Planning Report

An interactive website showing the association of words as synonyms

Project summary

The aim of this project is to re-design the English Thesaurus, using a network design model to visualise words and word associations.

This will be made available to users online either via a website or mobile application.

This project entails:

* The development of a back end database containing a reasonably large set of English words
* The development of a front-end website/app

Project context

This project is being done as a personal endeavour. As a soon to be computer science graduate, I want to make something that utilises the skills I’ve learned for the purpose of solving a problem I’ve always faced. I have always struggled with remembering word meanings. Dictionaries and Thesaurus are the two methods that have been around for a long time for looking up the definitions of words. These lexicons have been around since before computers and in my opinion had not been designed to be adapted to the current technology we have today. I will personally undertake every aspect of this project, from the planning, extraction of data, backend, to the user facing front end, I intend to make a fully functional tool for myself that hopefully others will find useful as well.

Project objectives:

The end result of this project will involve a fully functional front end and back end. The front end will be made available to the public via a website and an accompanying mobile application further down the line. The website will be hosted on Github to attract attention. This also provides the benefit of directly linking to my code which I hope will catch the attention of my future employer. The backend will be hosted on a server capable of scaling upwards in case the website gets traction. Whether the data is structured or not will determine which form of storage to use, sql or nosql. The website will be self managed, so scaling should be done automatically with minimal maintenance involved.

Project Quality and Compliance:

This project represents my ability as a computer scientist. I intend to make something I am proud of and am willing to show off to peers, friends, teachers, etc.

The primary goal is to get a **working website** up and running. Once a website is up and going, **extracting data** will become the primary focus. Once a proper means of extracting data has been found, the next goal will be setting up a **server** that can store data the projected amount of data. I will be reiterating each of these components throughout the development process, polishing each as I go.

System Users:

General public,

Those enthused about words or want to broaden their vocabulary

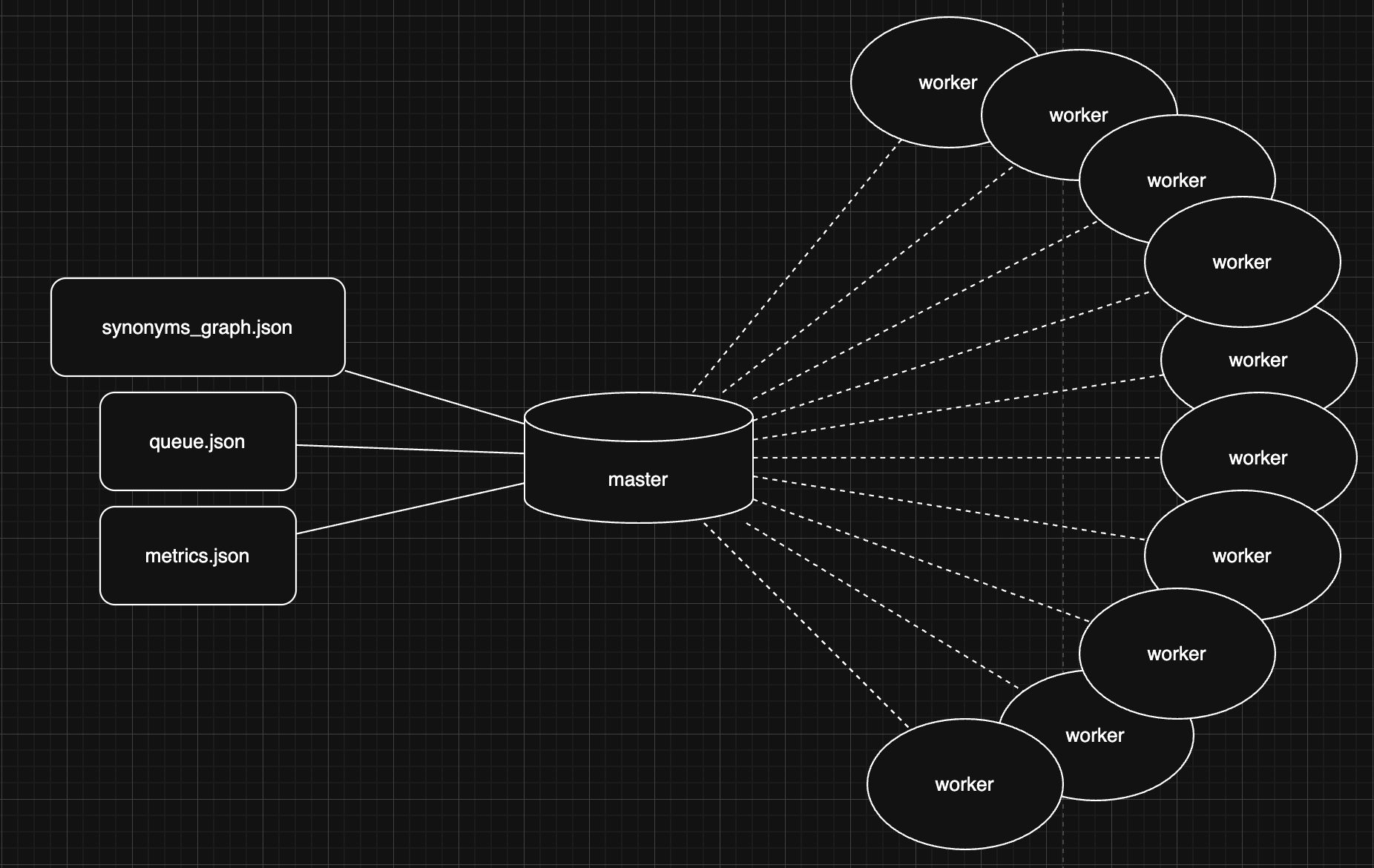
System requirements:

