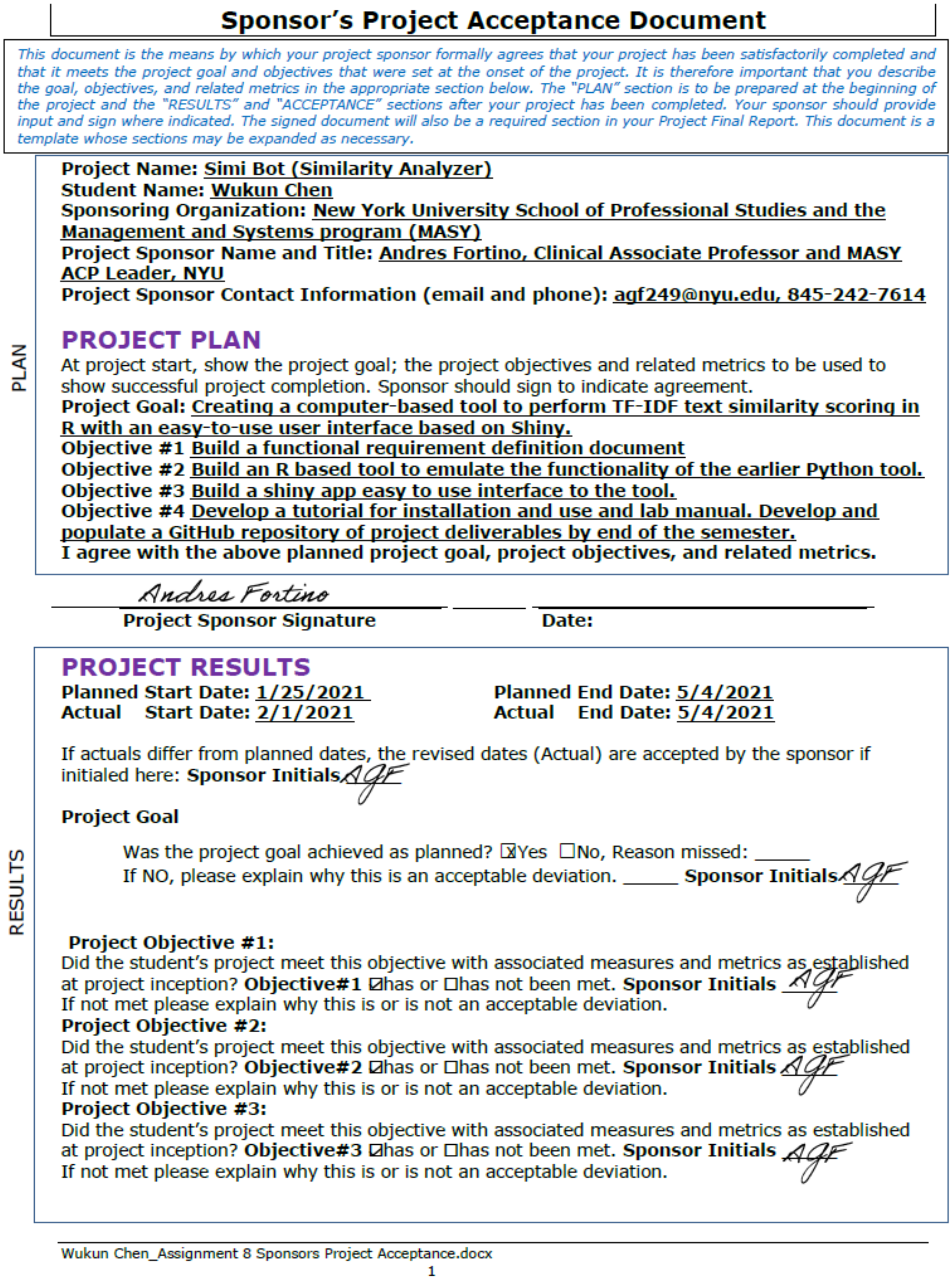
Kim, D., Seo, D., Cho, S., & Kang, P. (2019). Multi-co-training for document classification using various document representations: TF–IDF, LDA, and Doc2Vec. Information Sciences, 477, 15-29.

Meo, S. A., &amp; Talha, M. (2019). Turnitin: Is it a text matching or plagiarism detection tool? Saudi Journal of Anaesthesia, 13(5), 48. https://doi.org/10.4103/sja.sja_772_18

Robinson, J. S., and D. Text Mining with R: A Tidy Approach. 3 Analyzing word and document frequency: tf-idf | Text Mining with R. https://www.tidytextmining.com/tfidf.html

Appendix A

Project Acceptance Document



Sponsor's Project Acceptance Document

This document is the means by which your project sponsor formally agrees that your project has been satisfactorily completed and that it meets the project goal and objectives that were set at the onset of the project. It is therefore important that you describe the goal, objectives, and related metrics in the appropriate section below. The "PLAN" section is to be prepared at the beginning of the project and the "RESULTS" and "ACCEPTANCE" sections after your project has been completed. Your sponsor should provide input and sign where indicated. The signed document will also be a required section in your Project Final Report. This document is a template whose sections may be expanded as necessary.

PLAN

Project Name: Simi Bot (Similarity Analyzer)
Student Name: Wukun Chen
Sponsoring Organization: New York University School of Professional Studies and the Management and Systems program (MASY)
Project Sponsor Name and Title: Andres Fortino, Clinical Associate Professor and MASY ACP Leader, NYU
Project Sponsor Contact Information (email and phone): agf249@nyu.edu, 845-242-7614

PROJECT PLAN

At project start, show the project goal; the project objectives and related metrics to be used to show successful project completion. Sponsor should sign to indicate agreement.

Project Goal: **Creating a computer-based tool to perform TF-IDF text similarity scoring in R with an easy-to-use user interface based on Shiny.**
Objective #1 **Build a functional requirement definition document**
Objective #2 **Build an R based tool to emulate the functionality of the earlier Python tool.**
Objective #3 **Build a shiny app easy to use interface to the tool.**
Objective #4 **Develop a tutorial for installation and use and lab manual. Develop and populate a GitHub repository of project deliverables by end of the semester.**
I agree with the above planned project goal, project objectives, and related metrics.

Andres Fortino
Project Sponsor Signature **Date:**

RESULTS

PROJECT RESULTS

Planned Start Date: 1/25/2021 **Planned End Date:** 5/4/2021
Actual Start Date: 2/1/2021 **Actual End Date:** 5/4/2021

If actuals differ from planned dates, the revised dates (Actual) are accepted by the sponsor if initialed here: **Sponsor Initials** AGF

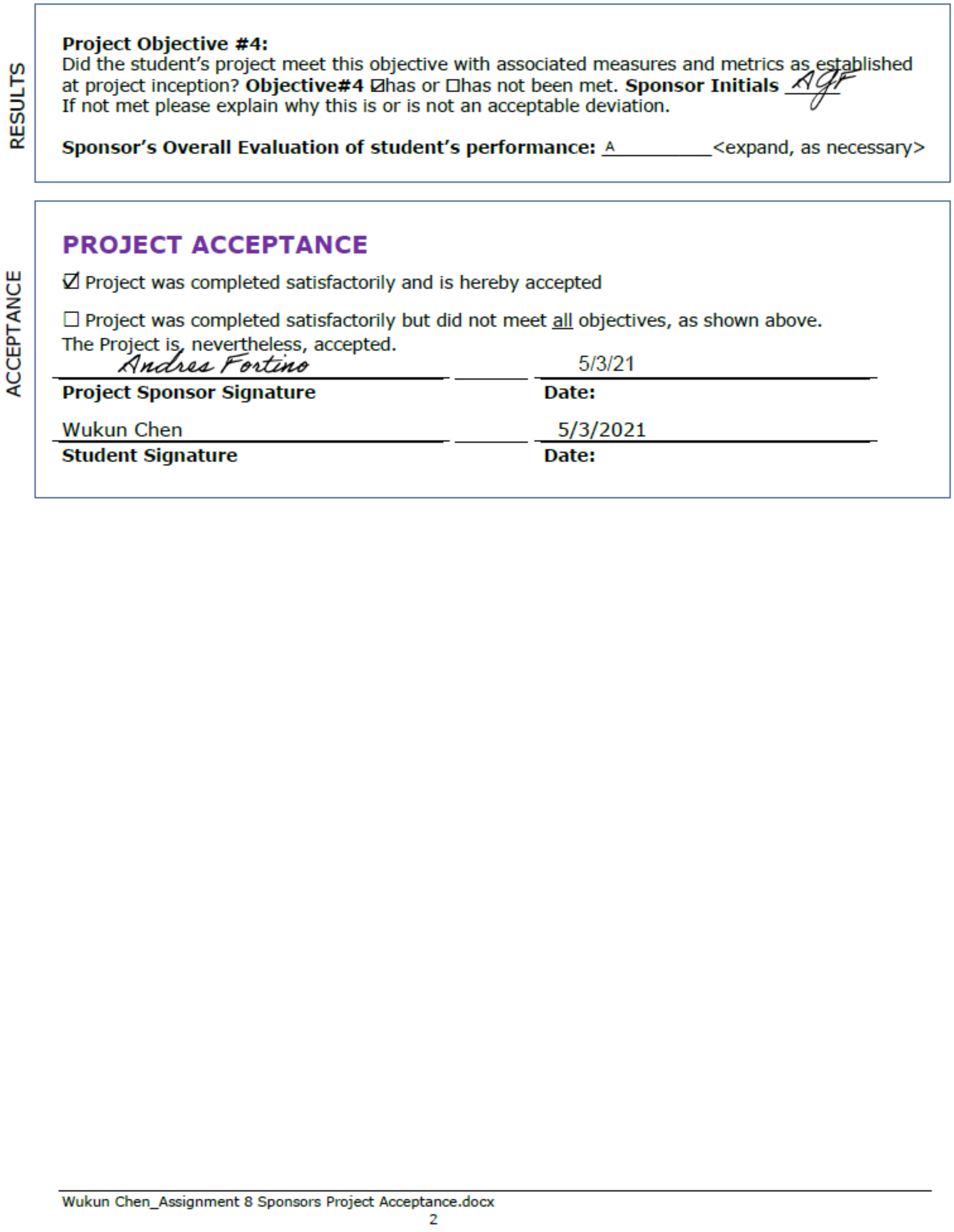
Project Goal

Was the project goal achieved as planned? ☒Yes ☐No, Reason missed: _____
If NO, please explain why this is an acceptable deviation. _____ **Sponsor Initials** AGF

Project Objective #1:
Did the student's project meet this objective with associated measures and metrics as established at project inception? **Objective#1** ☑has or ☐has not been met. **Sponsor Initials** AGF
If not met please explain why this is or is not an acceptable deviation.

Project Objective #2:
Did the student's project meet this objective with associated measures and metrics as established at project inception? **Objective#2** ☑has or ☐has not been met. **Sponsor Initials** AGF
If not met please explain why this is or is not an acceptable deviation.

Project Objective #3:
Did the student's project meet this objective with associated measures and metrics as established at project inception? **Objective#3** ☑has or ☐has not been met. **Sponsor Initials** AGF
If not met please explain why this is or is not an acceptable deviation.



RESULTS

Project Objective #4:
Did the student's project meet this objective with associated measures and metrics as established at project inception? **Objective#4** ☑has or ☐has not been met. **Sponsor Initials** AGF
If not met please explain why this is or is not an acceptable deviation.

