Stack Overflow Search Engine

College of Computing & Informatics, Drexel University 3675,
Market Street
Philadelphia Pennsylvania 19104

Jeevan Reddy Geereddy (jg3687@drexel.edu)
Pradeep Kumar Kankipati (pk593@drexel.edu)
Suresh Athanti (sa3663@drexel.edu)
Karthikreddy Kuna (kk3375@drexel.edu)

ABSTRACT

In this paper, we present Stack overflow search engine, a web search engine which is built on elastic search engine. It is a flask web application running on elastic search to index the documents and retrieve the relevant document as per the user query.

As resources for learning new programming languages and tools grows, learners always tend to confuse and seek help from others. But most of the times they do not get exactly what they want so to cater to the needs of these new programmers we built a search engine on top of stack overflow dataset specifically for **python** related queries. Through which they can get relevant documents for their queries.

1.INTRODUCTION

Search Engines have become quite popular over decades and in today's trend, it is one of the mandatory tools for collecting information over the web. It has changed the manner individuals retrieve information and gain knowledge, increasing their scope about almost any subject making it quickly accessible.

As the need increased, lot of issues are being discovered related to search results and their accuracy was not delivered at full fledge which can meet all types of users' requirements. Different parameters were used for search engine implementations and it has become difficult in filtering the parameters which can give relevant results to the end user. One such issue identified by our team was with the Stack overflow website and the results provided by them.

Stack overflow comprises of an abundant measure of data in the configuration of inquiries and answers accordingly making it elusive the specific data you required for your undertaking. In simple terms, it is a question and answer site for professional and enthusiast programmers exchanging their knowledge with each other by answering the queries related to different technologies.

Due to the websites poor performance of their search systems, a user would face difficulty in getting doubts cleared and would thus decide to use a more advanced search engine like google for their purpose. It will be a problem only when the Google search engine suggests any other resource than stack overflow to fulfil the end user requirements. For every user that leaves their website in search of better solution, they lose revenue.

It is a lot simpler for a professional to explore through this jumble since they know precisely what they are searching for. They have abundant command on the subject as they can call for the right keyword to get their work done.

However, the huge amount of information makes it difficult for a new programmer to search for the solution he is looking for and might get intimidated. For instance, if he wants to search 'how to implement modelling in machine learning', it is obvious that he might not use the terms like 'linear regression', 'random forest' in the search box. It is dubious that he is aware of these key words. So, we decided to implement a search engine based on elastic search engine which can index the documents and retrieve the relevant documents based on the rank feature given to it thus it can helps users towards getting more relevant results.

2. SYSTEM FEATURES/FUNCTIONS

Stack overflow search engine is running on Elastic search API. The Elastic search is a distributed, open source search and can analyze text, including text tokenization and filtering and it can support all types of data, like textual, numerical, geospatial, structured, and unstructured.

2.1 Click Marking:

Click Marking feature is implemented to highlight user accessed documents in the results page. This feature helps to users to save time while surfing the retrieved documents.

2.2 Highly voted post first

Stack overflow search engine has another feature which is to list the stack overflow posts based on the most numbers of positive votes to it by the other users which helps new programmers to get the exact solution for their problem within top 1 or 2 documents on the results page.

3. Implemented system Architecture

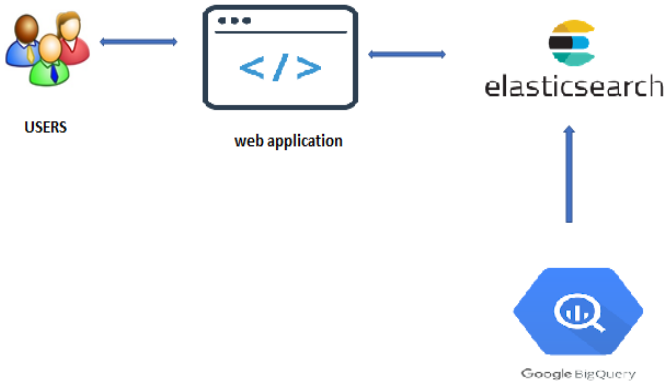


Fig1. Stack Overflow search engine's architecture

The Stack Overflow search uses the GoogleBigQuery to get the documents from the Archive stack overflow posts and feed the data into the elastic search engine to index the document and get the most relevant document as per user query. Our search engine is running on flask server to get the queries from users at the front end which is built using HTML and CSS. The UI interface interacts with the elastic search engine to filter the documents and displays on the web application.

In Elastic Search, we have used the "**Question_content**" field to be searched based on user's query and we have used "**over_all scores**" field for the ranking purpose in terms of document retrieval.

