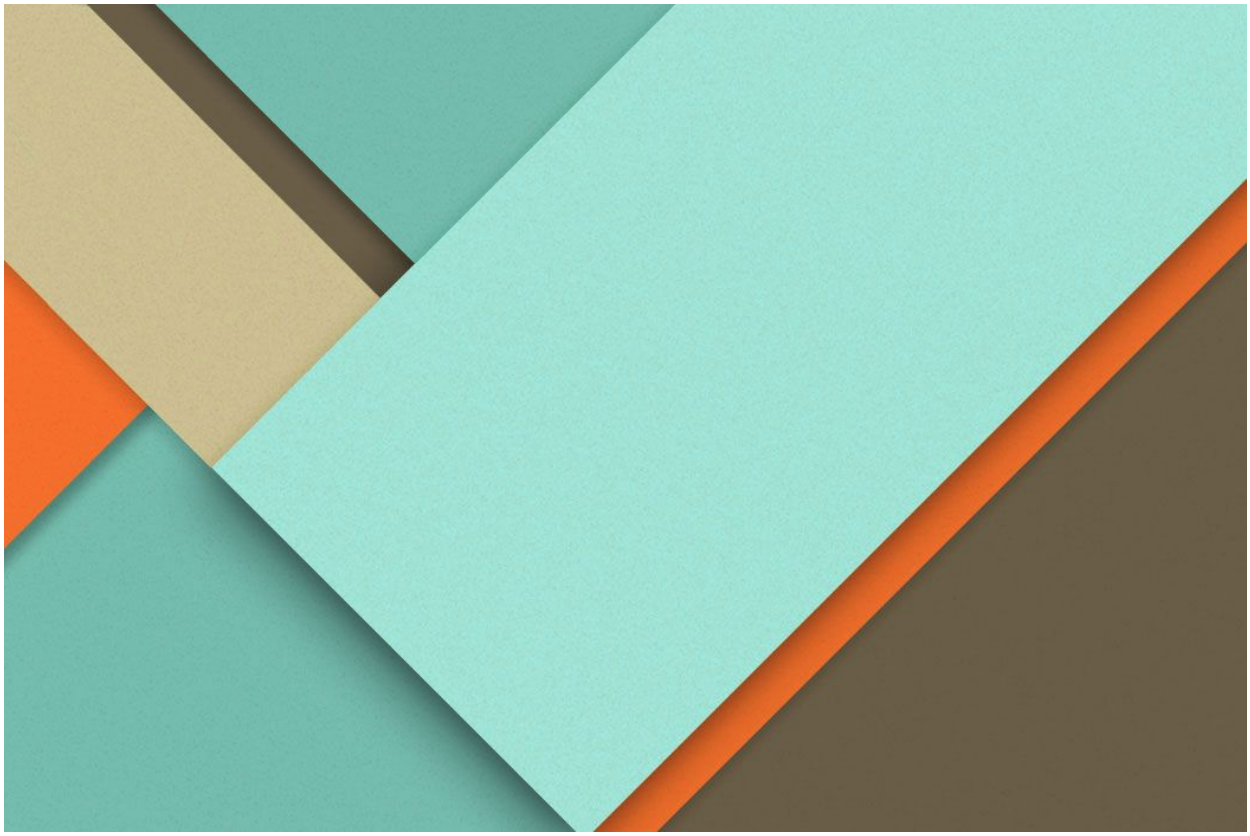

Artificial Intelligence for Requirement Analysis Tools

09.04.2019

Sddec19-08

SE 491

Table of Contents

Introductory Material

Problem Statement

Requirement tracing, or creating logical links between individual requirements, is essential in projects carried out by Collins Aerospace. When working with safety critical systems, it must be ensured that all necessary features are recognized, and no unnecessary features are included. Projects at Collins Aerospace may include thousands of requirements, each of which link to multiple other requirements. As of today, employees at Collins create and review these requirements by hand. Manually reviewing the accuracy of a requirement trace and deciding which links are good and which need to be removed is extremely expensive in terms of the time that must be dedicated to ensure that the requirement trace is sufficient for the project at hand. The purpose of this Capstone project is to develop a tool for Collins in order to automate requirement trace analysis.