Sponsor's Overall Evaluation of student's performance: A <expand, as necessary>

ACCEPTANCE

PROJECT ACCEPTANCE

☑ Project was completed satisfactorily and is hereby accepted

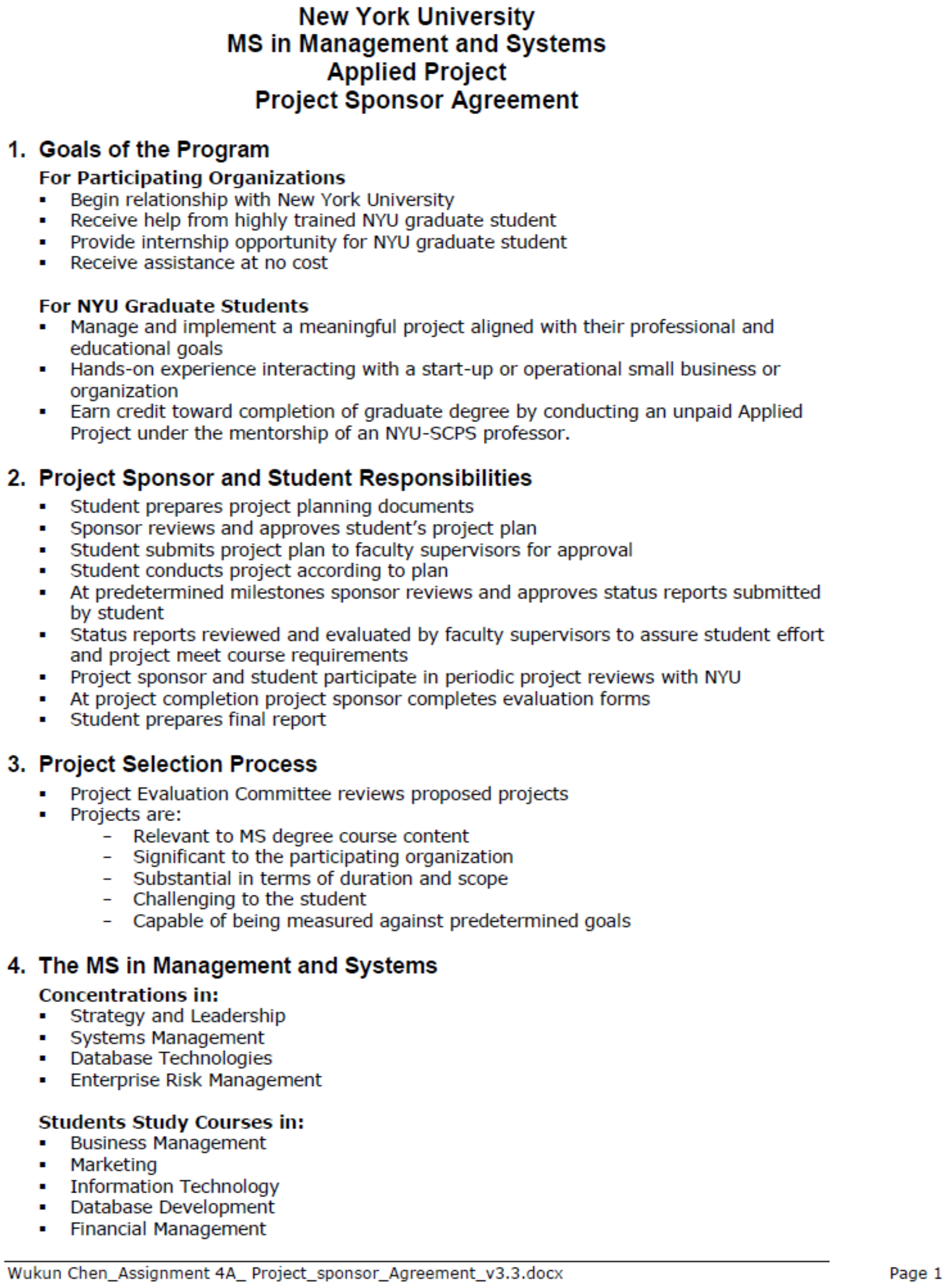
☐ Project was completed satisfactorily but did not meet all objectives, as shown above. The Project is, nevertheless, accepted.

Andres Fortino — 5/3/21
Project Sponsor Signature — **Date:**

Wukun Chen — 5/3/2021
Student Signature — **Date:**

Appendix B

Project Sponsor Agreement



New York University
MS in Management and Systems
Applied Project
Project Sponsor Agreement

1. Goals of the Program

For Participating Organizations

- Begin relationship with New York University
- Receive help from highly trained NYU graduate student
- Provide internship opportunity for NYU graduate student
- Receive assistance at no cost

For NYU Graduate Students

- Manage and implement a meaningful project aligned with their professional and educational goals
- Hands-on experience interacting with a start-up or operational small business or organization
- Earn credit toward completion of graduate degree by conducting an unpaid Applied Project under the mentorship of an NYU-SCPS professor.

2. Project Sponsor and Student Responsibilities

- Student prepares project planning documents
- Sponsor reviews and approves student's project plan
- Student submits project plan to faculty supervisors for approval
- Student conducts project according to plan
- At predetermined milestones sponsor reviews and approves status reports submitted by student
- Status reports reviewed and evaluated by faculty supervisors to assure student effort and project meet course requirements
- Project sponsor and student participate in periodic project reviews with NYU
- At project completion project sponsor completes evaluation forms
- Student prepares final report

3. Project Selection Process

- Project Evaluation Committee reviews proposed projects
- Projects are:
  - Relevant to MS degree course content
  - Significant to the participating organization
  - Substantial in terms of duration and scope
  - Challenging to the student
  - Capable of being measured against predetermined goals

4. The MS in Management and Systems

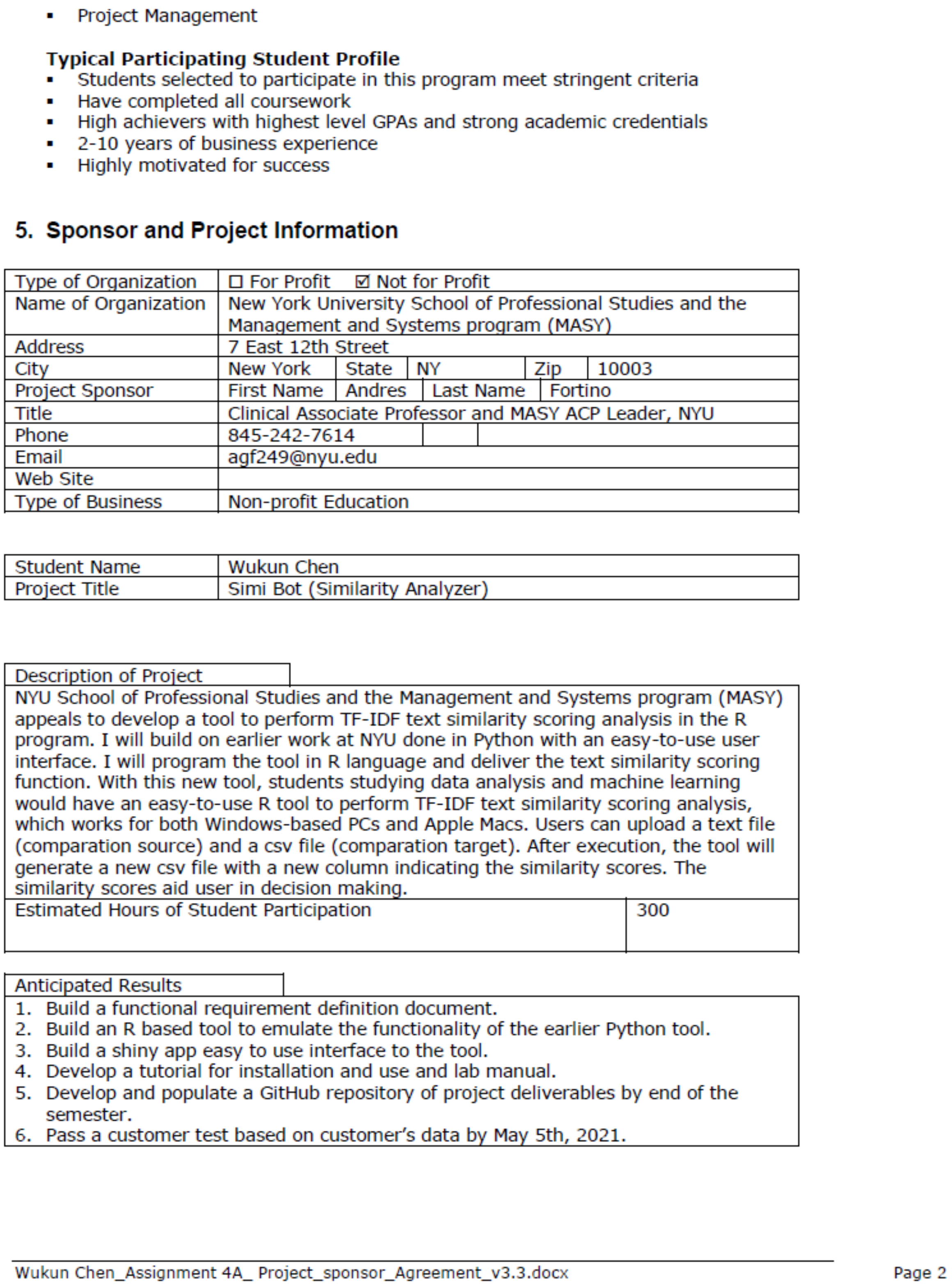
Concentrations in:

- Strategy and Leadership
- Systems Management
- Database Technologies
- Enterprise Risk Management

Students Study Courses in:

- Business Management
- Marketing
- Information Technology
- Database Development
- Financial Management

Wukun Chen_Assignment 4A_ Project_sponsor_Agreement_v3.3.docx Page 1



- Project Management

Typical Participating Student Profile

- Students selected to participate in this program meet stringent criteria
- Have completed all coursework
- High achievers with highest level GPAs and strong academic credentials
- 2-10 years of business experience
- Highly motivated for success

5. Sponsor and Project Information

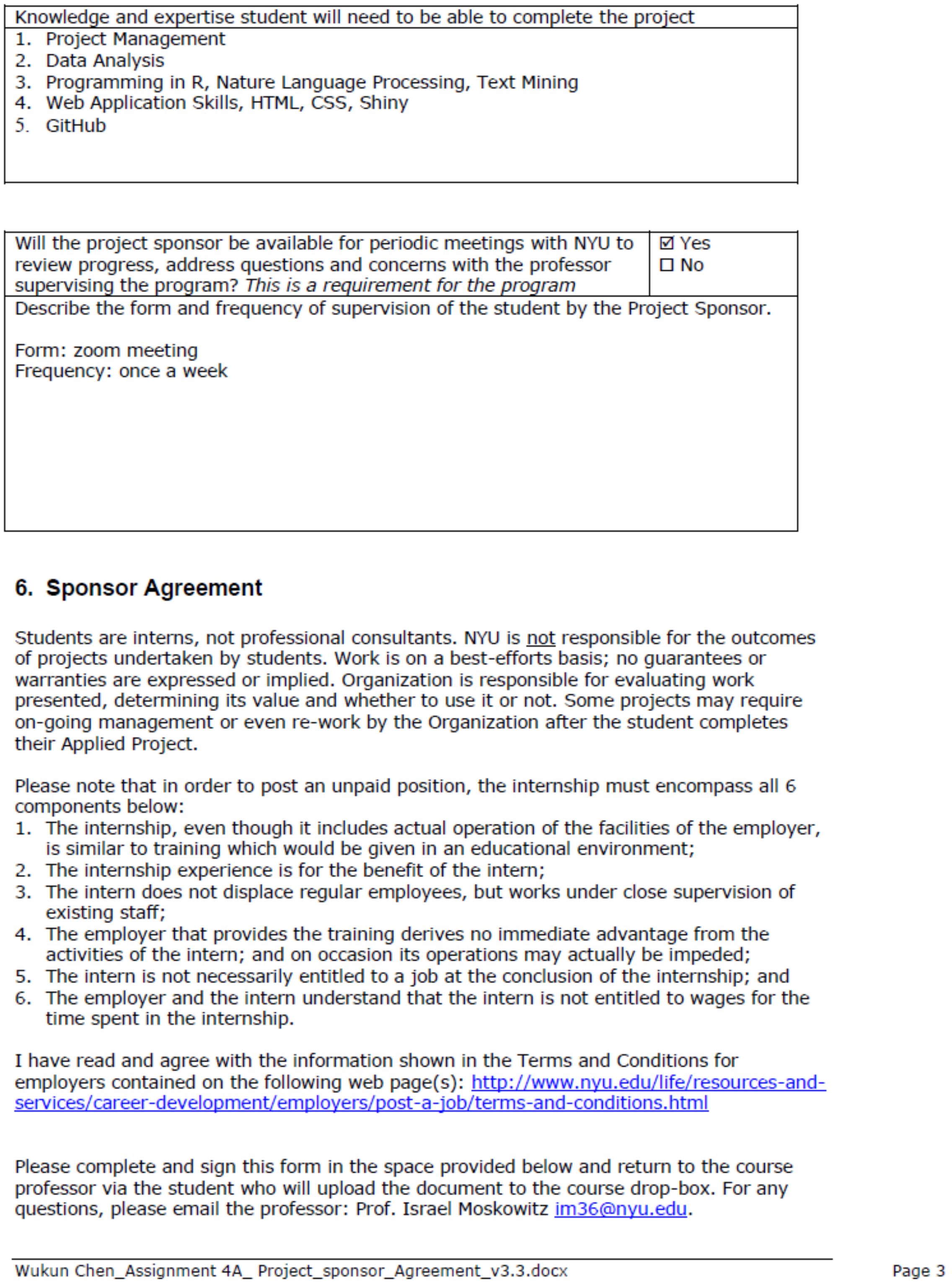
Type of Organization	☐ For Profit ☑ Not for Profit				
Name of Organization	New York University School of Professional Studies and the Management and Systems program (MASY)				
Address	7 East 12th Street				
City	New York	State	NY	Zip	10003
Project Sponsor	First Name	Andres	Last Name	Fortino	
Title	Clinical Associate Professor and MASY ACP Leader, NYU				
Phone	845-242-7614				
Email	agf249@nyu.edu				
Web Site					
Type of Business	Non-profit Education				

Student Name	Wukun Chen
Project Title	Simi Bot (Similarity Analyzer)

Description of Project	
NYU School of Professional Studies and the Management and Systems program (MASY) appeals to develop a tool to perform TF-IDF text similarity scoring analysis in the R program. I will build on earlier work at NYU done in Python with an easy-to-use user interface. I will program the tool in R language and deliver the text similarity scoring function. With this new tool, students studying data analysis and machine learning would have an easy-to-use R tool to perform TF-IDF text similarity scoring analysis, which works for both Windows-based PCs and Apple Macs. Users can upload a text file (comparation source) and a csv file (comparation target). After execution, the tool will generate a new csv file with a new column indicating the similarity scores. The similarity scores aid user in decision making.	
Estimated Hours of Student Participation	300

Anticipated Results

1. Build a functional requirement definition document.
2. Build an R based tool to emulate the functionality of the earlier Python tool.
3. Build a shiny app easy to use interface to the tool.
4. Develop a tutorial for installation and use and lab manual.
5. Develop and populate a GitHub repository of project deliverables by end of the semester.
6. Pass a customer test based on customer’s data by May 5th, 2021.



