Brian Loi
ISTA355 Final Project Proposal
April 18, 2019

**Task:** The NLP task I am going to perform is to ask a model a question, and the model will replicate the functionality of a search engine using dumped Wikipedia pages. The pages will be ranked by cosine similarity in correspondence to the question asked, and from the page that has the most cosine similarity, the paragraphs in that page will be calculated for cosine similarity, and then the sentence with the most cosine similarity will be used as the answer of the question.
Example 1: Input: "Who invented the light bulb?" Output: "Thomas Edison, inventor of the light bulb…"
Example 2: Input: "What basketball team is based in Arizona?" Output: "The Phoenix Suns are…"

**Data:** The data I am going to use is one of the recent dumps from Wikimedia's dumps. The snapshot of the pages/articles is of 3:56 on April 9, 2019. The dump was retrieved from https://dumps.wikimedia.org/enwiki/latest/.

**Model/Methods:** First, I will need to parse the data from an XML format to a plain text format. In order to do this, I will be using WikiExtractor. The output from WikiExtractor are text files containing multiple content of Wikipedia pages for each file. Therefore, I plan on iterating through the pages and parsing them again to store their information into a single CSV format for easier file reading. The program will ask the user for a question or take in a random shuffled question list for assessing performance. An algorithm will convert the question into a vector, then compare that vector using cosine similarity to every single Wikipedia page's vector from the data. After that, the question's vector will be compared to all the paragraphs in the page with the highest similarity score, and from that, the highest similarity score of the sentences from the paragraph with the highest score will be extracted as the answer to the question. Based on the results, extra time, and effort involved in this, extra features may be included with cosine similarity. For now, the algorithm is primarily based on cosine similarities.

**Existing Software:** I plan on using the WikiExtractor, which has a GNU General Public License and is free to the public. I also plan on using python libraries such as NumPy, random, NLTK, and a few other general python libraries. For now, this is all existing software I plan on using, but I may use others as I get further into the project and find the need to.
WikiExtractor: http://medialab.di.unipi.it/wiki/Wikipedia_Extractor#Introduction

**Analyses:** I will evaluate my task by creating a list questions that I know have answers within the data, along with a correct answer list. The performance will be based on how many of these questions the program is able to answer correctly. The list of questions may be extensive and randomly sampled and shuffled before asking the program. The task is done well if a large majority of the questions are answered correctly.

**Breakdown:**

| Task | Difficulty | Deadline |
|---|---|---|
| Get Data (Wikimedia Dumps) | Easy | April 18 |
| Clean and Sort Data | Easy | April 20 |
| Build Model to Find Most Relevant Page | Hard | April 22 |
| Extend Model to Find Most Relevant Paragraph | Medium | April 24 |
| Extend Model to Find Most Relevant Sentences | Easy | April 25 |
| Improve model to use other models in addition to cosine similarity | Medium/Hard | April 30 |

**Milestones:**
**Milestone 1 (Monday, April 22):** Get all data, clean/sort the data, complete a cosine similarity model to find most relevant pages corresponding to the asked question
**Milestone 2 (Monday, April 29):** Extend the model to find the most relevant paragraph and sentence using cosine similarity, add other features besides cosine similarity to increase answer accuracy

**Evaluation:** The bulk of the work in my project will be on my algorithms. Most of the expectations and effort will be on the algorithms as I am focusing on cosine similarities. I am pulling my data from Wikimedia's dumps so there will be no effort in collecting the data. I am using an existing tool to parse the XML format of the Wikimedia dump into text files. I am doing a little sorting of the data, but I do not expect it to take me to long (probably longer to run than it will be to write). Overall, I expect to be evaluated primarily on my algorithms as I will be focusing on implementing cosine similarities and any other answer searching/relevancy methods that may come along.