

NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA

Mini Project Report on

DEXTER: A SEARCH ENGINE FOR DATA DRIVEN DIAGRAMS

May 4, 2018



Submitted to:

Dr. Sowmya Kamath

Submitted by:

Aiman Abdullah Anees 15IT106

Salman Shah 15IT241

Rashika Chowlek 15IT135

Adwaith C D 15IT105

Certificate

This is to certify that this project report entitled “Dexter:Data driven diagram extractor” submitted to Department of Information Technology,NITK, is a bonafide record of work done by the team members under my supervision from January 2018 to April 2018.

Name of Instructor:

Signature of Instructor:

Declaration

We hereby declare that the project entitled ”**Dexter:A search engine for Data Driven Diagrams**” submitted is a record of original work by **Aiman Abdullah Anees (15IT106),Salman Shah(15IT241),Rashika Chowlek (15IT135), Adwaith C D (15IT105)** under the guidance of Dr. Sowmya Kamath,Dept of Information Technology, and this project is submitted in the partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Information Technology. The results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree.

Abstract

A lot of information is accessible just through information driven outlines which pass on more data and can be found as set of pictures containing Metadata. These graphs are a half and half of joining illustrations and message and are the consequence of confounded information driven generation pipeline frameworks. Sadly, neither content nor picture web indexes use these outline particular properties, making it troublesome for clients to discover important charts in an extensive corpus. To counter and take care of this issue, we propose Dexter, an internet searcher for discovering information driven graphs on the web. By recuperating the semantic parts of outline segments (e.g., the axis labels, the maximum values of the axis etc), we give ordering and recovery to different measurable graphs. A one of a kind component of Dexter is that it can "extend" questions to incorporate precisely coordinating outlines, as well as graphs that are probably going to be identified with the given Search Query.

Contents

1	Introduction	5
2	Literature Review	6
2.1	Background	6
2.2	Identified Gaps	6
2.3	Problem Statement	7
2.4	Objectives	7
3	Methodology	8
3.1	Offline Mode	8
3.2	Online Mode	9
4	Implementation	11
4.1	Work Done	11
4.1.1	Generation of dataset	11
4.1.2	Feature extraction	11
4.1.3	Preprocessing of text	11
4.1.4	Term weighing	12
4.1.5	Ranking of corpus	12
4.2	Results and analysis	14
4.3	Innovative Work	15
4.4	Details of each individual's work w.r.t. project tasks.	15
5	Conclusion and Future Work	17

List of Figures

figures	
Figure 1.....	8
Figure 2 and 3.....	9
Figure 4	10
Figure 5 and 6	12
Figure 7	13

List of Tables

Table 1	6-7
---------------	-----

1 Introduction

Information driven outlines (or factual designs) are a vital technique for conveying complex information. Diagrams, an adapted blend of illustrations and content, offer concise quantitative synopsis of information. Graphs offer quantitative outlines of information that motivate the general record's content. For numerous specialized archives, the graph might be searchers' only access to the crude information. For quantitative disciplines such as finance, public policy, and the sciences, certain graphs could be considerably more profitable than the encompassing text. Standard content based search might be able to recover the archives encasing the graphs. Image based search engines, which for the most part work by inspecting textual content that encompasses pictures instead of the visual characteristics of the pictures themselves, may recover a few graphs. Recently, some business seek systems, for example, Zanran and others can be used to query data driven diagrams. Be that as it may, searching systems to date have overlooked one distinct nature of information driven charts: an outline is the final result of a multistep pipeline. To begin with, the chart creator must pick a dataset to imagine, which is frequently only a little part of the aggregate accessible information. Second, the creator characterizes a "particular" of what they need showed either automatically or on the other hand through direct control. At last, a program takes the information and detail and renders a graphical show. These means are conceivably lossy: the envisioned dataset is likely smaller than the aggregate accessible database and both the creator and rendering framework may settle on choices about what imprints to display. Dexter is a Search Engine for Data Driven Diagrams that can "grow" questions to incorporate precisely coordinating outlines, as well as charts that are probably going to be connected as far as their generation pipelines. We exhibit the resulting search system by ordering pictures pulled from different datasets. The dataset is in the form of graphs and spreadsheet information.