Knowledge and expertise student will need to be able to complete the project
1. Project Management 2. Data Analysis 3. Programming in R, Nature Language Processing, Text Mining 4. Web Application Skills, HTML, CSS, Shiny 5. GitHub

Will the project sponsor be available for periodic meetings with NYU to review progress, address questions and concerns with the professor supervising the program? *This is a requirement for the program*	☑ Yes ☐ No
Describe the form and frequency of supervision of the student by the Project Sponsor. Form: zoom meeting Frequency: once a week	

6. Sponsor Agreement

Students are interns, not professional consultants. NYU is not responsible for the outcomes of projects undertaken by students. Work is on a best-efforts basis; no guarantees or warranties are expressed or implied. Organization is responsible for evaluating work presented, determining its value and whether to use it or not. Some projects may require on-going management or even re-work by the Organization after the student completes their Applied Project.

Please note that in order to post an unpaid position, the internship must encompass all 6 components below:

1. The internship, even though it includes actual operation of the facilities of the employer, is similar to training which would be given in an educational environment;
2. The internship experience is for the benefit of the intern;
3. The intern does not displace regular employees, but works under close supervision of existing staff;
4. The employer that provides the training derives no immediate advantage from the activities of the intern; and on occasion its operations may actually be impeded;
5. The intern is not necessarily entitled to a job at the conclusion of the internship; and
6. The employer and the intern understand that the intern is not entitled to wages for the time spent in the internship.

I have read and agree with the information shown in the Terms and Conditions for employers contained on the following web page(s): http://www.nyu.edu/life/resources-and-services/career-development/employers/post-a-job/terms-and-conditions.html

Please complete and sign this form in the space provided below and return to the course professor via the student who will upload the document to the course drop-box. For any questions, please email the professor: Prof. Israel Moskowitz im36@nyu.edu.

I agree to the all of the above

Participating Organization New York University School of Professional Studies and the Management and Systems program (MASY) Date 3/8/2021

By (signature): Andres Fortino
Project Sponsor

Printed Name: Dr. Andres Fortino

Title: Clinical Associate Professor of Management and Systems

7. Student Agreement

Students who are planning to conduct an unpaid Applied Project must read and agree to the "Important Considerations Before Accepting a Job or Internship" contained on the following web page(s): http://www.nyu.edu/life/resources-and-services/career-development/find-a-job-or-internship/important-considerations-before-accepting-a-job-or-internship.html.

Students do not register their Applied Project with the Wasserman Center.

I agree to the all of the above

Student Name (Print) Wukun Chen Date 3/8/2021

Signature: Wukun Chen

Appendix C

Project Charter

Simi Bot (Similarity Analyzer) Project Charter

Project Manager: Wukun Chen
Sponsor: Dr. Andres Fortino
Prepared by: Wukun Chen
Name and Location of Client Organization
New York University (NYU) School of Professional Studies and the Management and Systems program (MASY)
7 East 12th Street, New York, NY 10003

Revision History

Revision date	Revised by	Approved by	Description of change

Project Goal

The goal of the project is to help NYU SPS creating a computer-based tool to perform TF-IDF text similarity scoring in R with an easy-to-use user interface.

Problem/Opportunity Definition

We came across lots of information arrangements based on text similarity, such as matching one resume to multiple job descriptions or matching one faculty profile to multiple course descriptions. NYU MASY wishes to develop a tool to perform TF-IDF text similarity scoring analysis in the R program. They have earlier work done in Python with an easy-to-use user interface with Heroku. The problems of the previous work include long webpage loading time, incompatible file type, relying on web connection, etc. We now wish to complete the development of text analytic tools and make sure the tool works for both Windows-based PCs and Apple Macs.

Proposed Project Description

NYU School of Professional Studies and the Management and Systems program (MASY) appeals to develop a tool to perform TF-IDF text similarity scoring analysis in the R program. I will build on earlier work at NYU done in Python with an easy-to-use user interface. I will program the tool in R language and deliver the text-similarity scoring function. With this new tool, students studying data analysis and machine learning would have an easy-to-use R tool to perform TF-IDF text similarity scoring analysis, which works for both Windows-based PCs and Apple Macs. Users can upload a text file (comparison source) and a .csv file (comparison target). After execution, the tool will generate a new .csv file with a new column indicating the similarity scores. The similarity scores aid user in decision-making.

Project Sponsor

- Name and Title: Dr. Andres Fortino, Clinical Associate Professor and MASY ACP Leader, NYU
- Role within the organization: Dr. Fortino is the Clinical Associate Professor of Management and Systems major and the Academic Community of Practice Leader at NYU School of Professional Studies
- Role on the project: Dr. Fortino is not only the sponsor for this project but also my mentor in conducting the project. He supervises my work and gives me the necessary resources and guidance.

Objectives:

Technical Objectives:

- Delivery a working R script, a working shiny interface, and populate a GitHub repository of the project by end of the semester; Pass a customer test based on customer's data by end of the semester

Timing objectives

- Complete the entire project before May 4th, 2021

Resource objectives:

- Complete the project with free educational resource online (using free LinkedIn Learning access granted by NYU)

Budget objectives

- Zero budget (using free open-source programming languages and tools)

Budget objectives:

	Planned	**Actual**
Salaries	0	0
Documentation	0	0
Construction	0	0
Mover	0	0
Total	**$0**	0

Scope objectives:

- Help NYU School of Professional Studies and the Management and Systems program complete this one-to-many similarity analyzer and launch it on a public server. Deliver the final report before May 4th, 2021.

Project Selection & Ranking Criteria

Project benefit category:

□ Compliance/Regulatory ☑ Efficiency/Cost reduction □ Revenue increase

Portfolio fit and interdependencies

Projects across the company will be affected by this project

Project urgency

High

Cost/Benefit Analysis

Tangible Benefits

Benefit: N/A

Value & Probability: N/A

Assumptions Driving Value: N/A

Intangible Benefits

Benefit: Help business make easy decisions, help students to match right job faster, help the school to match appropriate faculty with courses

Value & Probability: N/A

Assumptions Driving Value: N/A

Cost Categories	**Amount**
Internal Labor hours	*300 hours*
External costs -------------	
Labor (consultants, contract labor)	N/A
Equipment, hardware, or software	N/A
List other costs such as travel & training	N/A

Financial Return

N/A

Other Business Benefits

With this new tool, students studying data analysis and machine learning would have an easy-to-use R tool to perform TF-IDF text similarity scoring analysis, which works for both Windows-based PCs and Apple Macs.

Assumptions

1. The source code provided by Dr. Fortino is accurate.
2. Text similarity scoring is needed for business and there is still room for development.
3. Dr. Fortino and the School of Professional Studies are willing to sponsor this similarity analyzer and publish the product for free.

Scope

- ■ **Quality**
  - Build an R based tool to accomplish the task of similarity scoring
  - Build a shiny app with an easy-to-use interface to include this tool
  - Develop a tutorial for installation and use and a lab manual
  - Develop and populate a GitHub repository of project deliverables
  - Pass a customer test based on customer's data with high accuracy
- ■ **Time**
  - Finalize the R code at the end of March
  - Create the Shiny web interface at the beginning of April
  - Finalized the report before May 4th, 2021
- ■ **Resource Allocation**
  - Utilize the available resource to collect accurate industry information
  - Leverage resources of NYU to integrate the previous accomplishment by other students and the frontier trend

Out of scope activities

- Deliver multi-source to multi-target comparison
- Deliver clustering function

Constraints

1. The limitations of time, budget, and scope result in limited functionality of this tool. For instance, this tool is only compatible with .txt and .csv file. Further compatibility requires more investment.

2. This tool is designed for general text similarity analysis regardless of the length and content of the text. This tool does not reduce the weight of frequent words in the text during the scoring process.

3. This tool will be hosted on the Shiny server. Server response times are up to the capacity of Shiny.io.

Risks and Mitigation Strategies

Risk: Since this tool will be hosted on the Shiny server, the performance of this tool relies on the capacity of Shiny. The worst scenario is this tool will be purged from the Shiny server if Shiny terminates the service.

Mitigation Strategies: This tool will be open for download. The source code will be synced to a GitHub repository.

Communications Plan

1. Frequency: Once per week
2. Method: Aligning with the COVID-19 protocol, we communicate via email and Zoom
3. Content: Weekly status report; Milestone notification; Project/Program update; Calling for assistance; Ad hoc request

Schedule Overview

Project Start Date: 2/1/2021, Monday

Estimated Project Completion Date: 5/4/2021, Tuesday.

Major Milestones

Project kick-off: Define specific objectives for the project - 2/4/2021

Milestone 1: The project plan is approved, and the project manager has permission to proceed to execute the project according to the project plan - 2/15/2021

Milestone 2: Team installs a development system for testing and customizations of user interfaces - 3/29/2021

Deliverable: Publishing the tool on the server and submit the final report - 5/4/2021

Impact of Late Delivery

If this project is delivered late, the following research based on this project (other students' capstone project) will be postponed.

Resources Required

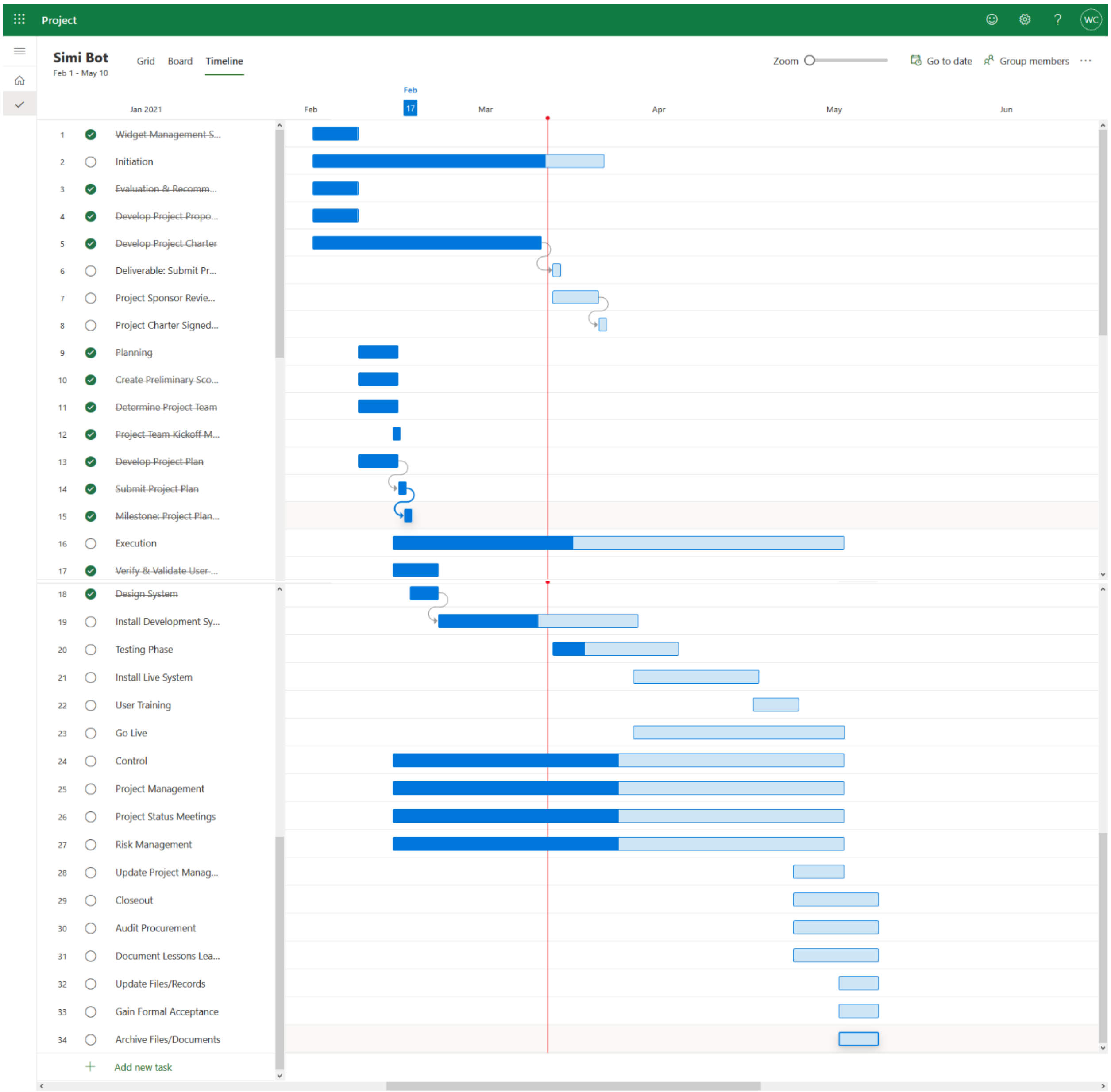
Role	Responsibilities	Duration of work	Qualifications needed
Consultant	Insight into text similarity analyzing approach and develop software based on R language	14 weeks	Information collection and integration capabilities; R programming skills
Web Developer	Build a Shiny application on the server	2 weeks	R programming and web design skills (HTML)

Facilities, Software, Hardware, and Other Resources

A computer is offered by the project manager. Software engaged (R, RStudio, Shiny) are open-sourced, which could be download for free from the internet. Industry analysis reports (IBISWorld) have been purchased by New York University, which could be accessed for free via the NYU library.

