**Identification of User Query on Stack Overflow Using Semantic Search**

Stack Overflow is a question and answer site for professional and enthusiast programmers. It's built and run by the community of developers, as part of the Stack Exchange network of Q&A sites. A lot of content is present in the form of stack overflow questions and answers, various studies point that developers face problems while development life cycles and they ask questions on stack overflow which gets answered by fellow developers across the globe.

For a new developer to understand a concept or solve an issue, it is very difficult to identify the problems. It involves domain experts in form of experienced software developers. The information present is overwhelming and at times can be too much to handle for a budding developer.

This project deals with the above issue where when developers enter their queries in our search engine, our search engine understands the essence of the query and gives a more related answer from Stack Overflow.

**Getting Started**

First things first, download the files from GitHub repo. You should have an active internet connection.

**Prerequisites**

* Python 3.7 or higher
* Tensorflow
* Node 8.12(optional)
* Jupyter Notebook(optional)

**Installing**

Python Libraries or packages required: (below commands to be run on terminal)

1. nltk

pip install nltk

2.tensorflow v1.13.1

pip install tensorflow

3.tensorflow\_hub

pip install tensorflow\_hub

4.requests

pip install requests

5.flask

pip install flask

6.flask\_cors

pip install flask\_cors

**About Libraries/Packages**

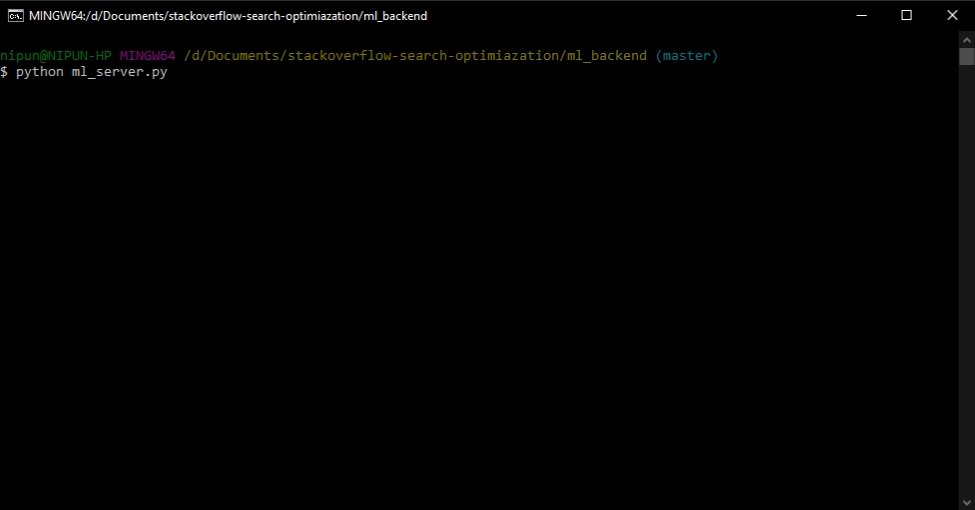
1. **Natural Language Toolkit(nltk)**: The Natural Language Toolkit, or more commonly NLTK, is a suite of [libraries](https://en.wikipedia.org/wiki/Library_(computer_science)) and programs for symbolic and statistical [natural language processing](https://en.wikipedia.org/wiki/Natural_language_processing) (NLP) for English written in the [Python programming language](https://en.wikipedia.org/wiki/Python_(programming_language)). We use NLTK to break down out string to tags containing nouns, adjectives etc
2. **TensorFlow & TensorFlow Hub:** TensorFlow is a [free](https://en.wikipedia.org/wiki/Free_software) and [open-source](https://en.wikipedia.org/wiki/Open-source_software) [software library](https://en.wikipedia.org/wiki/Library_(computing)) for [dataflow](https://en.wikipedia.org/wiki/Dataflow_programming) and [differentiable](https://en.wikipedia.org/wiki/Differentiable_programming) programming across a range of tasks. TensorFlow Hub is a library for the publication, discovery, and consumption of reusable parts of machine learning models. We use TensorFlow’s universal sentence encoder to perform similarity functions.
3. **Requests:** The [requests](http://docs.python-requests.org/en/master/) library is the de facto standard for making HTTP requests in Python. It abstracts the complexities of making requests behind a beautiful, simple API so that you can focus on interacting with services and consuming data in your application.
4. **Flask and Flask cors:** Flask is a micro [web framework](https://en.wikipedia.org/wiki/Web_framework) written in [Python](https://en.wikipedia.org/wiki/Python_(programming_language)). It is classified as a [microframework](https://en.wikipedia.org/wiki/Microframework) because it does not require particular tools or libraries.  A Flask extension for handling Cross Origin Resource Sharing (CORS). We are using flask to bridge between the JavaScript frontend and Python backend

**Running the tests**

After installing the above packages, open terminal of your respective OS and cd into the downloaded GitHub folder.