The proposed solution to this problem is to utilize machine learning libraries within Python (specifically, Gensim) which include algorithms such as Word2Vec, Word Movers Distance, and Doc2Vec. These algorithms analyze the similarities between text. Our tool will take multiple Excel documents as input, which specify all individual requirements and the other requirements that each are linked to. Using the algorithms mentioned above, we will implement a solution to analyze all of these requirement links and flag them as either good, bad, or missing links based on the similarity that is computed. The tool will also recognize requirements that are not currently linked to anything, and recommend possible links. Finally, a file containing a report that details the analysis will be output to the user.

Use Case Diagram

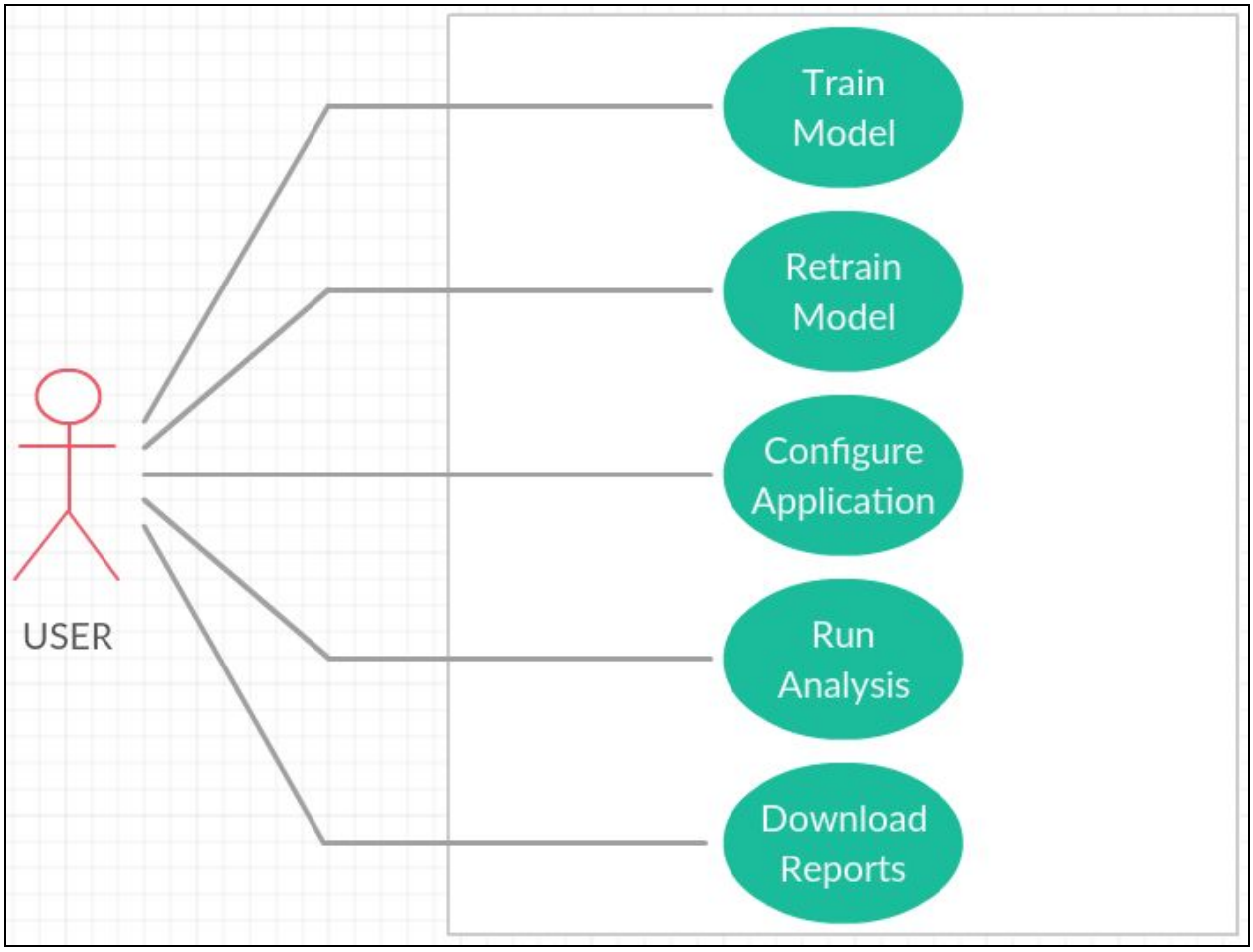

Figure 1: Use Case Diagram

The use case diagram for our system is very straight forward. For starters, any user will have access to all features of the system. Because of this, there will not be numerous actors. There are 5 major use cases that will be implemented in our system:

1. The user may train an artificially intelligent model with requirement documents. The model created here will be used to analyze links between requirements.
2. The user may retrain the model to account for words that are out of vocabulary.

3. The user may alter a configuration file which determines which documents the tool will analyze, creates links between documents, and how those documents will be read by the tool.
4. The user may use the tool to analyze links between requirements to identify whether these links are good, bad or suspicious, or even if links are missing.
5. The user may use the tool to access reports generated by the tools analysis process.

Operating Environment

This project is completely software based, and therefore will not be exposed to any sort of threatening external conditions such as weather, etc. This tool will operate on the servers within the Collins Aerospace facilities. The tool will consist of a front-end GUI, all of the back-end requirement analysis done using the Gensim machine learning library in Python. The only hazards associated with this project are security related, and we must be sure that all of the requirement information input by Collins as well as all of the reports that are generated as output have absolutely no chance of being seen by anyone outside of the organization. Our contacts at Collins were very specific that this tool will work with sensitive information, so security is essential.

