**Alexandria Persing**

**Progress Report Log**

Capstone Spring 2015

Elevator Pitch: I’m creating an information retrieval system; in other words, a search engine. Basically, it will allow you to pass in a document or query, then it determines its content and returns a list of other, similar, documents. This will be extremely useful for applications such as finding books or articles similar to one you’ve recently read.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sprint** | **Rating**  **(0-3)** | **Backlog Items** | **Status** | **Student Comments** | **Faculty Comments** |
| 1 | 2 | **1)** Create an api for document mapping to a given content tree  **2)** Unit tests for document mapping  **3)** Create an application to load/save/edit content trees  **4)** Change the way the file io for text extraction works, so that it uses the readAllLines iterator (so as not to use overly much memory), rather than the readAllText method | 100%  100%  100%  100% | Things I have completed already by the first meeting:  **1)** Finding test data  **2)** An api for creating content trees and adding and subtracting data from them  **3)** Unit tests for the content trees  **4)** An api for extracting individual post text from a file  **5)** Unit tests for text extraction  ------------------------------------------------------  I completed all tasks, and was able to verify that the parts of the application I have built so far all work together. | She is on track. She seems to be working a solid plan. |
| 2 | 2 | **1)** Create the api for comparing documents  **2)** Assemble a set of test documents/queries  **3)** Create a content tree based on the test documents  **4)** Actually test my system against the test documents and document results  **5)** Create an application for scanning documents which also allows you to view the created data tree in some way | 100%  100%  100%  100%  100% | Results of testing:  The program brings back 0 results for some queries, and between 1-50 results for most queries. In cases where there were results, most were not relevant according to expected results. (with an acceptance rate of .9 and a subsequent branch weight of 2.7.  Changing acceptance rate and the weight of subsequent branches did affect results. | Boyer-Moore string search algorithm  She is on track and making some progress. |
| 3 | 2 | **0)** precision / recall measurements  **1)** Word stemming  **2)** Allow duplicate words in content tree  **3)** Implement multiple words when scanning documents  **4)** Make document scanner also able to compare trees and return whether they match or not  **5)** Allow typed in queries through document scanner | 100%  100%  100%  100%  100%  100% |  | She is on track and has useful data about the performance of her system. Next she needs to spend more time looking at the actual documents and understanding specifically why the search results are so poor. Once she identifies these issues, she will address them one by one. |
| 4 | 2.1 | **1)** Study to determine whether adding more words to content tree increases precision  **2)** If it is, create a way to recommend words, else understand why it isn’t helping | 100%  100% | The problem lies in the fact that the queries have significantly fewer words than the documents; as a result, the query trees are weighted heavily, often in areas that aren’t necessarily what the query actually implies it wants as answers | She has some good results, but she still has many questions. She is beginning to formulate scientific experiments to answer these questions. |
| 5 | 2 | **1)** Experiment w/ tree ontology: flat ontology, see if it makes queries worse and document matching still  **2)** Hypothesis: not enough levels in the tree; test this hypothesis by creating a more detailed/multilayered tree  -------------------------  **3)** Implemented a tf-idf weighting system to recommend words | 100%  100%  100% |  | She learned that the ontological structure is significant. This week she needs to try to find and algorithm that compares profiles correctly. |
| 6 | 2 | **1)** Make the recommendation system able to determine words which should probably be in the same branch  **2)** Look at the algorithm again, and see how it can be modified to improve queries | 70%  100% | The recommendation system will return words it believes should also be included, but as of right now, it is extremely slow; I still need to work on making it efficient enough to be used regularly.  I have achieved ~18% precision, from 4% last week. It is not as much as I had hoped, but it is still more than 4 times better than before. | She has good metrics and some ideas on how to proceed. She is hypothesizing that the current biggest problem is the ontology is too limited. She has verified this by augmenting the ontology by hand and has seen an improvement. She will try to build a power tool to form the ontology. |
| 7 |  | **1)** Speed up recommendations by rewriting the  **2)** Make it so the trees don’t double count words/count words which are inside of others (ie finding “us” in “Russia”) without losing other words |  |  |  |
| 8 |  |  |  |  |  |
| 9 |  |  |  |  |  |