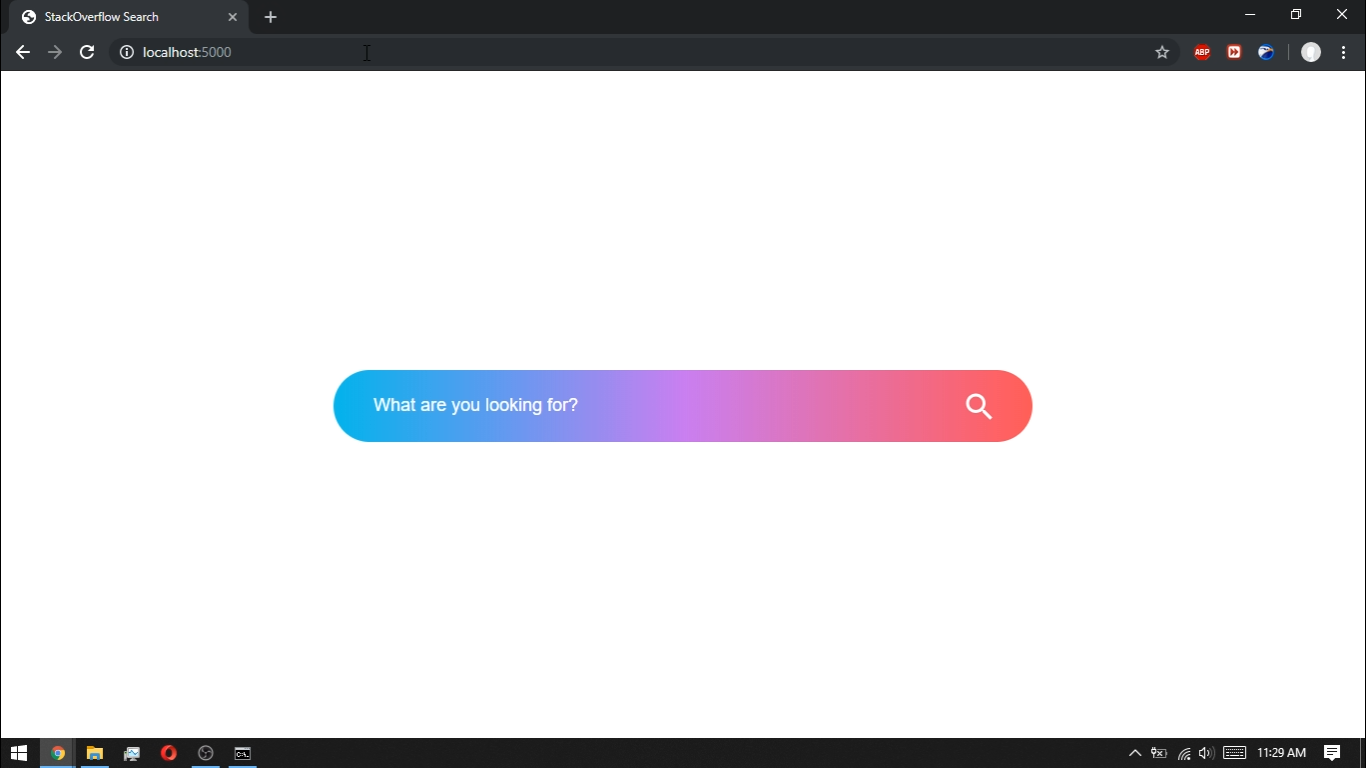
Now type the following to run the code:

python ml\_backend/ml\_server.py



This should successfully run the search engine on your default web browser.