Intended User(s) and Intended Use(s)

Intended Uses:

There is one base intended use for this tool. The tool will be used to generate reports on an analysis of links between requirements. The process of analyzing these links will include both classifying links as good, bad, or suspicious, and suggesting possible missing links between requirements. Once the analysis process completes, a report will be generated for the user to view and fix any errors found by the tool.

Intended Users:

The intended users of this tool will be Collins Aerospace engineers. There will be several engineering teams that will have access to the tool, however they will all have the same intended uses which allows us to group them into one type of user.

Assumptions and Limitations

Assumptions

- A configuration file will be the best way account for differing needs between teams and to give the tool parameters for how to read documents.

- HTML, CSS, Bootstrap and JavaScript will be sufficient to develop the front-end UI, and will help keep it simple.
- A clear, clean and simple UI will be best for user interaction and easy on the eyes.

Limitations

- The size of input will vary (the tool should handle any size)
- Inputted Excel documents may be formatted differently
- Must be easy to set up for new users

Expected End Product and Other Deliverables

By the end of senior design, we will deliver an artificially intelligent web based requirements analysis tool to Collins Aerospace. This tool will be capable of learning language based trends based off of user provided training documents. Once trained, the tool may be given similar documents and analyze the quality of links between requirements and suggest possible missing links. The tool will be configurable to account for several teams using differing document types or trained models. The process of analyzing links may be either manually run or scheduled to run by the user. When a link analysis finishes, the tool will generate and save its findings for the user to review and make necessary changes.

Acknowledgement

We would like to acknowledge all of our contacts at Collins Aerospace, specifically Jason Wong, as well as our faculty advisor Simanta Mitra for their time, attention, and technical guidance and expertise provided throughout the course of this project.

