Content FTS

Content Full-Text Search

SolarSPELL

May 26 - August 8, 2022

TABLE OF CONTENTS

Project Summary

The Content FTS project is aimed at improving SolarSPELL’s search feature. When the user types in a query, the search string will first be evaluated for any spelling errors. It will then be lemmatized to look at the variant forms of the search query. Then, the program will perform the search through the database for the query and return a list of results in a json format.

The current search does not support corrective spelling for the search query nor does it provide results for the lemmatized version of the query. The updated search feature addresses those shortcomings by utilizing python libraries.

Functions

1. corrected_spelling (str): returns a string with corrected spellings.
   a. ex) doggs -> dogs
2. pos_tagger(nltk_tag): returns a specific part of speech tag for the input (to be used for lemmatization).
3. lemmatizer (str) : returns a list containing separate lemmatized words.
   a. ex) “dogs” -> [“dog”]
4. searchQuery(str, int): returns the specified number of resources that match the given string.
   a. ex) dog, 15 -> [(19638, 'Vusi and Sinazo', 'Vusi_and_Sinazo.pdf', 'A Dog chases the Thief.', 115199.0)]
5. convertDict(list): converts the input list of solar spell resources into a dictionary of the desired format.
   a. ex) [[a,b,c,d,e]] -> [{'id': 'a', 'title': 'b', 'file_name': 'c', 'description': 'd', 'file_size': 'e'}]
6. completeSearch(str) : compiles the aforementioned methods to return a json result processable by the solarSPELL frontend application.

Required Libraries

Flask - a web application framework that allows us to make the python program into a web service that is accessible by the front-end application

Json - data formatter based on JavaScript object syntax, used to transfer data to web applications (send files so that it can be displayed on the front end web application)

NLTK - Natural Language Processing Tool Kit wordnet was used to tag the parts of speech for words in the search query before lemmatizing them or reducing them to their root word form (a word stripped of its prefixes and suffixes, or for verbs, reduced to its infinitive form)

PySpellChecker - utilizes a Levenshtein Distance algorithm and compares all possible permutations to a known list of words and returns the most likely results

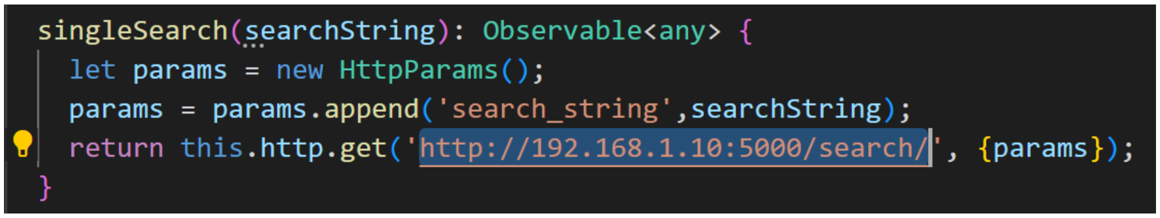
Python3 - version 3.10.5 was the python version that we used to code this file

Environment Set Up

https://github.com/atharva508/SolarSPELLsearch

The repository contains python code for running a more accurate search over the solarSPELL database. In order to run the file, follow the set up instructions below:

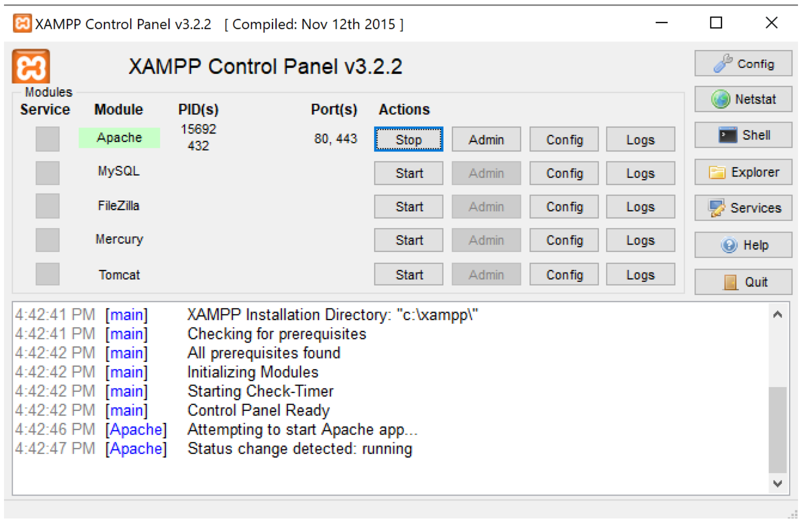
1. Open your command prompt
2. Set up a virtual environment (resources on page 10) in the directory with the backend files
3. Install all the packages in the requirements.txt file using the command:
    a. "pip install -r requirements.txt"
4. Run the nltkReq.py file to download all the required nltk packages in the virtual environment using the command:
    a. “python nltkReq.py”
5. Activate the virtual environment (resources on page 10) and run the content_fts3.py file using the command:
    a. “python content_fts3.py” for windows
    b. “python3 content_fts3.py” for mac
6. Copy the url visible in the terminal and paste it within the data.service.ts file from the services folder in the solarSPELL application; example below:



```
singleSearch(searchString): Observable<any> {
  let params = new HttpParams();
  params = params.append('search_string',searchString);
  return this.http.get('http://192.168.1.10:5000/search/', {params});
}
```