Procedures/ Methodology

Project Plan, Gantt, Resource Loading, etc.



1	✓	~~Widget Management S...~~
2	○	Initiation
3	✓	~~Evaluation & Recomm...~~
4	✓	~~Develop Project Propo...~~
5	✓	~~Develop Project Charter~~
6	○	Deliverable: Submit Pr...
7	○	Project Sponsor Revie...
8	○	Project Charter Signed...
9	✓	~~Planning~~
10	✓	~~Create Preliminary Sco...~~
11	✓	~~Determine Project Team~~
12	✓	~~Project Team Kickoff M...~~
13	✓	~~Develop Project Plan~~
14	✓	~~Submit Project Plan~~
15	✓	~~Milestone: Project Plan...~~
16	○	Execution
17	✓	~~Verify & Validate User...~~
18	✓	~~Design System~~
19	○	Install Development Sy...
20	○	Testing Phase
21	○	Install Live System
22	○	User Training
23	○	Go Live
24	○	Control
25	○	Project Management
26	○	Project Status Meetings
27	○	Risk Management
28	○	Update Project Manag...
29	○	Closeout
30	○	Audit Procurement
31	○	Document Lessons Lea...
32	○	Update Files/Records
33	○	Gain Formal Acceptance
34	○	Archive Files/Documents
	+	Add new task

Project Evaluation

1. **Project schedule:** Update and report the project schedule to the sponsor weekly.
2. **Project weekly status report and dashboard:** Report and document project progress weekly; mark important project milestones.
3. **Project communication plan, issues log, risk register:** Confirm the feasibility of communication to ensure the efficiency of communications; discuss problems and risks with the sponsor and carry out feasible solutions.
4. **Project monthly status report:** A comprehensive reporting meeting will be held on a monthly bias. The meeting focused on the progress of the project and estimated whether it could be delivered as scheduled.

Appendix D

Project Plan

Project Tasks Outline

OUTLINE VIEW

2. Widget Management System
   - 2.1 Initiation
     - 2.1.1 Evaluation & Recommendations
     - 2.1.2 Develop Project Proposal
     - 2.1.3 Develop Project Charter
     - 2.1.4 Deliverable: Submit Project Charter
     - 2.1.5 Project Sponsor Reviews Project Charter
     - 2.1.6 Project Charter Signed/Approved
   - 2.2 Planning
     - 2.2.1 Create Preliminary Scope Statement
     - 2.2.2 Determine Project Team
     - 2.2.3 Project Team Kickoff Meeting
     - 2.2.4 Develop Project Plan
     - 2.2.5 Submit Project Plan
     - 2.2.6 Milestone: Project Plan Approval
   - 2.3 Execution
     - 2.3.1 Verify & Validate User Requirements
     - 2.3.2 Design System
     - 2.3.3 Install Development System
     - 2.3.4 Testing Phase
     - 2.3.5 Install Live System
     - 2.3.6 User Training
     - 2.3.7 Go Live
   - 2.4 Control
     - 2.4.1 Project Management
     - 2.4.2 Project Status Meetings
     - 2.4.3 Risk Management
     - 2.4.4 Update Project Management Plan
   - 2.5 Closeout
     - 2.5.1 Audit Procurement
     - 2.5.2 Document Lessons Learned
     - 2.5.3 Update Files/Records
     - 2.5.4 Gain Formal Acceptance
     - 2.5.5 Archive Files/Documents

Work Breakdown Task Definition and Schedule

Level	WBS Code	Element Name	Definition	Due By
1	1	Widget Management System	All work to implement a new widget management system.	2/8/2021
2	1.1	Initiation	The work to initiate the project.	2/15/2021
3	1.1.1	Evaluation & Recommendations	Working group to evaluate solution sets and make recommendations.	2/8/2021
3	1.1.2	Develop Project Proposal	Project Manager to develop the Project Proposal.	2/8/2021
3	1.1.3	Develop Project Charter	Project Manager to develop the Project Charter.	2/8/2021
3	1.1.4	Deliverable: Submit Project Charter	Project Charter is delivered to the Project Sponsor.	2/8/2021
3	1.1.5	Project Sponsor Reviews Project Charter	The project sponsor reviews the Project Charter.	2/8/2021
3	1.1.6	Project Charter Signed/Approved	The Project Sponsor signs the Project Charter which authorizes the Project Manager to move to the Planning Process.	2/8/2021
2	1.2	Planning	The work for the planning process for the project.	2/15/2021
3	1.2.1	Create Preliminary Scope Statement	The Project Manager creates a Preliminary Scope Statement.	2/15/2021
3	1.2.2	Determine Project Team	The Project Manager determines the project team and requests the resources.	2/15/2021
3	1.2.3	Project Team Kickoff Meeting	The planning process is officially started with a project kickoff meeting which includes the Project Manager, Project Team, and Project Sponsor (optional).	2/15/2021
3	1.2.4	Develop Project Plan	Under the direction of the Project Manager, the team develops the project plan.	2/15/2021
3	1.2.5	Submit Project Plan	The project Manager submits the project plan for approval.	2/15/2021
3	1.2.6	Milestone: Project Plan Approval	The project plan is approved, and the Project Manager has permission to proceed to execute the project according to the project plan.	2/15/2021
2	1.3	Execution	Work involved executing the project.	5/4/2021
3	1.3.1	Verify & Validate User Requirements	The original user requirements are reviewed by the project manager and team, then validated with the users/stakeholders. This is where additional clarification may be needed.	2/22/2021
3	1.3.2	Design System	The technical resources design the new widget management system.	2/22/2021
3	1.3.3	Install Development System	The team installs a development system for testing and customizations of user interfaces.	3/29/2021

3	1.3.4	Testing Phase	The system is tested with a select set of users.	4/5/2021
3	1.3.5	Install Live System	The actual system is installed and configured.	4/19/2021
3	1.3.6	User Training	All users are provided with four hours training class. Additionally, managers are provided with an additional two-hour class to cover advanced reporting.	4/26/2021
3	1.3.7	Go Live	The system goes live with all users.	5/4/2021
2	1.4	Control	The work involved the control process of the project.	5/4/2021
3	1.4.1	Project Management	Overall project management for the project.	5/4/2021
3	1.4.2	Project Status Meetings	Weekly team status meetings.	5/4/2021
3	1.4.3	Risk Management	Risk management efforts as defined in the Risk Management Plan.	5/4/2021
3	1.4.4	Update Project Management Plan	The project manager updates the Project Management Plan as the project progresses.	5/4/2021
2	1.5	Closeout	The work to close out the project.	5/10/2021
3	1.5.1	Audit Procurement	An audit of all hardware and software procured for the project ensures that all procured products are accounted for and in the asset management system.	5/10/2021
3	1.5.2	Document Lessons Learned	The project manager along with the project team performs a lesson learned meeting and documents the lessons learned for the project.	5/10/2021
3	1.5.3	Update Files/Records	All files and records are updated to reflect the widget management system.	5/10/2021
3	1.5.4	Gain Formal Acceptance	The Project Sponsor formally accepts the project by signing the acceptance document included in the project plan.	5/10/2021
3	1.5.5	Archive Files/Documents	All project-related files and documents are formally archived.	5/10/2021

Appendix E

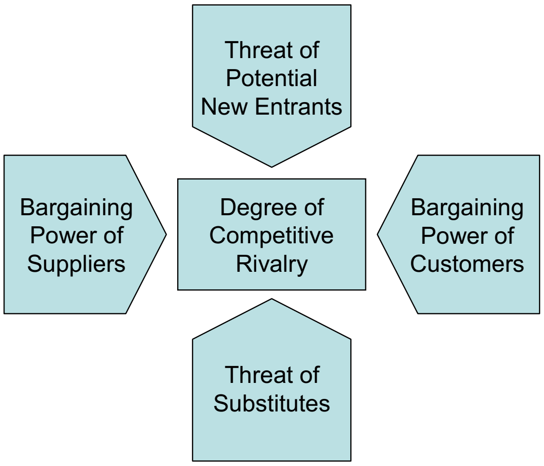
Situational Analysis

Applied Project Situation Analysis

The industry of my project belongs to Software computer, packaged, publishers (NACIS code 511210).

Competitors against whom the business competes include twinword inc., Grammarly, Turnitin, Copyleaks, Prepostseo, Duplichecker.

Our business cultures include user-centered, benefited from open-source, and give back to open-source, relentless optimization.


Threat of Potential New Entrants
Bargaining Power of Suppliers
Degree of Competitive Rivalry
Bargaining Power of Customers
Threat of Substitutes

Porter's Five-Forces Model

Industry Analysis

According to IBIS World, the software publishing industry in the US could be defined as "Software publishers disseminate licenses to customers for the right to execute software on their computers. Operators in this industry market and distribute software products and may also design the software, produce support materials and provide support services"(Cook, 2020).

Software as a service (SaaS) is "a model of software deployment in which a provider licenses an application to customers for use as a service on demand"(Cook, 2020).

This industry reported 293.1-billion-dollar revenue in 2020 and is expected to have 2.4% annual revenue growth in 2020-2025. Profit was reported at $83.2 billion with 5.2% average growth in the past 5 years. The average profit margin in the last 5 years was 28.4% with an outlook of 1.77pp downside(Cook, 2020).

In general, this industry is still in the growth section of the industry life cycle. Capital intensity and regulation are light, which gives us chance to break in as a newbie. However, industry assistance is low and would be steady. The technology change, industrial globalization, and competition are high and would be increasing.

The strength of this industry would be the growth life cycle stage, low imports, low capital requirements. The weaknesses are the low and steady level of assistance, high competition, high customer class concentration, high product and service concentration, and low revenue per employee. Potential opportunities are high revenue growth and investor uncertainty. At last, the threats are low outlier growth, low-performance drivers, and private investment in computers and software.