Proposed Approach

Block Diagram

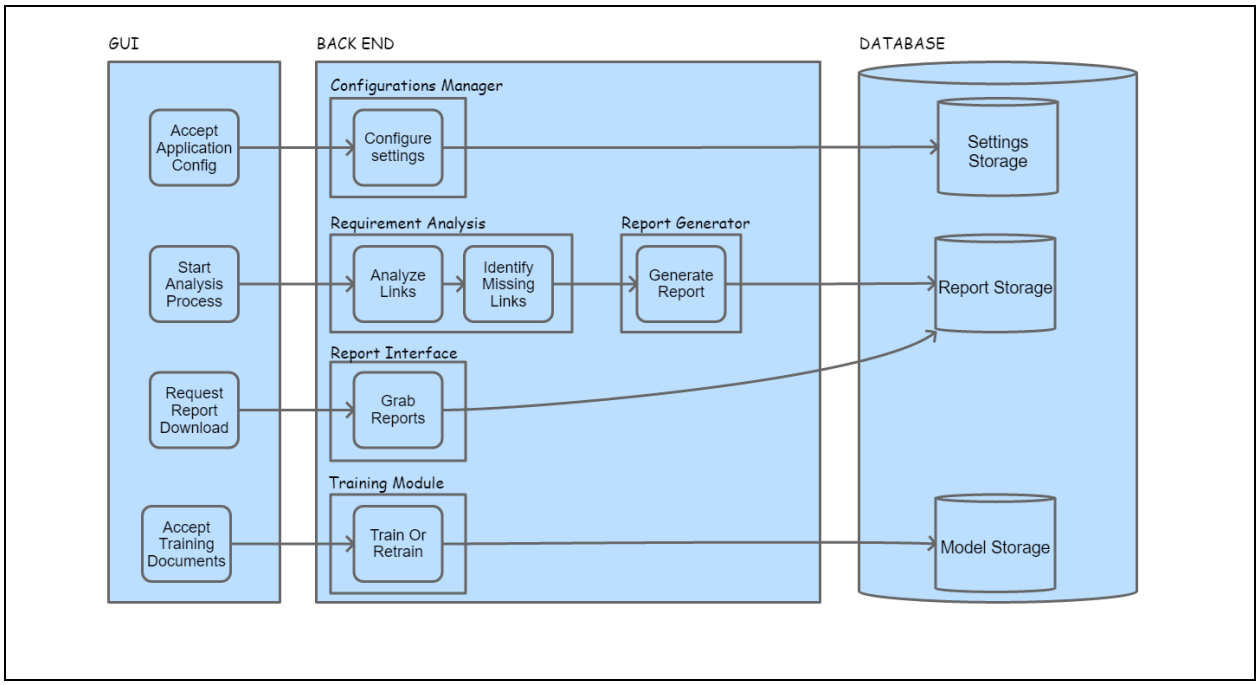

Figure 2: Block Diagram

Above is our general block diagram for our system. A close relation can be seen between our block diagram and our use case diagram. The four blocks inside our GUI capture the basis of all of our use cases.

To begin, the GUI will have an interface to accept changes to a configuration file. These changes will be sent into the Configurations Manager and be stored in the Configurations Database.

Secondly, the GUI will have an interface to begin the analysis process. The user will start the process here which will begin the Requirement Analysis. The algorithm will analyze the documents and will send its findings to the Report Generator which will create a report and save it to the Reports Database.

Third, the GUI will have an interface to Request a Report Download. The front end will send this request to the Report Interface which will search the database and give the report to the requesting user.