2 Literature Review

2.1 Background

Graphs offer quantitative outlines of information that motivate the general record's content. For numerous specialized archives, the graph might be searchers' only access to the crude information. For quantitative disciplines such as finance, public policy, and the sciences, certain graphs could be considerably more profitable than the encompassing text. Neither content nor picture web crawlers use these graph particular properties, making it troublesome for clients to discover pertinent graphs in an extensive corpus.

2.2 Identified Gaps

AUTHORS	METHODOLOGY	ADVANTAGES	LIMITATIONS
Sumit Bhatia, Prasenjit Mitra and C. Lee Giles	The total similarity score for an algorithm is the sum of three similarity scores.i.e. a TF-IDF based cosine similarity score is computed between i)(query, caption), (ii) (query, reference sentence) and (iii) (query,synopsis).	The user is presented top 10 algorithms for the query, along with their associated metadata. The algorithm caption is presented in bold and clicking on it directly takes the user to the PDF page of the concerned document on which the algorithm is present	The keyword “algorithm” needed to be added in hope to get more algorithms in the results. A result was considered relevant if it contained a valid algorithm.

Mohamed Abdel Fattah	Presenting comprehensive investigation of different proposed new term weighting schemes	The proposed term weighting approach results outperform traditional term weighting schemes	The total performance as a result of the proposed term weighting schemes is not improved to a great extent.
S. Carberry, S. Elzer, and S. Demir.	To recognize an information graphic's message,they have developed a Bayesian network to infer the message conveyed by one type of information graphic,simple bar charts.	Recognizing the primary message conveyed by an information graphic.	The model was applied only on simple bar charts.

2.3 Problem Statement

A search engine for discovering information driven charts on the web by recuperating the semantic parts of outline components(e.g., axes,labels,etc.) additionally to "extend" inquiries to incorporate precisely coordinating graphs, as well as outlines that are probably going to be connected to them.

2.4 Objectives

- Generating the dataset manually .
- To implement both offline and online model of search engine.
- Offline model includes a Diagram Extracter and an Index Builder.
- Online model includes a Search Ranker, Query Expander.
- To improve the search ranker technique by carrying out comprehensive investigation of different term weighting schemes.
- Providing Keyword search, Advanced Search and Similar Diagram functionality.

3 Methodology

The methodology proceeds in two ways. First the offline stage of Dexter needs to be handled. Then the online stage is taken care of.

3.1 Offline Mode

- 1) Graphs were generated using datasets fetched from various websites. Summary for each graph was created.
- 2) For all the PDF documents, a separate latex document was created. This was done so that features can be extracted easily.
- 3) Feature extraction was carried out using simple regex and all the features were stored in JSON format. 8 key fields are generated: Title, X-Label, Y-Label, X-min,X-max, Y-min, Y-max, and Description.
- 4) We are using a VSM model(Vector Space Model) where document and queries are represented as a point in n-dimensional vector space R_n where n is the size of the index vocabulary.
- 5) Preprocessing is carried out on all the fields in JSON document. It consists of i)normalization ii)tokenazitaion iii)lemmatization and iv)cleaning processes(removing tokens that contain punctuations,removing tokens that are just punctuations and removing tokens that have one character). This produces a multidimensional array where each array is a document. Also a vocabulary is created.
- 6) TF-IDF(Term Frequency-inverse Document Frequency)is carried out using the vocabulary to reflect how important a word is to a document in a collection or corpus and a multidimensional array created.
- 7) Cosine similarity used to rank the documents. Cosine similarity measures the similarity bu the calculating the cosine of the angles between the non zero vectors

