Stack Overflow Data Mining

Sai Sathvick Chirakala
Computer Science
Texas Tech Unversity
sai-sathvick.chirakala@ttu.edu

Maitreyi Naik
Computer Science
Texas Tech Unversity
maitreyi.naik@ttu.edu

Nikhila Ratakonda
Computer Science
Texas Tech Unversity
nikhila.ratakonda@ttu.edu

***Abstract*—With the inception of World Wide Web, the amount of data present on the internet is tremendous. A large number of dataset repositories, catalogs and portals are emerging in the science and government realms. Once a large number of datasets are published on such data portals, the question arises how to retrieve datasets satisfying an information need. This makes the task of navigating through this enormous amount of data quite difficult for the user. In this work, we present a dataset which provides information on the questions that are posted in stack overflow, which is little bit noisy. Stackoverflow-data-idf.json file has 20000 questions posted and 19 columns. 19 columns include post title, body, tags, dates and other media.**

***Keywords*—IR, TF, IDF, TF-IDF**

The remainder of this paper is organized as follows: Introduction is explained in section 1; Procedures is in Section 2; Section 3 describes the results and discussions; section 4 gives the conclusions.

I. Introduction

Information Retrieval (IR) is finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers). It is the activity of obtaining information resources relevant to the information need from a collection of information system.

These days we frequently think first of web search, but there are many other cases:

• E-mail search
• Searching your laptop
• Corporate knowledge bases • Legal information retrieval

And the goal is to retrieve documents with information that is relevant to the user's information need and helps the user complete a task

II. Procedure

A. Review the Data Set Stage

A Stack overflow dataset is taken, which is a little noisy. Stackoverflow-data-idf.json file has 20000 questions posted and 19 columns. The columns include post title, body, tags, dates and other media. After preprocessing the data, total terms left are 666637.

B. Methodology

First, we need to do the preprocessing of dataset, i.e. removing stop words and doing text normalization the following from the dataset. See the flow chart.

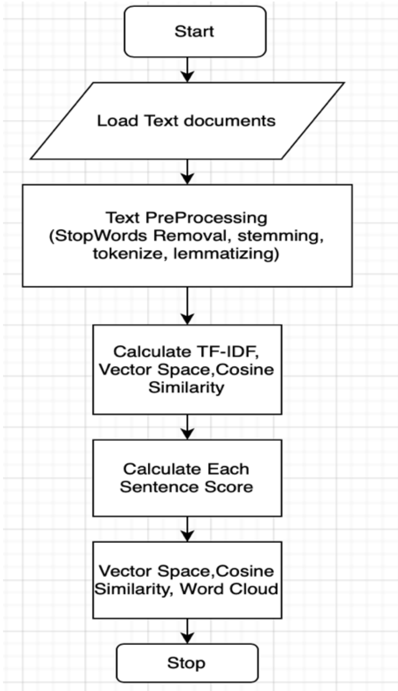


Fig.1.Flow Chart

2.1. Pre-Processing of the Data

i. **Stop words:** These are basically a set of commonly used words in any language [1]
ii. **Stemming: It** is the process of reducing a word to its word **stem** that affixes to suffixes and prefixes or to the roots of words known as a lemma [2]
iii. **Lemmatization: It** usually refers to doing things properly with the use of a vocabulary and morphological analysis of words [3]
iv. **Tokenize:** It is the process of tokenizing or splitting a string, text into a list of tokens [4]

This method is called preprocessing.

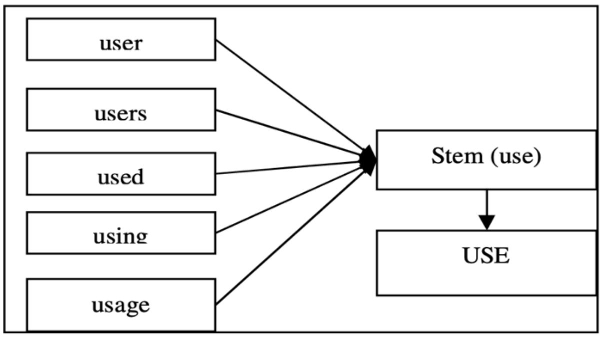


Fig.2. Stemming

2.2. TF-IDF

After the preprocessing we will define the query which is related to the document and calculate the term frequency, inverse document frequency.