Finally, the GUI will have an interface to Accept Training Documents. The supplied training documents will be used in the Training Module to either train or retrain our model. Once the model is trained, the model will be saved/updated in the Model Database

Functional Requirements

- The tool must be able to classify requirement links as good, bad, or suspicious
- The tool must be able to suggest possible missing links for requirements
- The tool must be configurable to run differently from team to team
- The tool must be able to run autonomously
- The tool must be runnable via command line
- The tool must be able to learn on its own after provision of training documents
- The tool must be able to re-train itself when out-of-vocabulary words are encountered
- The tool must be easily deployable on a new machine
- The tool must allow the user to select which model they are using to analyze documents

Constraints Considerations

With this being the first student project that Collins Aerospace is hosting, the company is still putting together how students will be granted access to this secure data. Thus, until our team is granted access, the project leads at Collins Aerospace will have to test the tool whenever testing or training with real data is needed.

Technology Considerations

Effectively training a model will require feeding it a large quantity of training data. Processing this information and using it to train the model will require high processing power which not all technology will have. We will have to keep this in consideration to make sure we have efficient platforms to train our models in a reasonable amount of time.

Security Considerations

This tool will be used by Collins Aerospace, which is a government facility that advances the country's military and aerospace technology. The input to the tool will be requirements for their projects which should not be leaked in any way to the public or other companies. Keeping this data secure will be a consideration for our project.

Safety Considerations

Since this project is entirely software based, we will not be working with any components other than our computers, therefore are no safety considerations that need to be taken into account. There is also no safety considerations for the user, since the end product will be a software tool.

Previous Work/Literature Review

Collins Aerospace has some system to deal with requirement tracing so the requirements are not completely written manually. In terms of defining and designing the project, they simply defined inputs and desired end results and allowed us to define and design how we get to these end result.

Possible Risks and Risk Management

- We have a team member who will be in India for about two weeks in March. Luckily we know that this is going to happen ahead of time which allows us to plan for it. With the time zone difference between Iowa and where he will be in India, we will still have an overlap of day time. He will use this time frame to work and discuss with us what he is working on. Essentially, he will simply be working remotely for this time.
- Since our team is comprised of full time students and part time employees, there is always the risk of being overwhelmed by homeworks, exams, and work between all our classes and jobs. To account for this, we ask that everyone on our team keep their schedules organized and make time for the projects.

Project Proposed Milestones

1. Final decisions upon an AI learning algorithm
2. Final design of link analysis algorithm
3. Finish design of Database
4. Finish design of GUI
5. Design presentation
6. Finish implementation and testing of each module
7. Finish GUI implementation
8. Finish testing of full application
9. Final presentation

Project Tracking Procedures

Our team has set up a Jira server for tracking the project. We will use Jira for creating tickets for tasks in each sprint and assign tasks that need to be completed. We are also using a Git repository for version control purposes.

Statement of Work

Semester 1:

1. First we will experiment and research the natural language models such as doc2vec, Word Movers Distance, Word2Vec, GLoVe, and PV-DM etc. We will spend 6 weeks doing this task since it is the core of our project, which recognizes the similarity between two requirements and evaluate the links. This will involve significant amount of programming. At the end, we can create some statistics of the error measure of each algorithm we will have experimented and decide on which algorithm to use.
2. Once the text processing part is done, we will work on designing the algorithm to evaluate and create the links between the project level requirements and component level requirements. The existing links can be classified as good, bad, or suspicious. The algorithm is also responsible for suggesting potential links based on the similarity. When no such link can be created, the requirement will be classified as missing links. We will spend 2 weeks for this task. The expected result of this phase is a function that takes requirement files as an input and generate an output which evaluates the existing links as well as suggests potential links.
3. Now that the core part is done. We will work on designing the GUI part. The main purpose of this web GUI is it allows users to analyze their requirement files and see the reports. It should have clean and user-friendly interface. This is just for a demonstration purpose and the working GUI with REST API and database will be developed in the second semester.
4. Make a design presentation for our project.