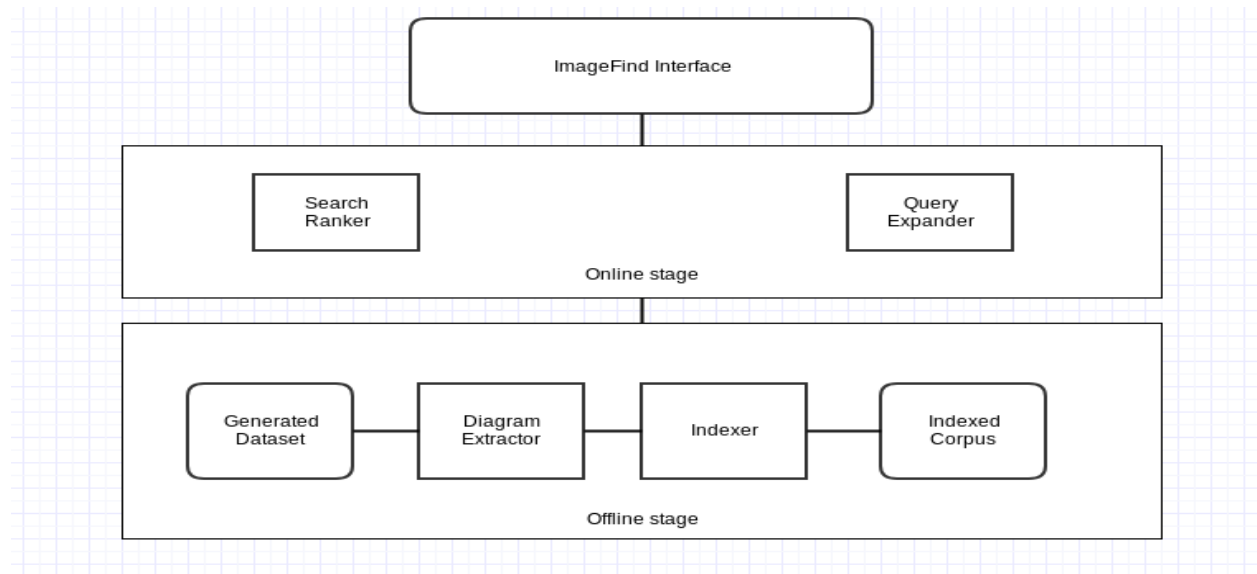


Fig 1: System Architecture

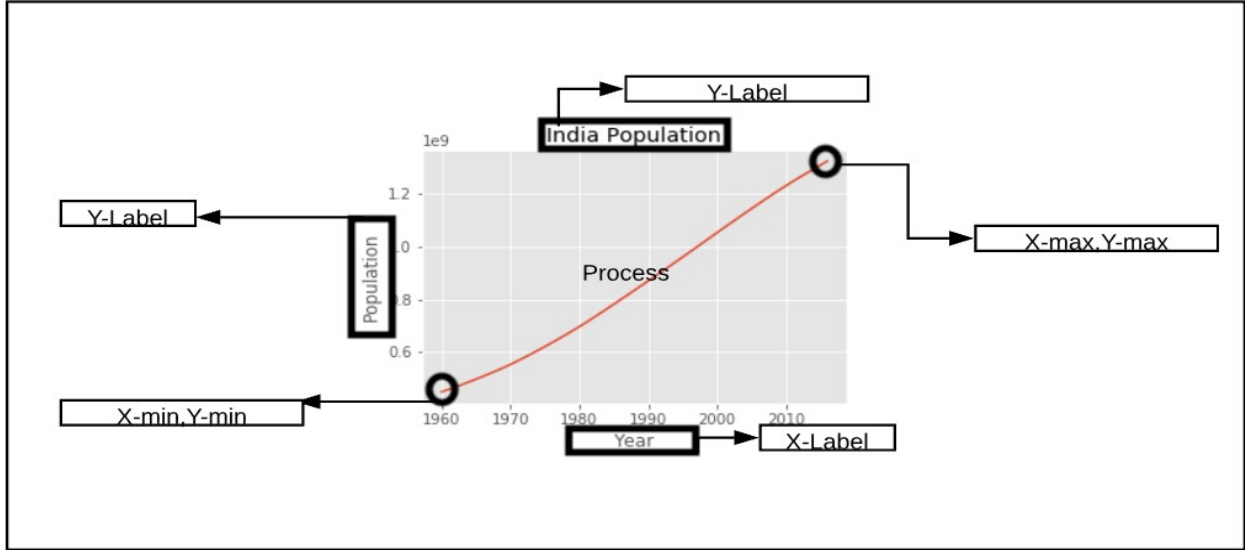


Fig 2:Labels(Seven of the eight fields shown above, eight field is description produced by wikiApi)

3.2 Online Mode

- 1)The approach involved calculating basic TFIDF score for each and every file and then using it to rank the documents in the descending order of their ranking.
- 2)cosine-similarity is used after this to rank the documents in the decreasing order of their ranking.