- **i.** **Term Frequency: TF** (**Term Frequency**) measures the **frequency** of a word in a document. TF = (Number of time the word occurs in the text) / (Total number of words in text). [5]
- **ii.** **Inverse Document Frequency**: IDF = (Total number of documents / Number of documents with word t in it). [6]

Thus, the TF-IDF is the product of TF and IDF: [7]

TF-IDF = TF * IDF

2.3. Vector Space

A vector space model is an algebraic model, involving two steps, in first step we represent the text documents into vector of words and in second step we transform to numerical format so that we can apply any text mining techniques such as information retrieval, information extraction, information, filtering etc. [8]

2.4. Cosine similarity

Mathematically, closeness between two vectors is calculated by calculating the cosine angle between two vectors. In similar lines, we can calculate cosine angle between each document vector and the query vector to find its closeness. To find relevant document to the query term, we may calculate the similarity score between each document vector and the query term vector by applying cosine similarity. Finally, whichever documents having high similarity scores will be considered as relevant documents to the query term. [9]

When we plot the term document matrix, each document vector represents a point in the vector space. In the below example query, Document 1 and Document 2 represent 3 points in the vector space. We can now compare the query with each of the document by calculating the cosine angle between them. Fig.3.

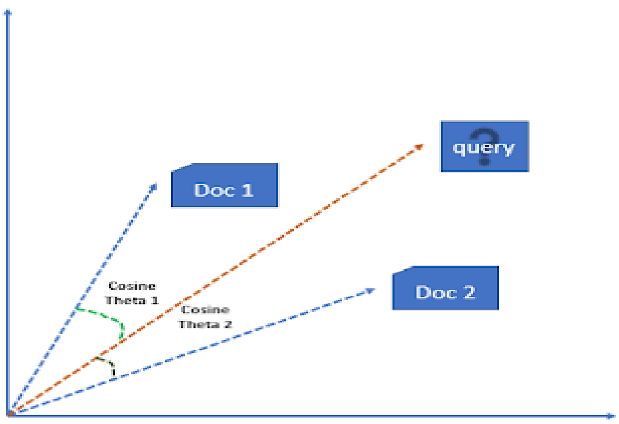


Fig. 3. cosine similarity

$$\vec{a} \cdot \vec{b} = \|\vec{a}\| \|\vec{b}\| \cos\theta$$

$$\cos\theta = \frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\| \|\vec{b}\|}$$

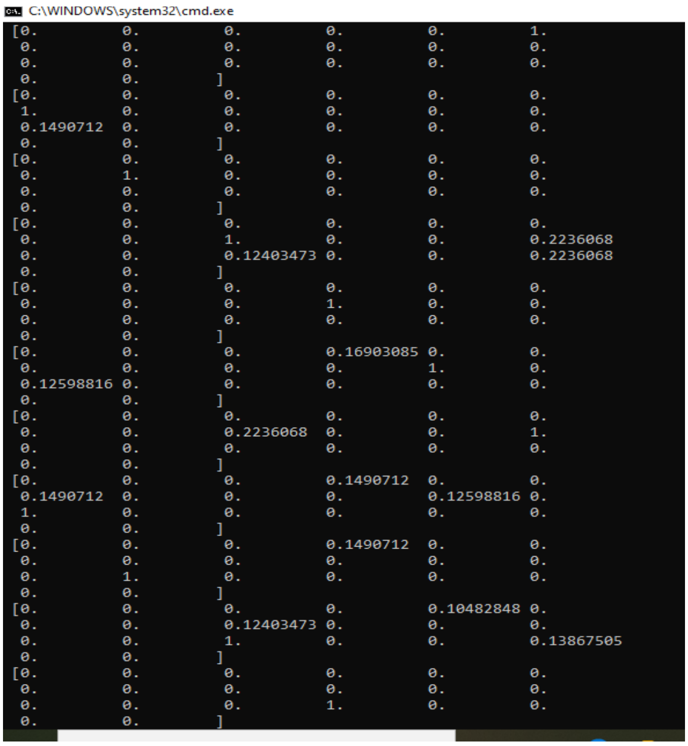
III. RESULTS

i. IDF

Variabl	7.896572202700973
Python	7.156172137290483
Color	7.381722428635257
String	5.551360293245164
Java	7.08206416513676
Framework	7.800529153680996
Ajax	7.815289569264116
Git	8.030103595325496
Help	5.585555017988336
Null	7.1504194848010325
File	4.984484546021622