Semester 2:

1. Start the semester by creating a connection between our front and back ends, and connecting the back end to the database.
2. Develop and test our GUI and API using html, javascript, bootstrap, and css.
3. Develop and test the configurations manager that will be able to tell our analysis algorithm how it needs to work.
4. Develop and test the training module that will be capable of two things. First, it will be able to train a new model with training documents. Secondly, it will need to be able to retrain a model when an out of vocabulary word arises.
5. Develop and test the link analysis algorithm to classify links and suggest any possible missing links.

6. Develop a report generator that takes input from our classification algorithm and puts it into a human readable format.
7. Test the full application extensively
8. Prepare for final presentation.

Estimated Resources

Personnel Effort Requirements

Task	Description	Effort needed
Research learning algorithms	Learn which artificially intelligent tools are available for comparing text	Since none of our team had worked with these algorithms prior to this project, researching them and learning how to apply them to the project requires a significant amount of time to truly understand them and be capable of implementing them efficiently.
Conduct experiments on learning algorithms	Once we have seen which artificially intelligent tools are available, we will conduct an experiment implementing the ones we found as plausible	The Gensim python library includes many of the algorithms that we will conduct experiments on. Since they are already created, they will be fairly easy to implement.
Analyze experiment results and choose one or more learning algorithms	Once we have experimented, we will compare the results of our implementations and make a decision of which algorithm(s) will work most efficiently with the input we provide it	Since the project has very strict and specific requirements, once we have seen how each algorithm works through research and experimentation, we will

		decide which algorithm(s) best suit our needs.
Design algorithm to analyze requirements	After deciding on the best learning algorithm, we must design an algorithm that analyzes the similarity computed by the learning algorithm in order to flag requirement links as good, bad, or suspicious. The algorithm must also identify requirements that do not have any links and suggest possible links based on computed text similarity.	There will be lots of attention given to this task. The algorithm designed here will define the efficiency of most of the system. With that said, we want this to be as efficient as we can possibly make it. Once an algorithm is created, it may continue to be improved all the way up until the end of the project.
Design database	We will need design a database in order to store all report information (Report Storage), store default or custom configuration settings (Configuration Storage), and save the models that have been trained or re-trained (Model Storage)	This bulk of this time will be put towards figuring out how to store our models, which we expect to be large. Once this is figured out, the rest of the design should go quickly
Design GUI	We will design an intuitive and clean looking GUI that provides a user friendly experience to our tool	This will be minimal effort since there are only a few tasks that our front end will need to perform
Connect front end, back end, and database	Establish connections between the front end, back end, and the database so that we may begin developing modules of the project	Initializing these connections will be easy and take very little time..

Begin developing modules	We will begin writing our modules for the project	This will be an intensive process. There will be several modules for us to implement and some of them may be quite large.
Test modules	Once a module is completed, we will thoroughly test it to ensure that it meets all given requirements	Testing will require a lot of our attention because we want to make sure that our requirement analysis works correctly under any circumstances. So we must be sure to write a very thorough suite of tests that analyzes all corner cases.
Insert tested modules into project structure	Once a module has been completed, we will merge it with other finalized modules in order to continue to add to the final build of the project.	This step is very straightforward, since it only requires pushing our work into the repository where we are storing finalized project modules.

Table 1: Personnel Effort Requirements

Other Resource Requirements

The tool will be web based meaning we will need to host our code on a server. We will also use jira to manage the project. Jira also requires a server to run so for these purposes we will use an apache server. We will also use the python gensim library to implement our learning algorithms. The gensim library requires other python libraries to run so we will include Miniconda which is a package manager for installing these dependencies.

Financial Requirements

The tool will be developed with python and open source software. Therefore, there will be no financial requirements to be met.