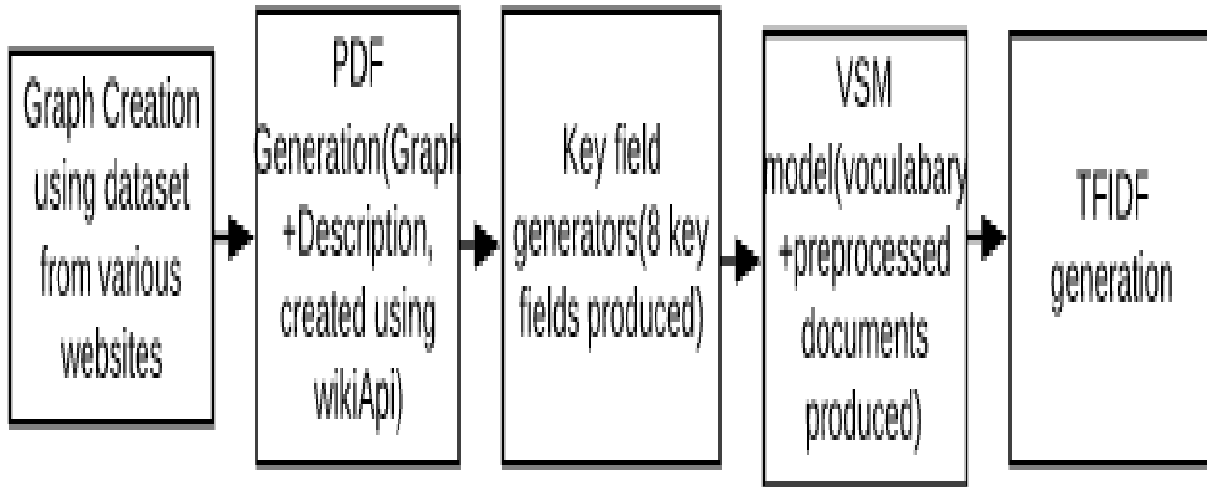


Fig 3:Offline mode flow chart

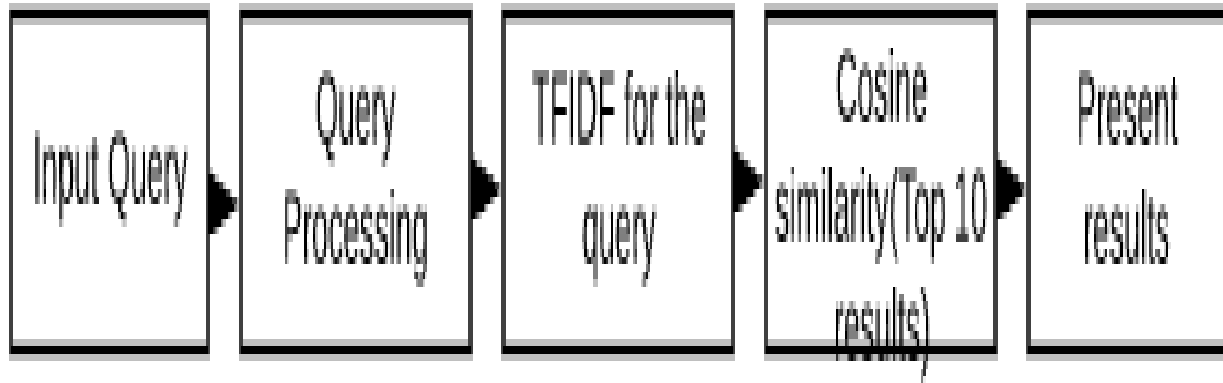


Fig 4:Online mode flow chart

4 Implementation

4.1 Work Done

Offline mode

Initially we started with the boolean model. Because of the following disadvantages we moved on to the VSM model

- It is often not simple to translate an information need into a Boolean Expression.
- Most users find it difficult and awkward to express their query requests in terms of Boolean expressions
- It doesn't provide ranking.

4.1.1 Generation of dataset

The dataset includes pdfs, graphs and spreadsheet data collected from various websites. Graphs are generated from the dataset. Summary for each graph is created using wikiAPI, a python library for extracting summary from Wikipedia for a particular topic. The graph and summary generated are added to a PDF file created using reportLab, a python library for generating PDF files. For all PDF documents, a separate latex document is created. This is done so that the features can be extracted easily. We use the matplotlib2tikz for converting matplotlib figures into PGFPlots (TikZ) for native inclusion into LaTeX documents.

4.1.2 Feature extraction

Feature extraction is carried out using simple regex and all the features are stored in JSON format. 8 key fields are generated namely: title, x-label, y-label, x-min, y-min, x-max, y-max and description.

4.1.3 Preprocessing of text

Initially preprocessing is carried out on the on all the fields in JSON document. It consists of Tokenization, Normalization, Lemmatization, Stopword Removal (Generating tokens, converting the words into lowercase, reducing tokens to their base or dictionary form, removing