Go to localhost:5000

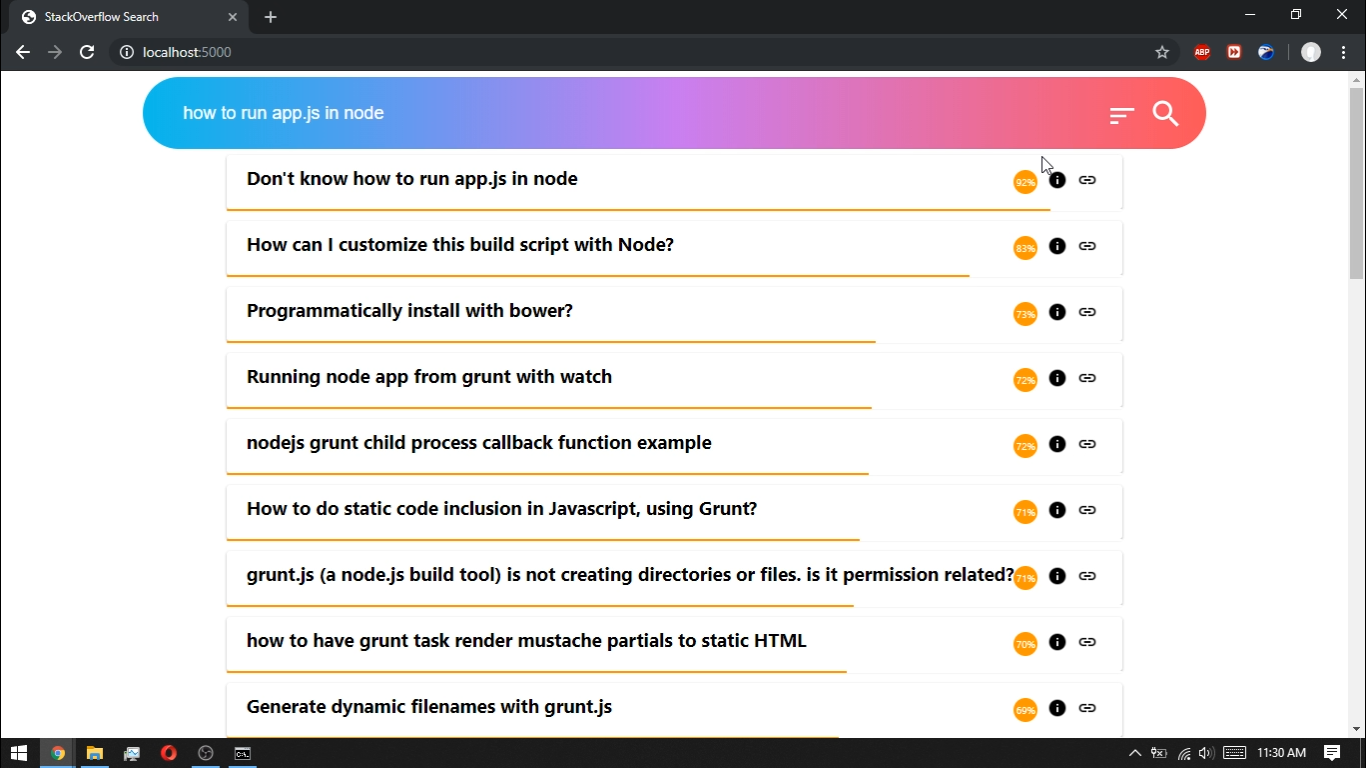


Now comes the fun part, testing.