7. Set up the solarSPELL application on your system following the linked document

8. Open up XAMPP, start up the Apache server:


XAMPP Control Panel v3.2.2 [Compiled: Nov 12th 2015]
XAMPP Control Panel v3.2.2
Config
Netstat
Shell
Explorer
Services
Help
Quit
Modules

Service	Module	PID(s)	Port(s)	Actions			
	Apache	15692 432	80, 443	Stop	Admin	Config	Logs
	MySQL			Start	Admin	Config	Logs
	FileZilla			Start	Admin	Config	Logs
	Mercury			Start	Admin	Config	Logs
	Tomcat			Start	Admin	Config	Logs

```
4:42:41 PM  [main]    XAMPP Installation Directory: "c:\xampp\"
4:42:41 PM  [main]    Checking for prerequisites
4:42:42 PM  [main]    All prerequisites found
4:42:42 PM  [main]    Initializing Modules
4:42:42 PM  [main]    Starting Check-Timer
4:42:42 PM  [main]    Control Panel Ready
4:42:46 PM  [Apache]  Attempting to start Apache app...
4:42:47 PM  [Apache]  Status change detected: running
```

9. Open a new command prompt and navigate into the solarspell-app folder, then run "ng serve"

10. Open your browser and go to http://localhost:4200, you should be able to see the SolarSPELL application there

Testing

1. Testing the content_fts3.py file -
    a. Run the file in the virtual environment
    b. Copy the port from the terminal and paste it into your browser
    c. Add in “?search_string=dogs” after “/search/”
    d. Json results for “dogs” should be pulled up, you can change the search string to test that the file returns results for other strings as well

Troubleshooting

1. Troubleshooting running the content_fts3.py file -
    a. Ensure that you have all the required packages downloaded
    b. Update pip install or setup tools
    c. Ensure that the correct database is included in the folder with the content_fts3.py file
    d. Ensure that you are using python3
    e. Ensure that the virtual environment is activated
    f. Ensure that you have navigated to the correct directory before running the commands
        i. Example path for running the application:
           C:\Users\Name\solarspell-app>ng serve
        ii. Example path for running the python file:
           (venv) C:\xampp\htdocs\backend>python content_fts3.py

Resources

For help with concepts or troubleshooting with set up, refer to the following resources:

1. Virtual environment (venv) -
   a. Commands to activate the virtual environment for all system types: https://docs.python.org/3/library/venv.html#:~:text=A%20virtual%20environment%20is%20a.part%20of%20your%20operating%20system.
   b. Virtual Environment Set Up (Windows)
   c. Virtual Environment Set Up (Mac/Linux)
2. Learning about the required libraries and NLP -
   a. Flask
   b. Intro to TensorFlow for deep learning gets you started using TensorFlow, and covers image classification, text classification, natural language processing, and time-series predictions
   c. Json
   d. NLTK Wordnet
   e. PySpellChecker
   f. Python 3.10.5

Next Steps

1. Increasing the performance and efficiency of the search feature enabling it to run smoothly on multiple parallel devices.
2. Adding keywords -

   Word embeddings to expand the list of keywords for each resource in the database to provide synonym support.
3. Rank Relevancy -

   Our idea was to gather search data (search strings and the resources that are accessed) from currently deployed devices and analyze which resources are the most relevant (clicked on) for a given search string. The gathered data would then be used to create a reinforcement learning model where the program would make connections and gain an understanding about the relevancy of different files.
4. Implementation of the updated solarSPELL search feature on the raspberry pis to be deployed.