the commonly used words/stopwords), Cleaning processes (removing tokens that contains punctuations, removing tokens that are just punctuations, removing tokens that have one character) and vocabulary creation. This produces multidimensional array where each array is a document. We have used nltk library for preprocessing steps. Preprocessing is carried out on all the fields in JSON document. It consists of i) normalization ii) tokenazitaion iii) lemmatization and iv) cleaning processes (removing tokens that contain punctuations, removing tokens that are just punctuations and removing tokens that have one character). This produces a multidimensional array where each array is a document. Also a vocabulary is created.

4.1.4 Term weighing

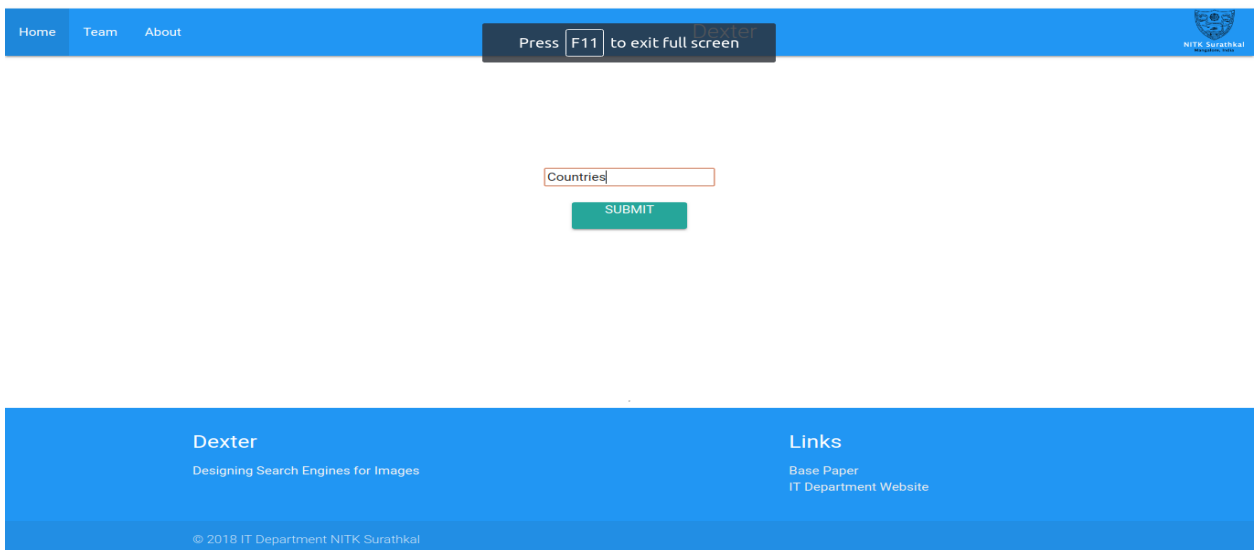
Each term in a document is assigned task using the vocabulary. This is done for all documents in the corpus. The term weighting process is done by assigning weight according to the tf-idf relevance score calculated for each term. tf-idf short for term frequency-inverse document frequency, is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus. It is often used as a weighting factor in searches of information retrieval, text mining, and user modeling. The tf-idf value increases proportionally to the number of times a word appears in the document and is offset by the frequency of the word in the corpus, which helps to adjust for the fact that some words appear more frequently in general. A VSM model (Vector Space Model) is used where document and queries are represented as a point in n-dimensional vector space R^n where n is the size of the index vocabulary. The multidimensional array stores these weights.