Competitors

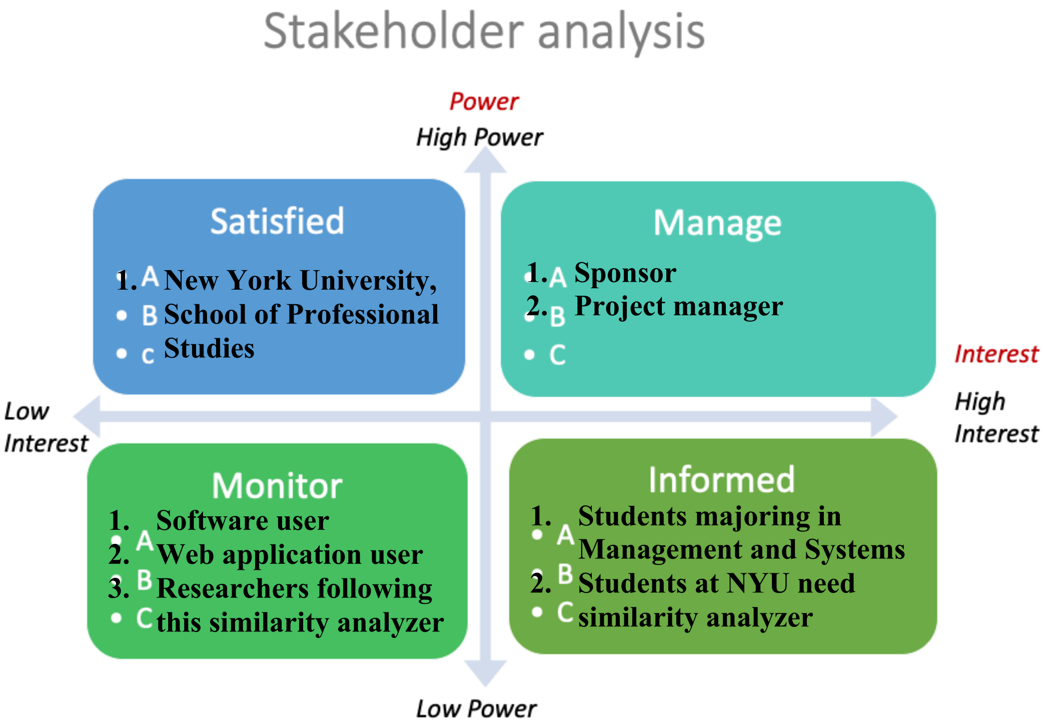
Current competitors in the text-similarity analysis software industry include twinword inc. (Amazon AWS), Grammarly, Turnitin, Copyleaks, Prepostseo, Duplichecker. These web service suppliers already offer mature text similarity analysis products. Potential competitors include Microsoft Corporation, International Business Machines Corporation, Apple Inc., Oracle Corporation, etc. These technology giants are extremely capable and wealthy. They are likely to create disruptive software with sufficient funds.

Stakeholders

List of all stakeholders :

11. Sponsor, Dr. Andres Fortino
12. Project manager, Wukun Chen
13. New York University, School of Professional Studies
14. Teammates
15. Colleagues
16. Students majoring in Management and Systems
17. Students at NYU need a similarity analyzer
18. Software user
19. Web application user
20. Researchers following this similarity analyzer

For internal stakeholders (1-3), their opinion will directly influence the business need and scope of this project. For external stakeholders (4-10), we will regard their opinion as feedback and keep improving this product.



Satisfied

1. New York University, School of Professional Studies

Manage

1. Sponsor
2. Project manager

Monitor

1. Software user
2. Web application user
3. Researchers following this similarity analyzer

Informed

1. Students majoring in Management and Systems
2. Students at NYU need similarity analyzer

References

Cook, D. (2020). Software Publishing in the US. IBISWorld.

Cost/Benefit Analysis

Tangible Benefits

Benefit: N/A

Value & Probability: N/A

Assumptions Driving Value: N/A

Intangible Benefits

Benefit: Help business make easy decisions, help students to match right job faster, help the school to match appropriate faculty with courses

Value & Probability: N/A

Assumptions Driving Value: N/A

Cost Categories	**Amount**
Internal Labor hours	300 hours
External costs -------------	
Labor (consultants, contract labor)	N/A
Equipment, hardware, or software	N/A
List other costs such as travel & training	N/A

Financial Return

Breakeven analysis (if appropriate)

Other Business Benefits

Appendix F

Risk Management Plan

Project
Simi Bot is an R-based similarity analyzer helping the user to sort the target text with a similarity score.

Risks

Number	Risk	Probability Score (1,2 or 3)	Impact Score (1,2 or 3)
1	Cannot deliver on time	2	3
2	Bugs and Error in the Code	2	2
3	Computer Deficiency	1	2
4	Client requests to change the project	2	1
5	Health issue	1	1

Risk Matrix

		RISK (exposure)		
		1.Slight	2. Moderate	3. High
Probability (of occurrence)	1. Very Unlikely	🟩 5	🟩 3	🟨
	2. Possible	🟩 4	🟨 2	🟧 1
	3. Expected	🟩	🟧	🟧

Contingency Plan

Risk	Description	Probability (1-3)	Exposure (1-3)	Contingency Plan
1	Cannot deliver on time	2	3	Fast track the project, leave enough cushion for debugging and troubleshooting
2	Bugs and Error in the Code	2	2	
3	Computer Deficiency	1	2	
4	Client requests to change the project	2	1	
5	Health issue	1	1	

Appendix G

Change Management Plan

PROJECT CHANGE MANAGEMENT PLAN

Project Name:	**Simi Bot**
Prepared by:	**Wukun Chen**
Date (MM/DD/YYYY):	**3/28/2021**

1. Purpose

The purpose of this Change Management Plan *is to:*

- Ensure that all changes to the project are reviewed and approved in advance
- All changes are coordinated across the entire project.
- All stakeholders are notified of approved changes to the project.

All project Change Requests (CR) must be submitted in written form using the Change Request Form provided.	**Link To Project Change Request Form**
The project team should keep a log of all Change Requests.	**Link To Project Change Request Log**

2. Goals

The goals of this Change Management Plan *are to:*

- Give due consideration to all requests for change
- Identify define, evaluate, approve, and track changes through to completion
- Modify Project Plans to reflect the impact of the changes requested
- Bring the appropriate parties (depending on the nature of the requested change) into the discussion
- Negotiate changes and communicate them to all affected parties.

3. Responsibilities

Those responsible for Change Management	*Their Responsibilities*
• Project Manager (with the Project Team)	Developing the Change Management Plan

3. Responsibilities

Those responsible for Change Management	*Their Responsibilities*
• Project Manager	Facilitating or executing the change management process. This process may result in changes to the scope, schedule, budget, and/or quality plans. Additional resources may be required.
• A designated member of the Project Team	Maintaining a log of all CRs
• Project Manager	Conducting reviews of all change management activities with senior management periodically
• The Executive Committee	Ensuring that adequate resources and funding are available to support the execution of the *Change Management Plan* Ensuring that the *Change Management Plan* is implemented

4. Process

The Change Management process occurs in six steps:

1. Submit written Change Request (CR)
2. Review CRs and approve or reject for further analysis
3. If approved, perform analysis, and develop a recommendation
4. Accept or reject the recommendation
5. If accepted, update project documents and re-plan
6. Notify all stakeholders of the change.

In practice, the Change Request process is a bit more complex. The following describes the change control process in detail:

1. **Any stakeholder can request or identify a change. He/she uses a *Change Request Form* to document the nature of the change request.**
2. **The completed form is sent to a designated member of the Project Team who enters the CR into the *Project Change Request Log*.**

   Link To Project Change Request Log
3. **CRs are reviewed daily by the Project Manager or designee and assigned one four possible outcomes:**

▪ *Reject:*	• Notice is sent to the submitter • The submitter may appeal (which sends the matter to the Project Team) • The project team reviews the CR at its next meeting.
▪ *Defer to a date:*	• The project team is scheduled to consider the CR on a given date • Notice is sent to the submitter • The submitter may appeal (which sends the matter to the Project Team) • The project team reviews the CR at their meeting.
▪ *Accepted for analysis immediately (e.g., emergency):*	• An analyst is assigned; impact analysis begins • The Project Team is notified.

4. Process

- *Accepted for consideration by the project team:*
  - The project team reviews the CR at its next meeting.

4. All new pending CRs are reviewed at the Project Team meeting. Possible outcomes:

- *Reject:*
  - Notice is sent to the submitter
  - The submitter may appeal (which sends the matter to the Project Sponsor and possibly to the Executive Committee)
  - Executive Committee review is final.
- *Defer to a date:*
  - The Project Team is scheduled to consider the CR on a given date
  - Notice is sent to the submitter.
- *Accepted for analysis:*
  - An analyst is assigned; impact analysis begins
  - Notice is sent to the submitter.

5. Once the analysis is complete, the Project Team reviews the results.[1] Possible outcomes:

- *Reject:*
  - Notice is sent to the submitter
  - The submitter may appeal which sends the matter to the Project Sponsor (and possibly to the Executive Committee)
  - Executive Committee review is final.
- *Accept:*
  - The project Team accepts the analyst's recommendation
  - Notice is sent to Project Sponsor as follows:
    - Low-impact CR – Information only, no action required
    - Medium-impact CR – Sponsor review requested; no other action required
    - High-impact CR – Sponsor approval required.
- *Return for further analysis:* The Project team has questions or suggestions that are sent back to the analyst for further consideration.

6. Accepted CRs are forwarded to the Project Sponsor for review of recommendations. Possible outcomes:

- *Reject:*
  - Notice is sent to the submitter
  - The submitter may appeal to the Executive Committee
  - Executive Committee review is final.
- *Accept:*
  - Notice is sent to the submitter
  - Project Team updates relevant project documents
  - Project Team re-plans
  - The project Team acts on the new plan.
- *Return for further analysis:*
  - The Sponsor has questions or suggestions that are sent back to the analyst for further consideration
  - Notice is sent to the submitter
  - Analyst's recommendations are reviewed by Project Team (return to *Step 5*).

5. Notes on the Change Control Process

1. A Change Request is:

[1] Note: Sponsor participates in this review if the analysis was done at Sponsor's request.

5. Notes on the Change Control Process

- Included in the project only when both Sponsor and Project Team agree on a recommended action.

2. The CR may be:

- *Low-impact – Has no material effect on cost or schedule. Quality is not impaired.*
- *Medium-impact – Moderate impact on cost or schedule, or no impact on cost or schedule but the quality is impaired. If the impact is negative, Sponsor review and approval is required*
- *High-impact – Significant impact on cost, schedule, or quality. If the impact is negative, Executive Committee review and approval is required*

3. For this project:

- *Low-impact* – Fewer than *7* days change in schedule; one or more major use cases materially degraded

4. All project changes will require some degree of the update to project documents:

- *Low-impact* – Changes likely require update only to requirements and specifications documents
- *Moderate- or high-impact* – depending on the type of change, the following documents (at a minimum) must be reviewed and may require update:

Type of Change:	*Documents to Review (and update as needed):*
▪ Scope	▪ Scope Statement and WBS ▪ Budget ▪ Project Schedule ▪ Resource Plan ▪ Risk Response Plan ▪ Requirements ▪ Specifications
▪ Schedule	▪ Project Schedule ▪ Budget ▪ Resource Plan ▪ Risk Response Plan
▪ Budget	▪ Budget ▪ Project Schedule ▪ Resource Plan ▪ Risk Response Plan
▪ Quality	▪ Budget ▪ Project Schedule ▪ Resource Plan ▪ Risk Response Plan ▪ Quality Plan ▪ Requirements ▪ Specifications

5. Project documents:

Whenever changes are made to project documents, the version history is updated in the document and prior versions are maintained in an archive. Edit access to project documents is limited to the Project Manager and designated individuals on the Project Team.

- For this project, all electronic documents are kept in (select one of the following and describe it in the adjacent space provided):

[ ] Version Control System:

5. Notes on the Change Control Process

[ ] Central storage available to the Project Team:

[ ] Other:

- For this project, all paper documents are kept in (select one of the following and describe it in the adjacent space provided):

[√] Project file maintained by the Project Manager:

[ ] Other:

- The following individuals have edit access to project documents: Wukun Chen

Role	*Documents*
▪ Project Manager	▪ All current documents ▪ Project archive
▪	▪
▪	▪
▪	▪
▪	▪