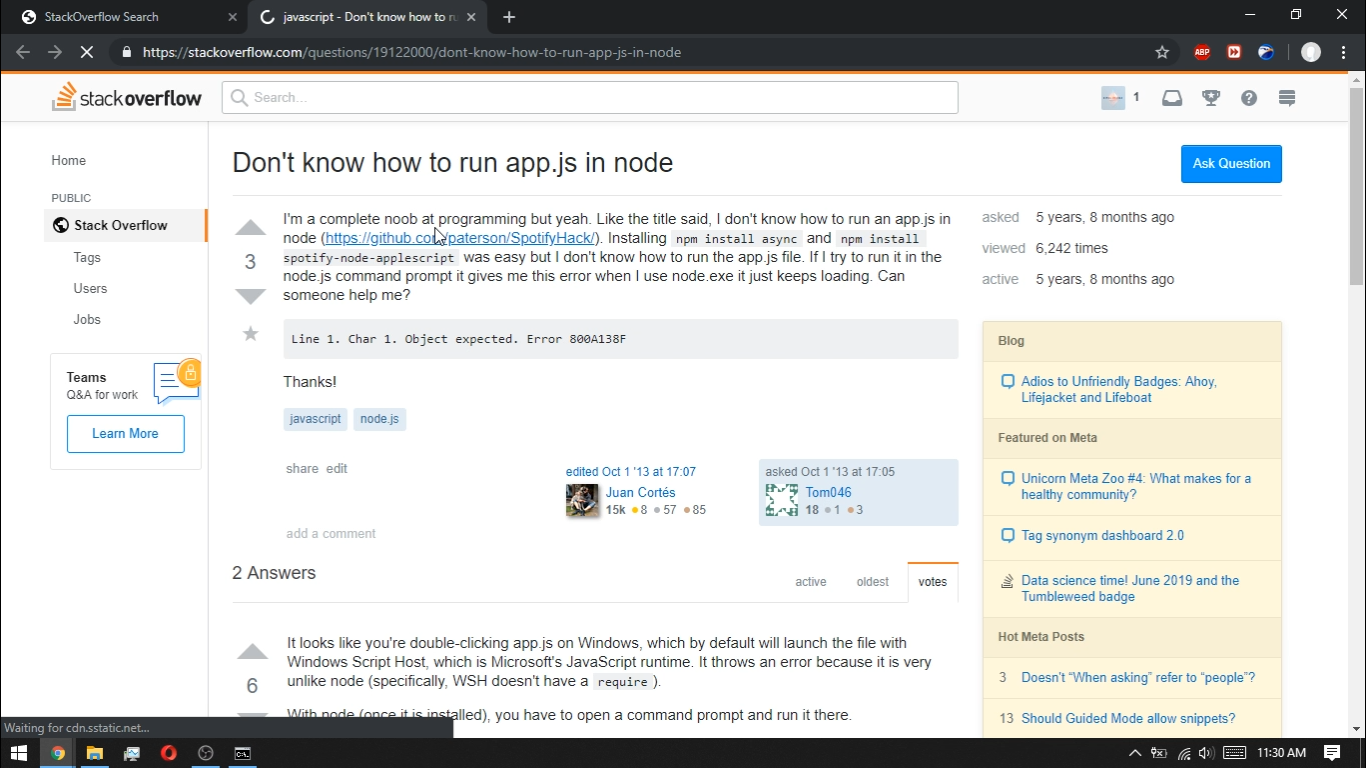
Before testing make sure you have a stable internet connection. For the first run alone, the TensorFlow hub downloads an 800MB file which is essential for proper running of the search engine.

**Break down of code**

* First the @app.route(“/”) is executed which calls an index.html file containing the search bar where we can enter the query. Now when we enter the query, the @app.route(“/compute”, method = [‘POST’]) is called upon which first stores our query as a string into a variable.
* Now this variable is passed into the function get\_tags of the core.py file to get tags. Process of tag extraction is done by the nltk package. Now these tags are sent to the Stack Overflow API to get back similar registered tags of Stack Overflow. This list of API tags is returned by the get\_tags function.
* Now the tags along with the query is sent as parameters to the get\_questions function of core.py. Here, first the elements of the list of tags are permuted within themselves to get all possible combinations of the elements. Now the list got is sent to the API of Stack Overflow to return us all the questions that are got from list items.
* Now for the questions got, the questions along with the query is sent to the get\_similarity function of code.py to get us the similarity percentage of the questions with the query. Now with the percentage in hand, the questions are sorted accordingly, and the server responds with the sorted question list and Stack Overflow URL links corresponding to the questions.



* Now all that is left for us is to click on URL link and we will be redirected to that respective Stack Overflow answers to that question.



**Deployment**

This search engine can be deployed in a cluster and using something like Apache Spark to create a distributed version of it, now the universe is not a single machine but many.

**Built With**

* React.js - The frontend framework
* Flask – web framework (for static file serving and computation)
* Stack Overflow API – To get the questions and their links related to the tags

**Versioning**

We use [Git](http://semver.org/) for versioning.

**Authors**

* **Nipun Haldar**
* **K Kesava**
* **Rahul Hebbar**

**License**

This project is licensed under the MIT License

**Acknowledgments**

* https://tfhub.dev/google/universal-sentence-encoder-large/3