4.1.5 Ranking of corpus

The documents are ranked using cosine similarity of a document and the query entered.

Online Mode

- 1) The approach involved calculating basic TFIDF score for each and every file and then using it to rank the documents in the descending order of their ranking.
- 2) The tf for each term was calculated and stored in a file called tf.txt. After this the idf for each file was calculated and stored in the idf.txt.
- 3) The two values were then multiplied and stored in a file called TFIDF.txt.
- 4) cosine-similarity is used after this to rank the documents in the decreasing order of their ranking.



Home Team About

Press F11 to exit full screen

Dexter

NITK Surathkal

Countries

SUBMIT

Dexter

Designing Search Engines for Images

Links

Base Paper

IT Department Website

© 2018 IT Department NITK Surathkal

Fig 5:Result with a specific range

The Search Engine allowed users to enter a valid query and based on that query, appropriate Top 10 results would be generated and then sent across the user. This leads the user to results page where relevant Documents for the Users Query is retrieved.

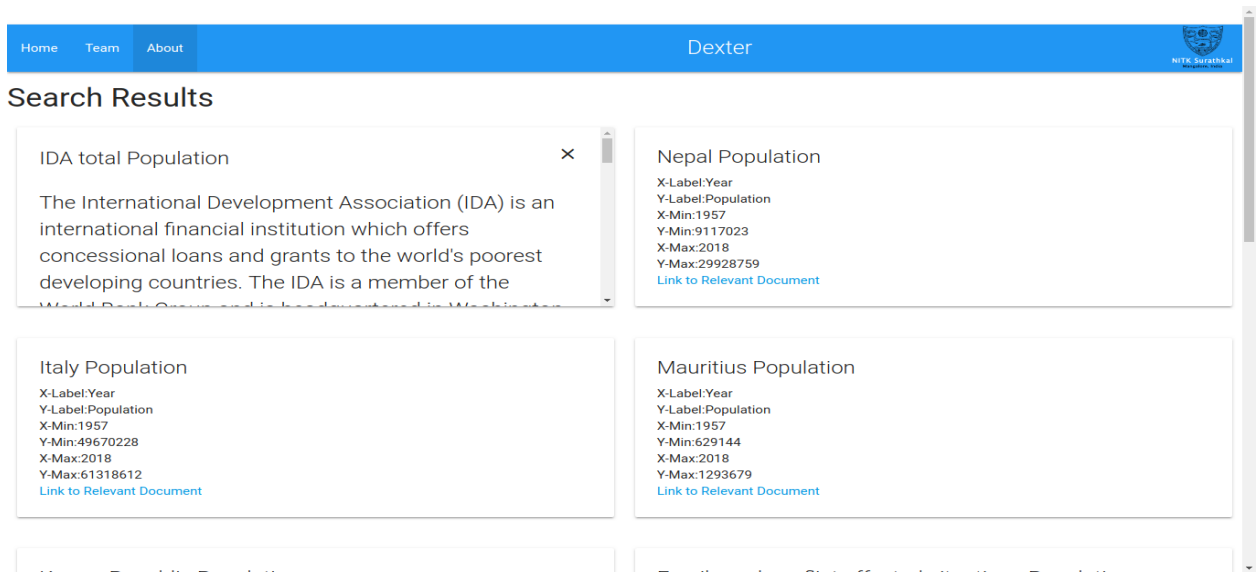


Fig 6: Result with no specific range

Once the query has been entered by the User, it leads to him to the Search Results Page, where he can find parameters such as the following for his query:

1. Title
2. Description
3. X-Label
4. Y-Label
5. X-Max
6. Y-Max
7. X-Min
8. Y-Min
9. Relevant Document

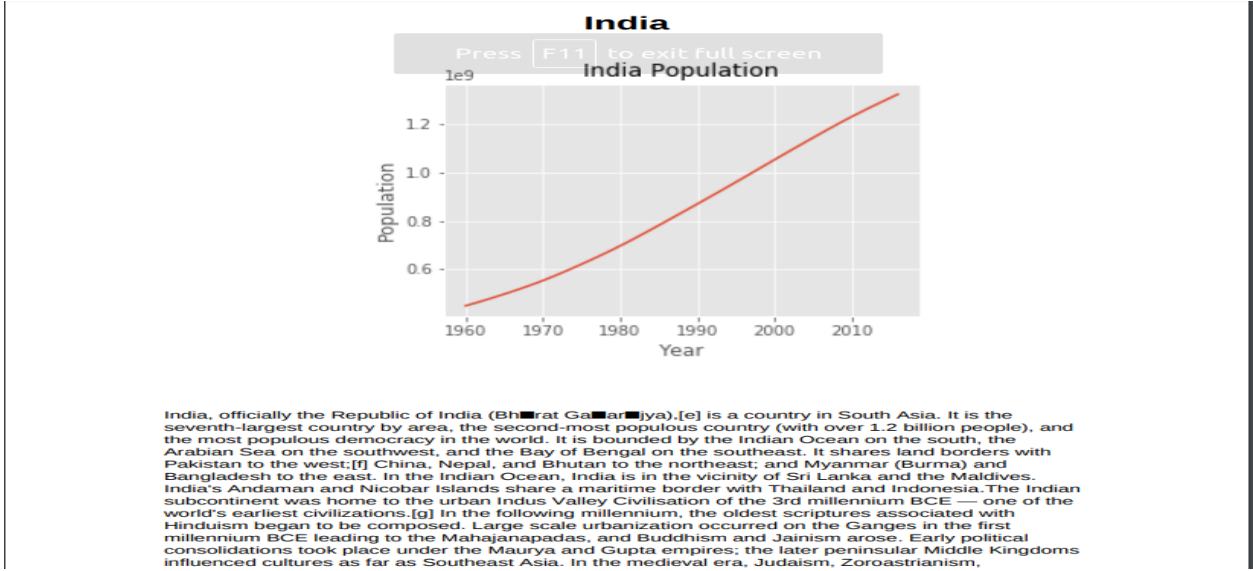


Fig 7:Result with an upper bound

The online mode of the application also allows users to view the associated / PDF linked with their user query. The PDF of the document contains the Data Driven Diagram associated with the query and a description of that context.

4.2 Results and analysis

Evaluation Metrics The metrics considered are precision, recall, average precision and R precision.