6. Project Change Management Plan / Signatures

Project Name:	Simi Bot
Project Manager:	Wukun Chen

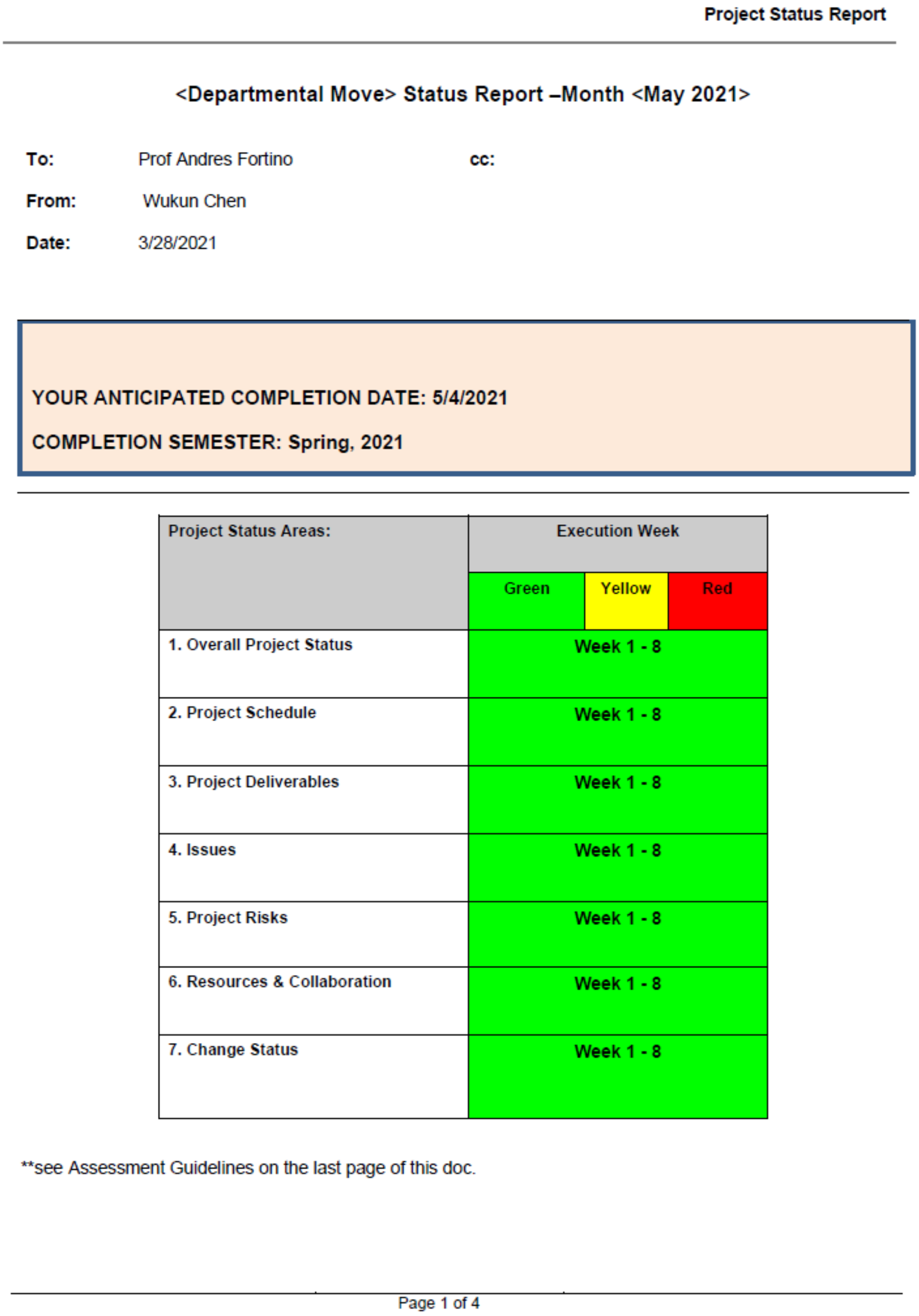
I have reviewed the information contained in this Project Change Management Plan *and agree:*

Name	Role	Signature	Date **(MM/DD/YYYY)**
Wukun Chen	Project Manager	Wukun Chen	3/28/2021

The signatures above indicate an understanding of the purpose and content of this document by those signing it. By signing this document, they agree to this as the formal Project Change Management Plan.

Appendix H

Status Reports



Project Status Report

<Departmental Move> Status Report –Month <May 2021>

To: Prof Andres Fortino **cc:**

From: Wukun Chen

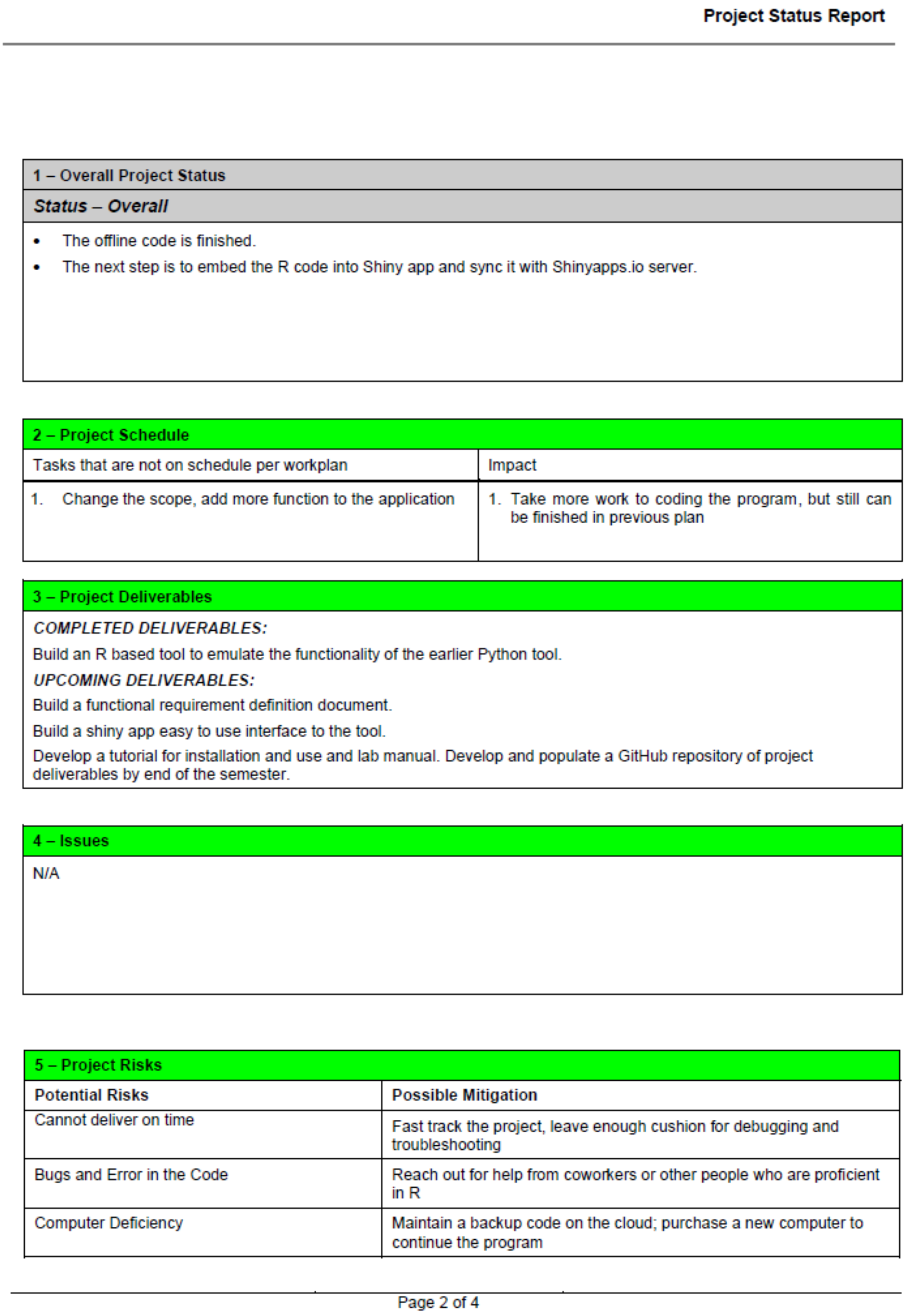
Date: 3/28/2021

YOUR ANTICIPATED COMPLETION DATE: 5/4/2021

COMPLETION SEMESTER: Spring, 2021

Project Status Areas:	Execution Week		
	🟩 Green	🟨 Yellow	🟥 Red
1. Overall Project Status	🟩 Week 1 - 8		
2. Project Schedule	🟩 Week 1 - 8		
3. Project Deliverables	🟩 Week 1 - 8		
4. Issues	🟩 Week 1 - 8		
5. Project Risks	🟩 Week 1 - 8		
6. Resources & Collaboration	🟩 Week 1 - 8		
7. Change Status	🟩 Week 1 - 8		

**see Assessment Guidelines on the last page of this doc.



1 – Overall Project Status
Status – Overall
• The offline code is finished. • The next step is to embed the R code into Shiny app and sync it with Shinyapps.io server.

2 – Project Schedule	
Tasks that are not on schedule per workplan	Impact
1. Change the scope, add more function to the application	1. Take more work to coding the program, but still can be finished in previous plan

3 – Project Deliverables

COMPLETED DELIVERABLES:

Build an R based tool to emulate the functionality of the earlier Python tool.

UPCOMING DELIVERABLES:

Build a functional requirement definition document.

Build a shiny app easy to use interface to the tool.

Develop a tutorial for installation and use and lab manual. Develop and populate a GitHub repository of project deliverables by end of the semester.

4 – Issues

N/A

5 – Project Risks	
Potential Risks	**Possible Mitigation**
Cannot deliver on time	Fast track the project, leave enough cushion for debugging and troubleshooting
Bugs and Error in the Code	Reach out for help from coworkers or other people who are proficient in R
Computer Deficiency	Maintain a backup code on the cloud; purchase a new computer to continue the program

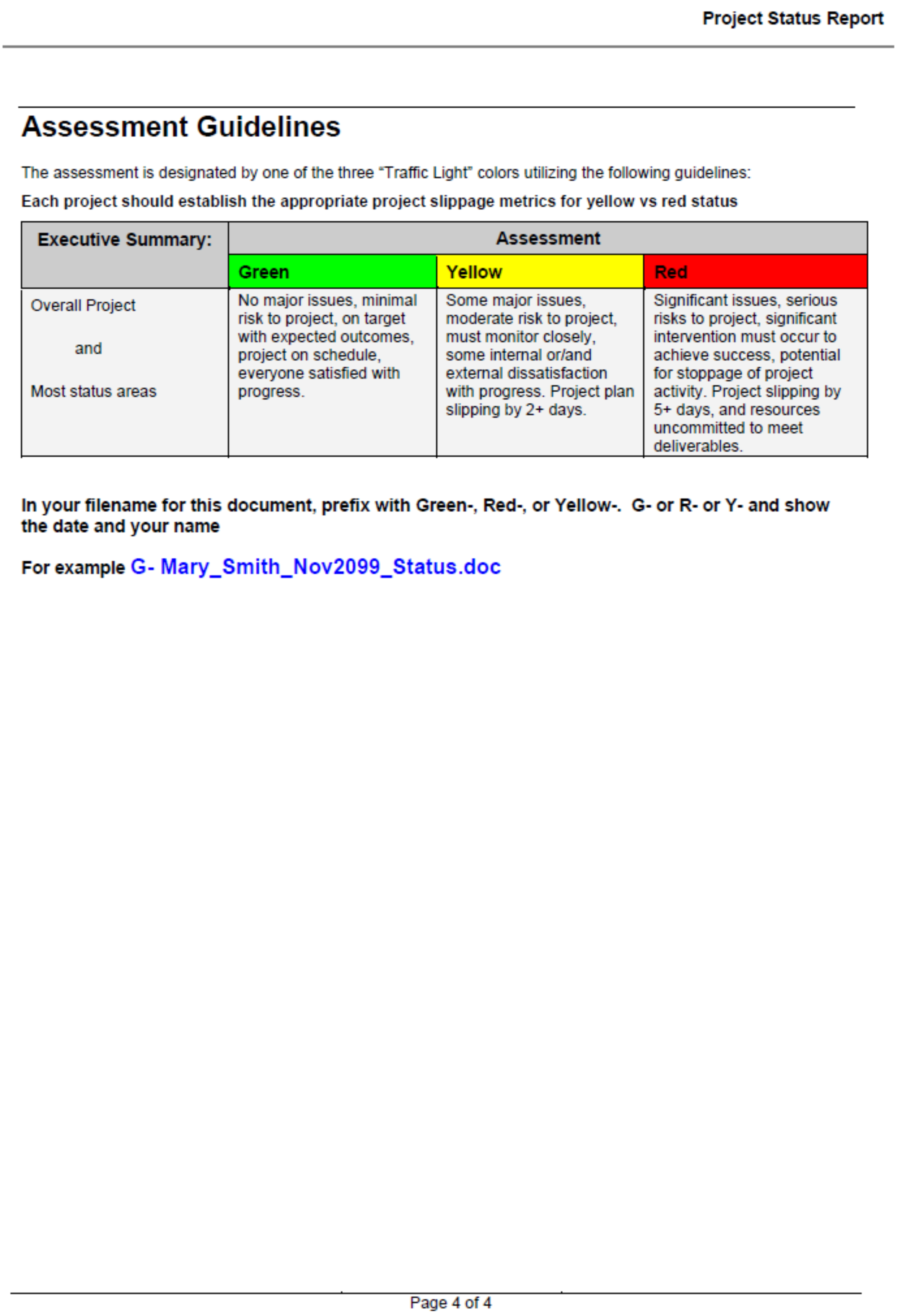
Client requests to change the project	Negotiate about new business needs and opportunities to continue the project
Health issue	Normal health issue would not influence the progress of this project

6– Resources and Collaboration
• Resources are abundant currently.

