Project Proposal: Wikipedia Speedrun Automation

Alex Lewin, Michael Kvicala

3/5/21

---

1 Executive Summary

This project falls under the **software track**. It will be a stand-alone tool due to the novelty of the use case. Perhaps in the future we will contribute this tool into a larger library of online cheat engines, but we do not believe this exists (yet).

Wikipedia has long served as the de facto source of information about any topic. Since its conception, students have turned to Wikipedia to gain valuable insights when writing papers, studying for tests, and completing assignments alike. In recent years, some students have found a way to utilize Wikipedia to help pass the time: Wikipedia Speedrunning.

---

2 Challenge Description

2.1 Wikipedia Speedrunning

Wikipedia Speedrunning challenges players to navigate between articles as quickly as possible. Participants do this by utilizing the structure of Wikipedia pages, taking advantage of hyperlinked words in an article. By clicking on a hyperlinked word, the participant is presented another Wikipedia article describing the clicked word. Using these hyperlinked words, it is possible to maneuver between virtually any two Wikipedia articles. The object of the game: navigate from `X` (the "*starting*" article) to `Y` (the "*goal*" article) as quickly as possible.

For example, consider a game with `Mario` as a starting article and `WWII` as the goal article. In this game, a player could reach the, click on `Italian` as it is a hyperlinked word to bring up Italy, and then click on `WWII` as that is a hyperlinked word in the `Italian` article and brings you to the `WWII` goal article.

We plan to create a program that will be able to autonomously complete this challenge.

Note: We acknowledge this may not be a useful application, per say, but we do believe this will be an novel, elegant application of the information retrieval principles. While there may not be any *commercial value* to this product, we believe that this has the potential to make a significant contribution to a niche community.

---

3 Product Proposal

3.1 Users

We predict to attract three types of users for our product:
1. **Speedrunners**: Within the culture of Wikipedia speedruning, many players train to improve their skills, thus improving their times. We expect these players to use our software as a training tool - similar to chess

players training with an engine.
2. **Cheaters**: Our software will undoubtedly be used by dishonest players, attempting to outpace their friends. We hope to create a product that will allow cheaters to flourish.
3. **Hobbyists**: Perhaps there may be a sector of users that are interested in our tool, purely out of interest in the game.

3.2 Implementation Strategy

To conceptualize this problem, we model Wikipedia as an unweighted graph - treating **documents** as **nodes** and **hyperlinks** as **edges**. Using this model, we will perform an A* search to find a path between the *starting* and *goal* documents. The A* algorithm will use the document's distance to the *goal* article as the search heuristic.

The program will continue searching until the *goal* document is found.

3.3 Information Retrieval Strategy

Our implementation strategy is as follows:
1. **Create normalized TF-IDF vectors for "accessible" documents.** Initially this will only include the start document, the goal document, and the documents linked on the starting document.
2. **Rank linked documents by their *distance* to the goal document.** We will use a distance measure under the vector space model to rank the documents. (Cosine Similarity, Jaccard Distance, Euclidean Distance, etc.)
3. **Select the document with the minimum distance to the goal document.** This is the A* heuristic.
4. **Update the TF-IDF vectors with the new data.** The new data comes from all of the linked documents found in the selected document.
5. **Repeat steps 2-4 until the target document is found.**

---

4 Analysis

4.1 Validation Measures

To show the validity of our solution, we will analyze two measures of success:
- **Real Time Speed**: Amount of time it takes to perform a search.
- **Length of Path**: Number of documents in the path between a starting and goal document.

We will use these measures to compare different configurations of the variables (below). In addition, we will compare our solution to the performance of humans completing the same task.

4.2 Variables

There are several variables that will influence our product's efficiency. We will test implementations for each of the following: - **Smoothing Formula**: (Bayesian Smoothing, Laplace Smoothing, Linear Interpolation)
- **Hyperparameters**: for each of the smoothing strategies that utilize hyperparameters.
- **Distance Measure**: (Cosine Similarity, Euclidean Distance, Jaccard Distance)

---

5 Technologies

Python 3, Wikipedia API: We will use Python 3 for development, utilizing a Wikipedia API pip package. This package allows us to query for documents by title and query for linked documents directly. - https://pypi.org/project/Wikipedia-API/

---

6 Project Roadmap

We will implement this software in three phases:
1. **Create MVP CLI Application**: We will create a Minimum Viable Product using a single configuration of the variables, which will run on a Command Line Interface. (Done by 3/20/21)
2. **Test and Optimize Configurations**: In this stage, we will test and analyze the success of each configuration of the above variables. (Done by 4/1/21)
3. **Deploy within product**: If time permits, we aim to package this application in a Chrome Extension, allowing for real-time completion of the challenge. (Done by 4/19/21)