4.SOFTWARE/ HARDWARE REQUIREMENTS

Operating systems: windows/ windows XP/vista/Unix/Linux

Server: Flask

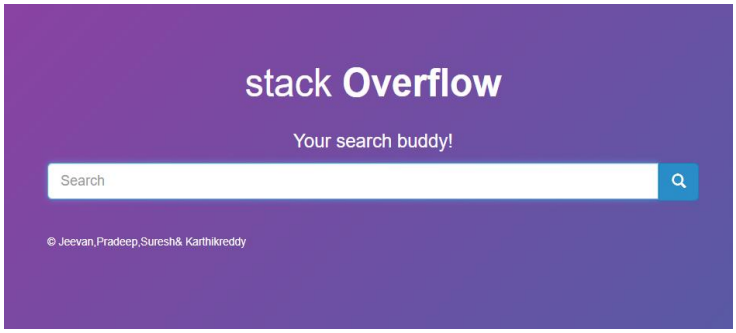
Python, HTML, CSS

Tools: Elastic Search, Google Big Query

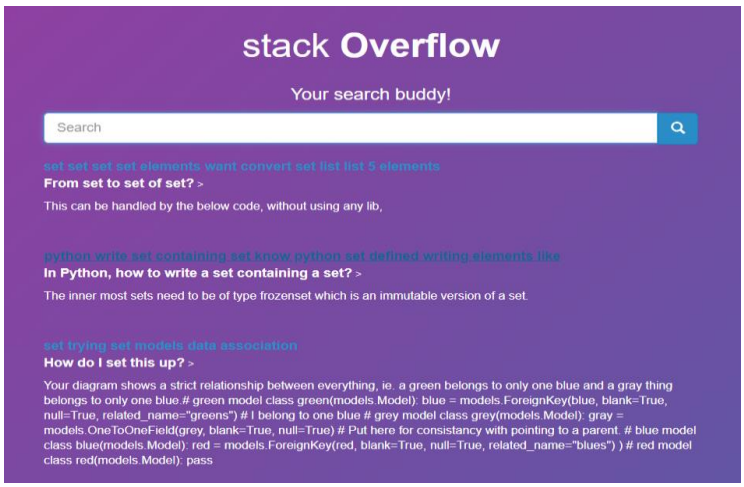
The current web application is running on flask server locally to run the application we need a web browser like IE 9, Google Chrome 11.0 and Firefox

5.SCREENSHOTS OF THE UI:

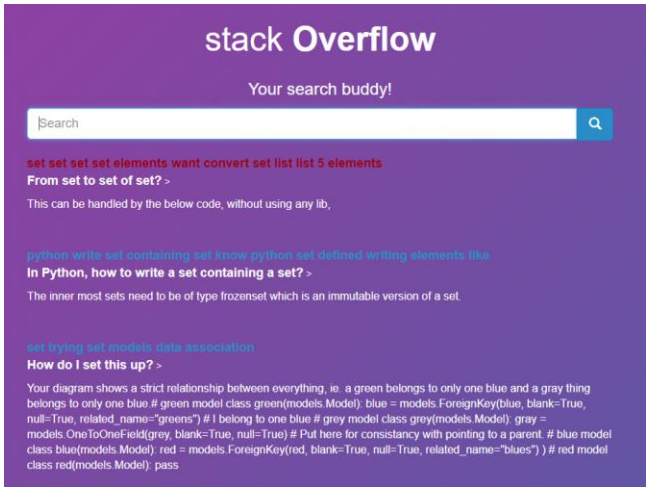
5.1 Search engine page UI



5.2 Search Engine results page



5.3 Search engine results page with Click marking feature



6. Evaluation of Search Engine

To understand the search engine statistics, we must evaluate the efficiency of it. We have used certain Information retrieval performance evaluation metrics on the retrieved results. nDCG is the metric which determine the accuracy of our results.

In our case, we have manually created different scenarios using different types and lengths of queries. In each case, thinking from end user's perspective, we have categorized results into relevant and non-relevant results and used them to calculate nDCG manually.
Below are the results and scores of different types of queries used during evaluation:

Query1: "**for loop syntax in python**"

nDCG=1

Explanation: requires python for loop syntax

Query2: "**How to store dataframe to csv**

nDCG:0.80

Exp: expected usage of to_csv() method

Query3:"**How to implement linear regression in python**"

nDCG:0.75

Exp: expected usage of sklearn.linear_regression method

Query 4: "**Java Polymorphism**"

nDCG:1

Exp: expected polymorphism in java explanation

Query 5 ;"**Dict vs list**"

nDCG:0.78

Exp: expected difference between list and dictionary datatypes in python

7. Limitations:

The following are some limitations that cannot be implemented due to time and space constraints.

1. Confinement of the data to only python related questions
2. Restricted the possible tags to only 14k.
3. Data points were confined to 200k for faster processing.

8. Future Improvements:

1. Enhancing UI design to give a very user-friendly feel.
2. Considering a greater number of tags and data points which would likely provide enhance results.
3. Assessing different other types of search engines and working on upgradation on current search engine.

9. Conclusion

Stack Overflow search engine is designed to provide more efficient, approachable and scalable results. It uses stack overflow API and immediate goal is to provide accurate results from stack overflow. It filters out results using keywords what user has searched. Our search engine was able to

successfully retrieve most relevant results in majority cases and the nDCG metrics that was used to evaluate it never gave us a disappointing score.

10.References:

[1]https://stackoverflow.com/questions
[2]https://www.elastic.co/guide/en/app-search/current/engines.html
[3]https://www.elastic.co/guide/en/logstash/current/plugins-outputs-google_bigquery.html
[4]https://www.xplenty.com/integrations/google-big-query/elasticsearch