Fig. 4. IDF

ii. Tf-idf Score



```
C:\WINDOWS\system32\cmd.exe
[0.         0.         0.         0.         0.         1.
 0.         0.         0.         0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.         0.        ]
[0.         0.         0.         0.         0.         0.
 1.         0.         0.         0.         0.         0.
 0.1490712  0.         0.         0.         0.         0.
 0.         0.        ]
[0.         0.         0.         0.         0.         0.
 0.         1.         0.         0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.         0.        ]
[0.         0.         0.         0.         0.         0.
 0.         0.         1.         0.         0.         0.2236068
 0.         0.         0.12403473 0.         0.         0.2236068
 0.         0.        ]
[0.         0.         0.         0.         0.         0.
 0.         0.         0.         1.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.         0.        ]
[0.         0.         0.         0.16903085 0.         0.
 0.         0.         0.         0.         1.         0.
 0.12598816 0.         0.         0.         0.         0.
 0.         0.        ]
[0.         0.         0.         0.         0.         0.
 0.         0.         0.2236068  0.         0.         1.
 0.         0.         0.         0.         0.         0.
 0.         0.        ]
[0.         0.         0.         0.1490712  0.         0.
 0.1490712  0.         0.         0.         0.12598816 0.
 1.         0.         0.         0.         0.         0.
 0.         0.        ]
[0.         0.         0.         0.1490712  0.         0.
 0.         0.         0.         0.         0.         0.
 0.         1.         0.         0.         0.         0.
 0.         0.        ]
[0.         0.         0.         0.         0.10482848 0.
 0.         0.         0.12403473 0.         0.         0.
 0.         0.         1.         0.         0.         0.13867505
 0.         0.        ]
[0.         0.         0.         0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.         0.         0.         1.         0.         0.
 0.         0.        ]
```