identify the functional and non-functional requirements of

the system being developed and describe these in the form of a Requirements Trace

Matrix

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Req# | Para# | Requirement | Type | Use Case |
| 1 |  | The website needs to be able to run on all browsers following current web standards | SWC |  |
|  |  | The graph data will be stored in a database | SWC |  |
|  |  | There will be a search bar that allows user to input a word | SWC |  |
|  |  | Website displays *specific* word at screen centre surrounded by its synonyms displayed in a graph format | SW | User\_loads\_website |
|  |  | If the user doesn’t have an account, the first word will be vocabulary | SWC |  |
|  |  | If the user has an account, it will display their most recently searched word | SWC |  |
|  |  | The degree of terms on the screen will be determined by the browser window size |  |  |
|  |  | The website will allow the user to pan around the screen showing different areas of the graph | SW | User\_pans\_screen |
|  |  | As user clicks and drags, the page will pan, showing a different section of the graph; |  |  |
|  |  | This should be done dynamically – when the user reaches the boundary of the cached word\_network ( nth degree term is just outside the visible window ), a new call will be made to the server to pull a new local network ( should be the lemma closest to the centre of the screen ) |  |  |



I would like the following worker-master model for the scraping of the Collins English dictionary

Lemma = unique word-definition pair

Most words will have multiple lemma ( think homonyms )

The key here is being able to extract each lemma, store it with a unique id, and link the correct lemma with each synonym ( inference is key )

Synonyms\_graph.json contains the an array of lemmas, each lemma has a list of synonyms that relate to other lemmas in the file ( act as foreign keys ).

The python script collins\_scrape.py handles the extraction of lemmas one word at a time.

It does so by going to the word url.

For each meaning found, the script extracts a lemma, appending it to the json file.

Due to collins anti-scraping protocols, the script needs to act slower to act more human like to avoid detection.

This is a problem as there are currently 200,000 English words ( in use ) that need to be extracted.

Currently, this process is done on only one thread, which is far too slow to be practical.

To circumvent this, I will need to implement a multithreaded processing.

This will follow a similar approach, however instead of having only a single worker processing each word in the queue, a master will be allocating words from the queue to multiple workers.

The process will go as follows:

Given

* a json file to store all lemma
* a queue
* a metrics file

The master will have access to the json file, the queue, and the metrics file.

The master will “pop” the top item in the queue.

* It will take the first item in the queue
* Send it to a worker in order to be processed
* Await for the worder to send back a json object containing a array of lemma
* Assign a unique id to each lemma
* Append the found lemma to the json file
* Remove the word from the queue

As each worker acts independently of another, this can be done many many times concurrently.

While the master is awaiting a response from the worker, the master will be sending the next word to another worker and so on and so forth, for as many workers as feasibly possible.

Our goal here is to have as many workers as possible scraping lemma from each page on the collins online english dictionary

This should net us with a complete json file contain all word-meaning paris with their associated synonyms.

Linking ( replacing the term with the lemma unique id ) will be done afterwards \*inference using synonyms to an nth degree