MISSION ECOSYSTEM MODEL GENERATION AND DISPLAY

Customer: Viz

Joseph S. Podnar

Matthew A. Turley

Ritika Maknoor

Ziyao Wang

Daniel Blackford

30 January 2017

CS 1981 - SPRINT 1 DELIVERABLE

PROJECT OVERVIEW

General Dynamics/Viz Data Scientist (our point of contact) is looking to develop a functional prototype as a demonstration of value by which to secure funding for enterprise level software design. The overall goal of the project is to create software capable of dynamically generating a mission environment model (MEM) from a processed, non-standardized document. This relies heavily on programmatically representing relationships between entities (People, Locations, Events, Resources, and Tasks).

Due to the nature of the company/project, we do not have knowledge or access to the full technology stack currently being used, or even what is under consideration to be used during the later stages of this project. Our POC, rather, has guided us to focus on an incremental approach, implementing preliminary stages and functionality, and then extending it. On one hand, this perfectly fits the concept of agile development, and we should have working software at the end of every sprint. Our POC is aware of the learning objective of this course, and is regulating his release of information to dual effect: enforce said objective, and obfuscate the inner workings of the military defense contractor. Further, this approach at the project provides us some flexibility in how we design it-- our options aren't limited by a known system we will be integrating with. However, not understanding exactly which technologies we will need to integrate with, or could possible make use of, handicaps our planning process. Also, another downside of this is that we may potentially have issues later on in which our current working code base is not relevant or extensible enough to meet the end software objectives. Therefore, our intermediate goal is to begin to implement natural language processing (NLP) of the aforementioned documents as a means to begin contextually defining these relationships.

PROJECT SPECIFICS

Communication:

We have chosen to communicate via the popular team messaging system known as Slack. We hope that its constant presence means that it is easy to keep everyone up to speed and properly segment communication when necessary. We are also using a separate slack channel as our basis for file sharing. We will all be required to check slack at least once a day to ensure it is an effective communication tool for troubleshooting and knowing when to pull down any changes.

Slack's features: seamless desktop app/mobile app/browser app integration and syncing, ability to create multiple channels of discussion/resource sharing to de-clutter the central communication space and streamline workflow, file sharing capabilities which recognize coding languages we will be using, and push notifications to ensure that the entire group is updated when content is added or changes are made.

Languages:

We have chosen to use python in developing our project. This language has many tools developed for data science that fit well with our needs for natural language processing. We have chosen to use the wisdom and knowledge of others for our NLP, as none of us are masters in the field of computational sociology. In the same vein, as we move past the initial goals of NLP toward visualizing our results, the same "data science" community already leaning heavily on python, has a variety of tools and libraries we might use in displaying the results or in defining relationships (pandas, matplot, sklearn, etc.). This will all be back-end.

Our current plan for the front-facing interface will be implemented in javascript. We chose this language in particular because have some of our teammates with UI design experience within the language. Also, it will allow our prototype to be easily accessible and demonstrable, which is one of the key goals of the project manager.

Frameworks:

One such framework we will use is the Natural Language Toolkit. This platform, hereafter referred to as NLTK, provides abstracted interfaces for working with “over 50 corpora and lexical resources.” We hope to find the right tools to effectively strip out what is important in our documents. We can use flask as our back-end framework. It is a micro framework that can meet our current requirement for the demo of this project.

Testing:

For the initial stages of the product, we will predominantly use unit tests, along with using a known and consistent input to ensure the program delivers the desired output. Since we are using python, we will use unittest framework (sometimes referred to as PyUnit) to implement the test cases. Similar to JUnit for java, unittest was based on work by Kent Beck and has become the language standard testing framework.

Subsystems:

Looking at the system as a whole, from a very general point of view, we need a backend in python to process text-files and perform the text analysis algorithm, and front-end in javascript to give an interface for that and show the data visualization.

The back-end NLP part can be divided into separate parts:

1. The system shall filter out any noise words that are considered irrelevant to the M.A.M. these are words such as articles, helping verbs, and adverbs.
2. The system shall group similar words together, such as pronouns to common human names (ex him to Herrald) when they appear within the same sentence or commonplace descriptors to their proper title (eg Hungarian Capital to Budapest, Largest American City to New York, or big British sightseeing wheel to London Eye)
3. The system shall recognize events by combining names to places (eg Eric ate a waffle in the Dutch capital)

Additional Tools:

We will need user experience design tools and graphic design tools in order to design the visualization part of the data. We might also need additional frameworks or libraries to complete the project. We do not know yet.

Possible Future Issues:

We might not be able to process the text correctly. The input is a pure text file, and we need to find those related vocabularies to generate the relationships between them, highlight different words, etc. The output might not meet the expectation that can be useful. However, even if do find the correct output, it may be difficult to display that output and visualize the data in a user friendly way so that users can understand it. Another possible future issue is that the lib resources or framework we found or wanted to use might either not work well or might potentially have poor performance.

Operating System Compatibility:

Because we are writing our program in Python it will be cross platform by nature. However, as everyone on our team uses either macOS or Windows, our project will be primarily tested on these two operating systems.

While early implementations will likely run (and be tested) on multiple OS's (python-based, csv is non-proprietary format), this software is ultimately being developed for military use. As, to the best of our knowledge, the military and DoD is largely Windows based, so we will try to skew towards Windows in the end.

Customer Interaction:

We will meet with our customer once per sprint. We will meet the customer in person to discuss the project and take notes on our progress and where changes need to be made.

LISTING OF PRIORITIZED USER STORIES

PRIORITY LEVEL: HIGH

- As a user
  I want to be able to select a file to process
  So that I can begin working on a document.
- As a user
  I want unimportant words to be ignored
  So that only useful information is stored.
- As a user
  I want similar words to be tracked as the same word
  So that the system understands they mean the same thing to me.
- As a user
  I want the system to identify human names as people
  So that I can properly track the people related to the mission.
- As a user
  I want the system to identify place names as locations
  So that I can properly track the locations related to the mission.
- As a user
  I want events to be identified as such
  So that I can properly track events related to the mission.
- As a user
  I want the things available for a mission to be identified as resources
  So that I can properly track resources related to the mission.
- As a user
  I want the goals the mission needs to be completed to be identified as tasks
  So that I can properly track tasks related to the mission.

PRIORITY LEVEL: MEDIUM

- As a user
  I want words that have different meanings when next to each other to be treated as their own unique words
  So that their meaning is clear.

PRIORITY LEVEL: LOW

- As a user
  I want to see these categorizations before a MEM is generated
  So that I can understand what choices the preprocessor made.

- As a user
  I want to be able to output the MEM to a CSV file
  So that I can use additional software for further processing of the MEM.
- As a project manager
  I want a visual representation of all changes made to the original body of text
  So that I can highlight how the software worked to non-technical stakeholders.
- As a project manager
  I want visual output that is easily digestible
  So that I can easily demonstrate value to non-technical stakeholders.
- As an engineer
  I want a chance to edit the choices NLTK makes
  So that I can help the software better understand what is similar or important.

PRIORITY LEVEL: VERY LOW

- As an engineer
  I want a chance to save the changes I've made to the choices NLTK makes
  So that I can get more benefit from processing similar documents in the future.