Fig. 4. Tf-Idf Score

iii. Term frequency bar chart

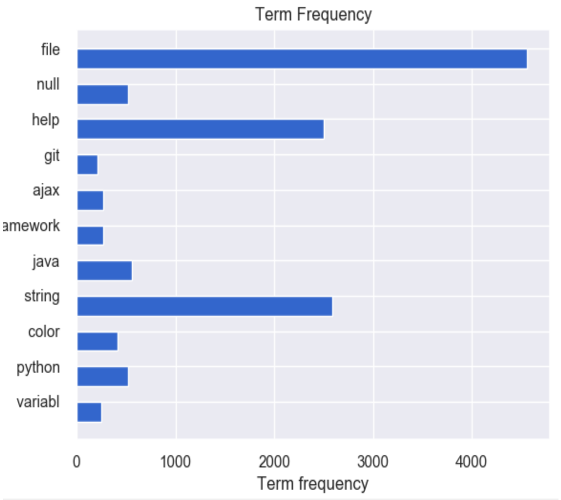


Fig. 5. Term Frequency Bar Chart

iv. IDF Bar Chart

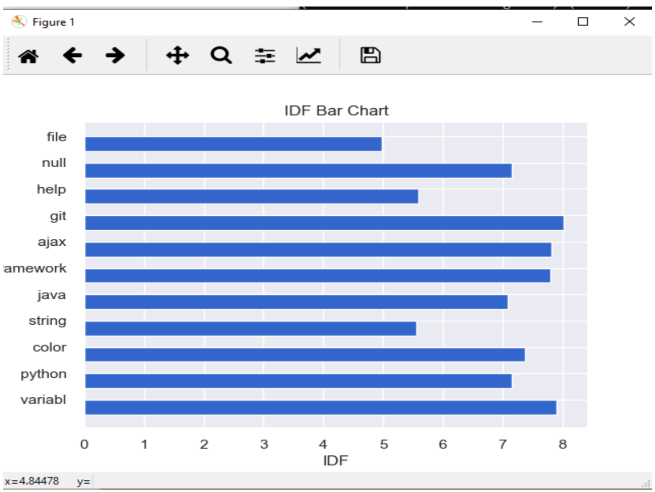


Fig. 6. IDF Bar Chart

v. Vector Space Heat Map

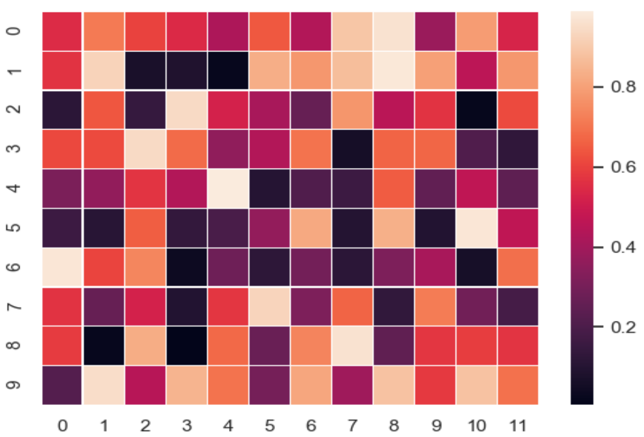


Fig. 7. Vector Space Heat Map

vi. Cosine Similarity

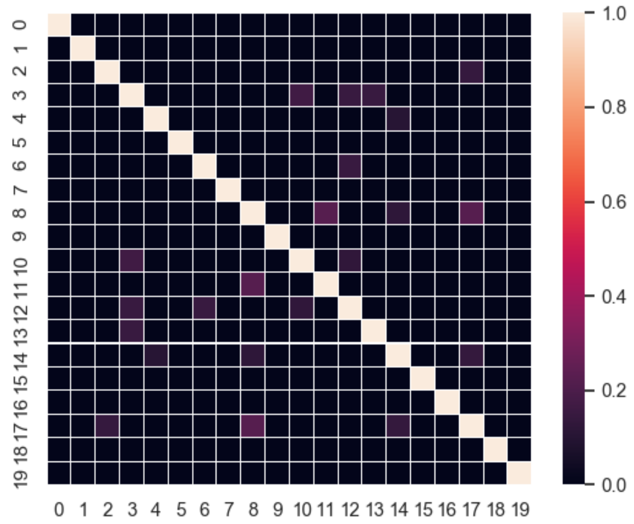


Fig. 8. Cosine Similarity

vii. World Cloud

Fig. 9. Word Cloud

IV. CONCLUSION

We have seen that TF-IDF is an efficient and simple algorithm for matching words in a query to documents that are relevant to that query. From the data collected, we see that TF-IDF returns documents that are highly relevant to a particular query. If a user were to input a query for a particular topic, TF-IDF can find documents that contain relevant information on the query. Furthermore, encoding TF-IDF is straightforward, making it ideal for forming the basis for more complicated algorithms and query retrieval systems

Despite its strength, TF-IDF has its limitations. In terms of synonyms, notice that TF-IDF does not make the jump to the relationship between words. Since TF-IDF is merely a staple benchmark, numerous algorithms have surfaced that take the program to the next level.

Enhancing the already powerful TF-IDF algorithm would increase the success of query retrieval systems, which have quickly risen to become a key element of present global information exchange.

References

[1] Wilbur, W. John, and Karl Sirotkin. "The automatic identification of stop words." *Journal of information science*18.1 (1992): 45-55.

[2] Lovins, Julie Beth. "Development of a stemming algorithm." *Mech. Translate. & Comp. Linguistics* 11.1-2 (1968): 22-31.

[3] Plisson, Joël, Nada Lavrac, and Dunja Mladenic. "A rule-based approach to word lemmatization. " *Proceedings of IS*. Vol. 3. 2004.

[4] Singh, Vikram, and Balwinder Saini. "An Effective tokenization algorithm for information retrieval systems." *Department of Computer Engineering, National Institute of Technology Kurukshetra, Haryana, India* (2014).

[5] Xia, Tian, and Yanmei Chai. "An improvement to TF-IDF: Term Distribution based Term Weight Algorithm." *JSW* 6.3 (2011): 413-420.

[6] Robertson S. Understanding inverse document frequency: on theoretical arguments for IDF. Journal of documentation. 2004 Oct 1.

[7] Ramos J. Using tf-idf to determine word relevance in document queries. In Proceedings of the first instructional conference on machine learning 2003 Dec 3 (Vol. 242, pp. 133-142)

[8] Salton, Gerard, Anita Wong, and Chung-Shu Yang. "A vector space model for automatic indexing." *Communications of the ACM* 18.11 (1975): 613-620.

[9] Muflikhah, L., & Baharudin, B. (2009, November). Document clustering using concept space and cosine similarity measurement. In *2009 International Conference on Computer Technology and Development* (Vol. 1, pp. 58-62). IEEE.