Project Timeline

Gantt Chart

(submitted separately as well)

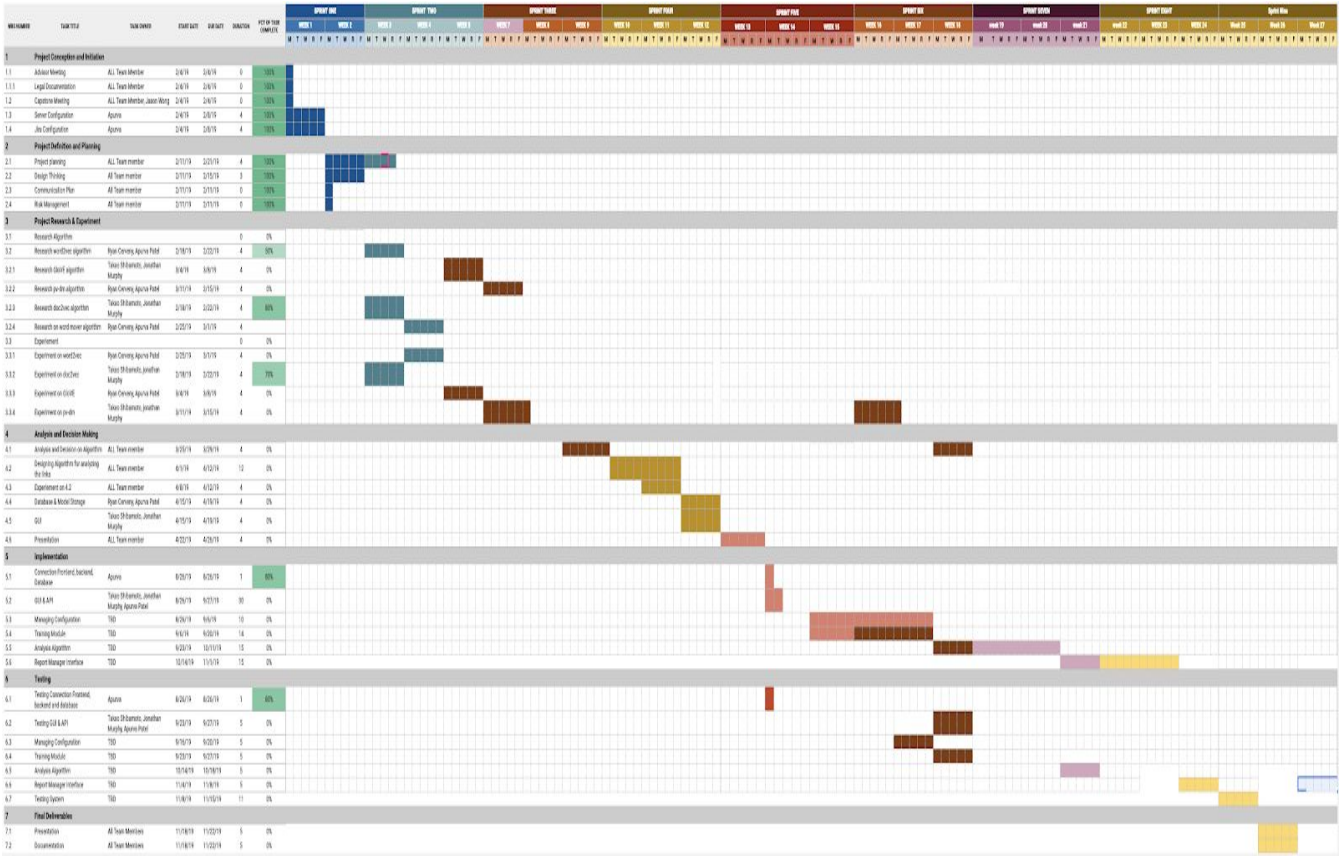

Figure 3: Gantt Chart

Closure Materials

Closing Summary

Requirement tracing is fundamental in developing reliable software. Our tool should make it easier for the engineers to trace thousands of requirements on big projects, where safety

is critical and the number of bugs must be minimized. This project will efficiently replace the current method engineers use which misses many errors and also will add the feature of suggesting links that may have been missed in the creation process. All in all, completion of this project will save engineers lots of time and will give more efficient results in a vital process that is used often by Collins Aerospace.