7 – Change Status	
Scope Changes	**Status** (Requested | Approved | Completed)
Add target clustering function	Completed
Add text box into user interface	Approved

Comments/Actions

8 – Sponsor Signoff	
Sponsor indicates agreement with the above status report.	
	3/29/21



Assessment Guidelines

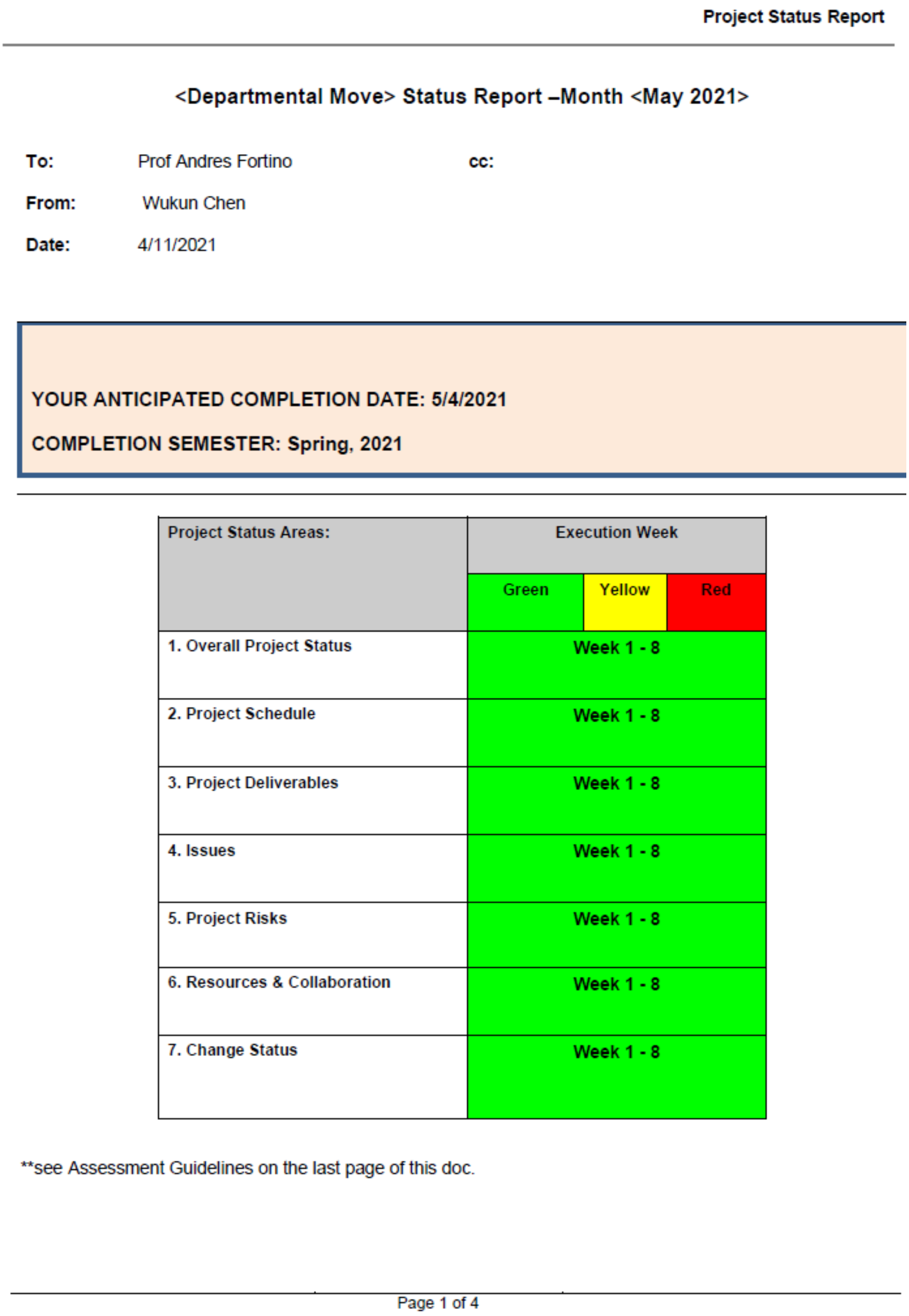
The assessment is designated by one of the three "Traffic Light" colors utilizing the following guidelines:

Each project should establish the appropriate project slippage metrics for yellow vs red status

Executive Summary:	**Assessment**		
	Green	**Yellow**	**Red**
Overall Project and Most status areas	No major issues, minimal risk to project, on target with expected outcomes, project on schedule, everyone satisfied with progress.	Some major issues, moderate risk to project, must monitor closely, some internal or/and external dissatisfaction with progress. Project plan slipping by 2+ days.	Significant issues, serious risks to project, significant intervention must occur to achieve success, potential for stoppage of project activity. Project slipping by 5+ days, and resources uncommitted to meet deliverables.

In your filename for this document, prefix with Green-, Red-, or Yellow-. G- or R- or Y- and show the date and your name

For example **G- Mary_Smith_Nov2099_Status.doc**



Project Status Report

<Departmental Move> Status Report –Month <May 2021>

To: Prof Andres Fortino **cc:**

From: Wukun Chen

Date: 4/11/2021

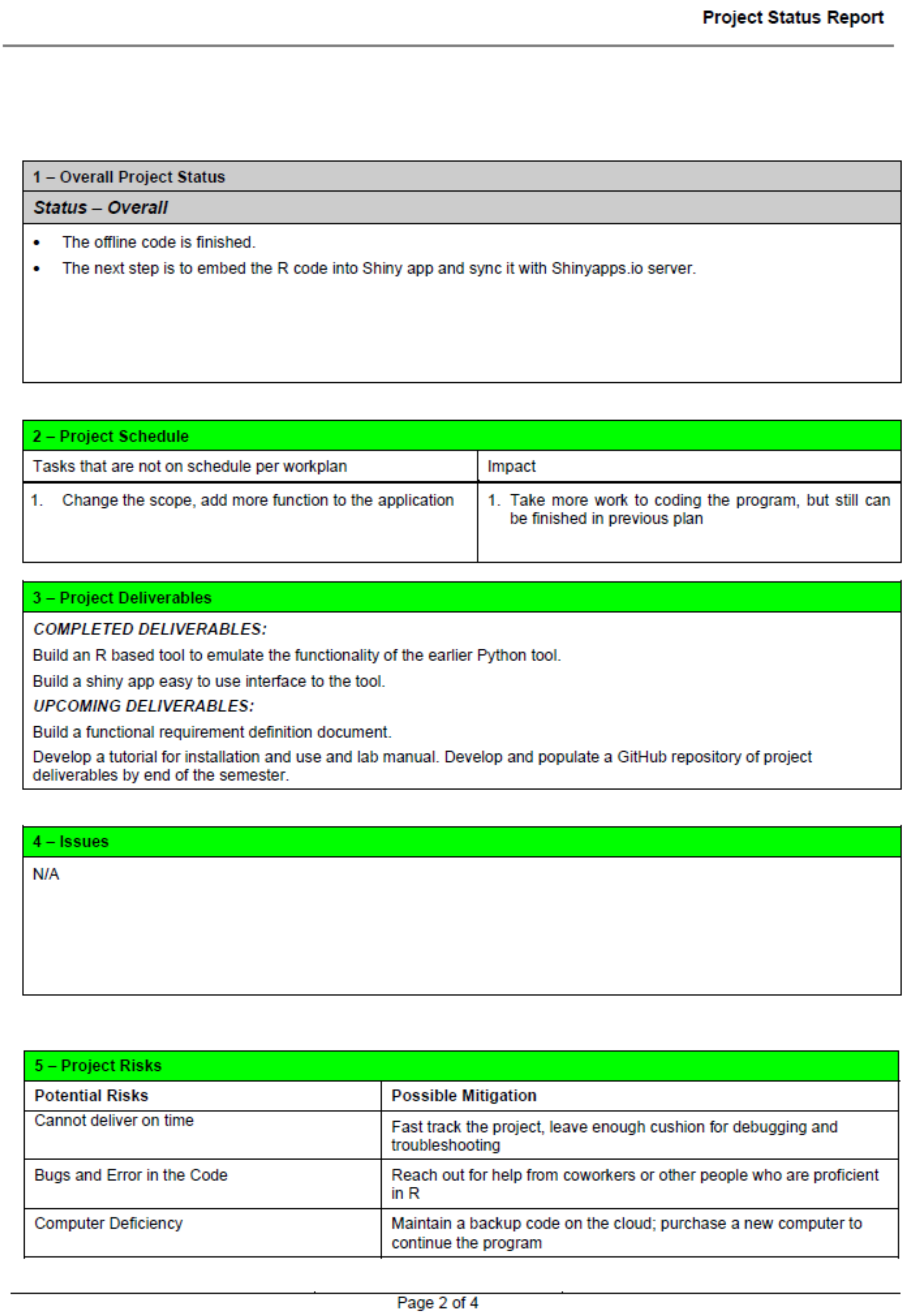
YOUR ANTICIPATED COMPLETION DATE: 5/4/2021

COMPLETION SEMESTER: Spring, 2021

Project Status Areas:	Execution Week		
	Green	Yellow	Red
1. Overall Project Status	Week 1 - 8		
2. Project Schedule	Week 1 - 8		
3. Project Deliverables	Week 1 - 8		
4. Issues	Week 1 - 8		
5. Project Risks	Week 1 - 8		
6. Resources & Collaboration	Week 1 - 8		
7. Change Status	Week 1 - 8		

**see Assessment Guidelines on the last page of this doc.

Page 1 of 4



1 – Overall Project Status

Status – Overall

- The offline code is finished.
- The next step is to embed the R code into Shiny app and sync it with Shinyapps.io server.

2 – Project Schedule	
Tasks that are not on schedule per workplan	Impact
1. Change the scope, add more function to the application	1. Take more work to coding the program, but still can be finished in previous plan

3 – Project Deliverables

COMPLETED DELIVERABLES:

Build an R based tool to emulate the functionality of the earlier Python tool.

Build a shiny app easy to use interface to the tool.

UPCOMING DELIVERABLES:

Build a functional requirement definition document.

Develop a tutorial for installation and use and lab manual. Develop and populate a GitHub repository of project deliverables by end of the semester.

4 – Issues

N/A

5 – Project Risks	
Potential Risks	**Possible Mitigation**
Cannot deliver on time	Fast track the project, leave enough cushion for debugging and troubleshooting
Bugs and Error in the Code	Reach out for help from coworkers or other people who are proficient in R
Computer Deficiency	Maintain a backup code on the cloud; purchase a new computer to continue the program

Client requests to change the project	Negotiate about new business needs and opportunities to continue the project
Health issue	Normal health issue would not influence the progress of this project

6– Resources and Collaboration
• Resources are abundant currently.

7 – Change Status	
Scope Changes	**Status** (Requested | Approved | Completed)
Add target clustering function	Completed
Add an n-gram length scroll bar	Approved

Comments/Actions

8 – Sponsor Signoff	
The sponsor indicates agreement with the above status report.	
Andres Fortino	4/11/21

Assessment Guidelines

The assessment is designated by one of the three "Traffic Light" colors utilizing the following guidelines:

Each project should establish the appropriate project slippage metrics for yellow vs red status

Executive Summary:	**Assessment**		
	Green	**Yellow**	**Red**
Overall Project and Most status areas	No major issues, minimal risk to project, on target with expected outcomes, project on schedule, everyone satisfied with progress.	Some major issues, moderate risk to project, must monitor closely, some internal or/and external dissatisfaction with progress. Project plan slipping by 2+ days.	Significant issues, serious risks to project, significant intervention must occur to achieve success, potential for stoppage of project activity. Project slipping by 5+ days, and resources uncommitted to meet deliverables.

In your filename for this document, prefix with Green-, Red-, or Yellow-. G- or R- or Y- and show the date and your name

For example G- Mary_Smith_Nov2099_Status.doc

Appendix I

Annotated Bibliography

References

1. Beel, J., & Gipp, B. (2009). Google Scholar's ranking algorithm: an introductory overview. In Proceedings of the 12th International Conference on Scientometrics and Informetrics (ISSI'09) (Vol. 1, pp. 230-241).

> *Google Scholar is one of the major academic search engines but its ranking algorithm for academic articles is unknown. We performed the first steps to reverse-engineering Google Scholar's ranking algorithm and present the results in this research-in-progress paper. The results are: Citation counts the highest weighted factor in Google Scholar's ranking algorithm. Therefore, highly cited articles are found significantly more often in higher positions than articles that have been cited less often. As a consequence, Google Scholar seems to be more suitable for finding standard literature than gems or articles by authors advancing a new or different view from the mainstream. However, interesting exceptions for some search queries occurred. Moreover, the occurrence of a search term in an article's title seems to have a strong impact on the article's ranking. The impact of search term frequencies in an article's full text is weak. That means it makes no difference in an article's ranking if the article contains the query terms only once or multiple times. It was further researched whether the name of an author or journal has an impact on the ranking and whether differences exist between the ranking algorithms of different search modes that Google Scholar offers. The answer in both of these cases was "yes". The results of our research may help authors to optimize their articles for Google Scholar and enable researchers to estimate the usefulness of Google Scholar concerning their search intention and hence the need to use further academic search engines or databases.*

This literature examines the relationship between an article's citation count and its ranking in Google Scholar. It provides statistical support on how factors influence the ranking, such as the impact of age, search term occurrence in full text, search term frequency in full text. The result shows citation count doses have a significant impact on the ranking in Google Scholar. Also, Google Scholar ranked the articles heavily on words in the title. Another surprising finding is, Google Scholar does not weigh the frequency in terms that occur in the full text. It is different from what I am going to do in my project.

2. Dai, Z., Li, Q., Yang, G., Wang, Y., Liu, Y., Zheng, Z., Yu, B. (2019). Using literature-based discovery to identify candidate genes for the interaction between myocardial infarction and depression. BMC Medical Genetics, 20(1). https://doi.org/10.1186/s12881-019-0841-8

Background: A multidirectional relationship has been demonstrated between myocardial infarction (MI) and depression. However, the causal genetic factors and molecular mechanisms underlying this interaction remain unclear. The main purpose of this study was to identify potential candidate genes for the interaction between the two diseases. Methods: Using a bioinformatics approach and existing gene expression data in the biomedical discovery support system (BITOLA), we defined the starting concept X as "Myocardial Infarction" and end concept Z as "Major Depressive Disorder" or "Depressive disorder". All intermediate concepts relevant to the "Gene or Gene Product" for MI and depression were searched. Gene expression data and tissue-specific expression of potential candidate genes were evaluated using the Human eFP (electronic Fluorescent Pictograph) Browser, and intermediate concepts were filtered by manual inspection. Results: Our analysis identified 128 genes common to both the "MI" and "depression" text mining concepts. Twenty-three of the 128 genes were selected as intermediates for this study, 9 of which passed the manual filtering step. Among the 9 genes, LCAT, CD4, SERPINA1, IL6, and PPBP failed to pass the follow-up filter in the Human eFP Browser, due to their low levels in the heart tissue. Finally, four genes (GNB3, CNR1, MTHFR, and NCAM1) remained. Conclusions: GNB3, CNR1, MTHFR, and NCAM1 are putative new candidate genes that may influence the interactions between MI and depression and may represent potential targets for therapeutic intervention.

In this article, the author used literature mining methods to discover that the GNB3, CNR1, MTHFR, and NCAM1 genes were identified and directly or indirectly implicated in the regulation of MI and depression. They marked different keywords associated with clusters and conducted the intersect correlation analysis.

3. Fortino, A., Zhong, Q., Yeh, L., Fang, S. (2020). Selection and Assignment of STEM Adjunct Faculty Using Text Data Mining, IEEE ISEC'20 Conference, Princeton University, NJ, August 1, 2020.

This paper presents the development and testing of a text data mining tool to assist in the selection and assignment of adjunct faculty to teach STEM courses. The tool scores the resume of a faculty member against course descriptions in a STEM graduate program. The tool returned a similarity score between a resume and course descriptions, which was then used as an indicator of faculty suitability to teach courses in the program. We enhanced the original tool with an improved user interface and deployed it to search for new faculty searches and in the process of assigning courses. A TD-IDF text analytic technique was used for similarity scoring. Our research question was to investigate whether a similarity-scoring tool for faculty resumes against course descriptions would be useful in the search and assignment process to hire faculty to teach specific courses. As part of our methods, we developed a friendly user interface to the existing tool using a student-centered coding contest. We applied the tool to the hiring and assignment of adjunct faculty. We measured success as the processing of a large number of open positions in a relatively short period and found a significantly high number of good fits between faculty and their course assignments. We investigated whether the scoring system positively correlated with the courses assigned to them. We successfully filled

> *over 50 unassigned courses with appropriate faculty over three months, where 30% were new hires. In the process, we discovered that the vast majority of the incumbent's similarity scores positively correlated to the courses assigned to them. This generated sufficient confidence that the description scoring system has been integrated as part of our faculty hiring and assignment processes in our programs.*

This article uses similarity scoring tools like the one I am going to make. It is more specific for faculty-course matching. The result is this tool can significantly improve assignment effectiveness, hiring effectiveness, and ascertain if the assignments were in the right discipline for each faculty member. The process of hypothesis and validation could guide my project.

4. Fortino, A., Zhong, Q., Huang, W., Lowrance, R. (2019). Application of Text Data Mining to STEM Curriculum Selection and Development, Awarded Best Paper Award, IEEE ISEC'19 Conference, Princeton University, NJ, March 2019.

> *We applied text data mining techniques from machine learning to position (job) descriptions posted on NYU's job search site, Bureau of Labor Statistics (BLS) standard U.S. job descriptions, course descriptions, and curricula descriptions. Our work compared Term Frequency-Inverse Document Frequency (TD-IDF) to Latent Semantic Indexing (LSI) and found that TD-IDF was preferred in this application. We used TD-IDF to measure the extent of coherence among the collections of our documents. We then leveraged those measurements to developed novel approaches to assist students and curricula designers in answering these questions: (1) for students, given an interest in specific jobs, which degrees and courses are most relevant; (2) for students, given courses that have been taken, which jobs are most likely to result in initial interviews; (3) for curricula designers, how aligned are degree programs with specific groups of jobs (for example, with STEM jobs); (4) for curricula designers, to what extent do current and proposed degrees address different job opportunities. Other similar applications are possible by composing our Python and JMP code. Our work could be extended by providing open-source implementation of the algorithms.*

This paper demonstrates how a Python-based tool scores different text contents with the TF-IDF method. It is a good reference for my R-based tool. Also, it offers a vision of comparing LSI vs. non-LSI algorithms. The authors select non-LSI algorithms according to the higher match rate in the samples.

5. Fortino, A., Zhong, Q., Yeh, L., Fang, S. (2020). Using Text Data Mining to Enhance the Literature Search Process for Novice Researchers, IEEE ISEC'20 Conference, Princeton University, NJ, August 1, 2020.

> *A literature search can be an arduous process, especially for novice researchers. We have developed a tool that allows a researcher to rank order a list of references that are returned by a keyword-based search engine, based on similarity to known exemplars. This significantly accelerates literature searches by novices. Our research question was can we produce a text analytic tool that, when used by an inexperienced scholar, rank-*

orders a list of references against an exemplar, so that the time needed to find relevant literature is reduced, and the literature survey section of their paper will be superior. An experiment was set up where one-course section used the tool to produce the literature review section of a thesis proposal, and the other class used traditional literature research tools. We surveyed both sections to self-report the time used for the literature search. We found some time savings by some of the students using the tool. We also provided blind, randomly selected pairs of completed proposals to SME faculty who teach that same class to assess the quality of the literature sections of the samples. We found that the tool-using section of students reported significantly less time to do the literature search, and the quality of their literature review produced had a significantly higher quality.

This scholarly article mainly researched the methodology of text scoring with Python program. The target text contents are literature papers in this case. My project, which would be more general, will cater to both resumes, papers, descriptive paragraphs, etc. This paper gives me a good understanding of what validation test I need to do for my tool.

6. Inan, E. (2020). SimiT: A Text Similarity Method Using Lexicon and Dependency Representations. New Generation Computing, 38(3), 509–530. https://doi.org/10.1007/s00354-020-00099-8

Semantic textual similarity methods are becoming increasingly crucial in text mining research areas such as text retrieval and summarization. Existing methods of text similarity have often been computed by their shallow or syntactic representation rather than considering their semantic content and meanings. This paper focuses mainly on computing the similarity between sentences without a supervised learning approach, only considering their word-level coherence which is calculated by a hybrid method of dependency parser and lexicon embeddings. Hence, we concentrate on structural similarity between text pairs by regarding their dependency parser embeddings. Our hybrid method also pays attention to the semantic information of words implied in the sentences. In the evaluation, we compare our method with the state-of-the-art semantic similarity measures in a well-known dataset. Our method outperforms most of the studies in the literature and the overall performance achieves better results when combining the similarity scores of both embedding models.

This research paper states using a lexicon-based embedding model which is a new version of the linked open data resource ConceptNet to capture the semantics of word sequences and dependency parse tree information of sentences without using any supervised learning methods. The reason why I want to choose this literature information is that ConceptNet represents more generalized meanings behind the words, and it tends to improve the performance of natural language applications.

7. Jayasudha, J., Christina Esther, A. (2019) Mining Sequential Pattern of Data in Textual Document Using Data Mining Classification Technique. Asian Journal of Computer Science and Technology (AJCST), Volume 8 No.1 Special Issue: February 2019 pp 41-45.

https://www.trp.org.in/issues/mining-sequential-pattern-of-data-in-textual-document-using-data-mining-classification-technique.

> *Text documents were transmitted over the internet for text communication. So they occurred many problems like repeated text occurred because of same data were provided on the internet. To characterize and extracting that is a most critical task for the researchers. Many researchers were characterized and applied in many fields like real-life scenarios, such as real-time monitoring on abnormal user behaviors, etc. In this case, detecting and characterize the personalized behavior of the user was provide some drawbacks. To solve this problem, this paper analyzing the sequential data and characterize the user behavior with the help of the data mining sequential pattern matching algorithm.*

According to this article, the frequent pattern growth algorithm shows the effective result for sequential analyses of the data.

8. Kim, D., Seo, D., Cho, S., & Kang, P. (2019). Multi-co-training for document classification using various document representations: TF–IDF, LDA, and Doc2Vec. Information Sciences, 477, 15-29.

> *The purpose of document classification is to assign the most appropriate label to a specified document. The main challenges in document classification are insufficient to label information and unstructured sparse format. A semi-supervised learning (SSL) approach could be an effective solution to the former problem, whereas the consideration of multiple document representation schemes can resolve the latter problem. Co-training is a popular SSL method that attempts to exploit various perspectives in terms of feature subsets for the same example. In this paper, we propose multi-co-training (MCT) for improving the performance of document classification. To increase the variety of feature sets for classification, we transform a document using three document representation methods: term frequency-inverse document frequency (TF–IDF) based on the bag-of-words scheme, topic distribution based on latent Dirichlet allocation (LDA), and neural-network-based document embedding known as the document to vector (Doc2Vec). The experimental results demonstrate that the proposed MCT is robust to parameter changes and outperforms benchmark methods under various conditions.*

This paper used three methods, TF-IDF, LDA, and Doc2Vec to transform an unstructured document into a real-valued vector and match text content. In my project, I will use the TF-IDF method, but it is good to know the pros and cons of different methods.

9. Meo, S. A., &amp; Talha, M. (2019). Turnitin: Is it a text matching or plagiarism detection tool? Saudi Journal of Anaesthesia, 13(5), 48. https://doi.org/10.4103/sja.sja_772_18

> *Institutional integrity constitutes the basis of scientific activity. The frequent incidences of similarity, plagiarism, and retraction cases created the space for frequent use of similarity and plagiarism detecting tools. Turnitin is software that identifies the matched material by checking the electronically submitted documents against its database of*

> *academic publications, the internet, and previously submitted documents. Turnitin provides a "similarity index," which does not mean plagiarism. The prevalence of plagiarism could not reduce tremendously in the presence of many paid and un-paid plagiarism detecting tools because of an assortment of reasons such as poor research and citation skills, language problems, underdeveloped academic skills, etc., This paper may provide adequate feedback to the students, researchers, and faculty members in understanding the difference between similarity index and plagiarism.*

The article states that many plagiarism prevention tools are introduced for the ease of researchers to check the originality of their work before publishing the document(s). Many ethical committees and codes of ethics have been introduced to avoid plagiarism and deal with misconduct cases at institutional levels. The author explained several situations that would be considered plagiarism, even rephrasing and restructuring the sentence. In my case, these are not considered so the matching result of my tool might be different from plagiarism detecting tools like Turnitin.

10. Robinson, J. S., and D. Text Mining with R: A Tidy Approach. 3 Analyzing word and document frequency: tf-idf | Text Mining with R. https://www.tidytextmining.com/tfidf.html

> *A central question in text mining and natural language processing is how to quantify what a document is about. Can we do this by looking at the words that make up the document? One measure of how important a word maybe is its term frequency (tf), how frequently a word occurs in a document, as we examined in Chapter 1. There are words in a document, however, that occur many times but may not be important; in English, these are probably words like "the", "is", "of", and so forth. We might take the approach of adding words like these to a list of stop words and removing them before analysis, but some of these words might be more important in some documents than others. A list of stop words is not a very sophisticated approach to adjusting term frequency for commonly used words.*
>
> *Another approach is to look at a term's inverse document frequency (idf), which decreases the weight for commonly used words and increases the weight for words that are not used very much in a collection of documents. This can be combined with term frequency to calculate a term's tf-idf (the two quantities multiplied together), the frequency of a term adjusted for how rarely it is used.*

This website explained what is TF-IDF thoroughly. It also contains an explicit tutorial for how to conduct TF-IDF text comparison with R.