Query1-India population from 2005 to 2010

Precision	0.27
Recall	0.75
Average Precision	0.32
R Precision	0

Query2-Arab world population statistics

Precision	0.63
Recall	1
Average Precision	0.46
R Precision	0.42

Query3-Population of East Asia and Pacific Countries

Precision	0.63
Recall	0.875
Average Precision	0.96
R Precision	0.875

Query4-Central Europe population since 1960

Precision	0.36
Recall	1
Average Precision	0.67
R Precision	0.75

Query5- Population in highly indebted poor countries

Precision	0.18
Recall	0.4
Average Precision	1
R Precision	0.4

4.3 Innovative Work

Graphs were generated using datasets fetched from various websites. Summary for each graph was created. For all the PDF documents, a separate latex document was created. This was done so that features can be extracted easily. Feature extraction was carried out using simple regex and all the features were stored in JSON format. 8 key fields are generated: Title, X-Label, Y-Label, X-min, X-max, Y-min, Y-max, and Description. Preprocessing is carried out on all the fields in JSON document. It consists of Tokenization, Normalization, Lemmatization, Stopword Removal (Generating tokens, converting the words into lowercase, reducing tokens to their base or dictionary form, removing the commonly used words/stopwords), Cleaning processes (removing tokens that contain punctuations, removing tokens that are just punctuations, removing tokens that have one character).

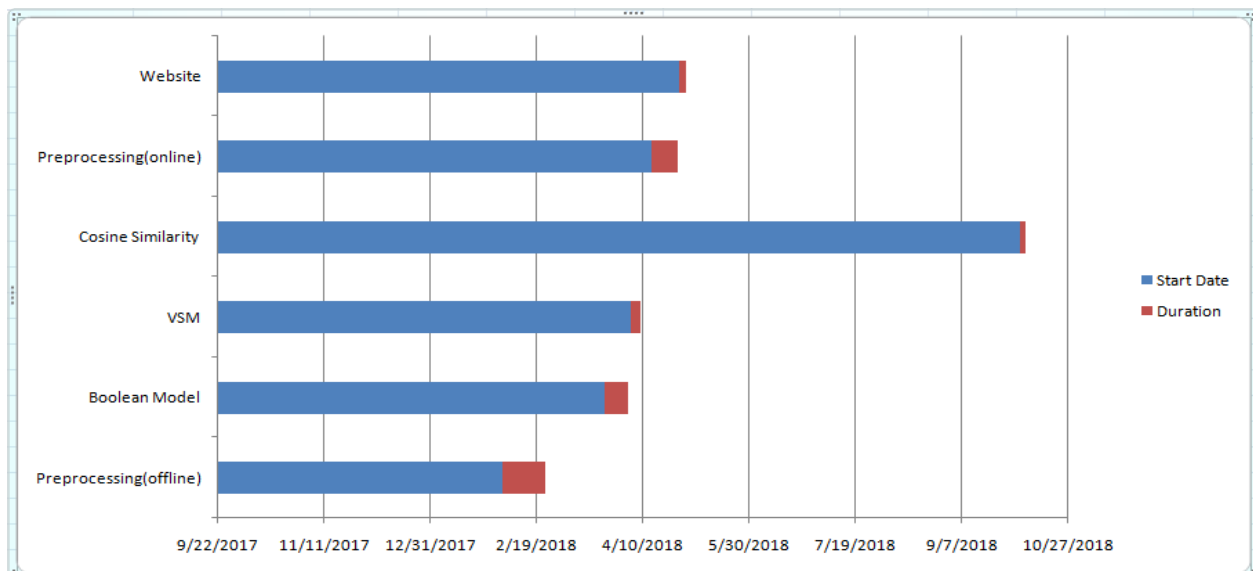
4.4 Details of each individual's work w.r.t. project tasks.

Offline mode (preprocessing, boolean model, VSM, cosine similarity):

Aiman And Salman

Online mode (preprocessing, website, System Evaluation):

Rashika and Adwaith



5 Conclusion and Future Work

We have implemented both the online and offline phases of search engine. The offline mode is basically comprised of fetching csv datasets from various websites and using these to generate graphs with their descriptions and finally making pdfs so that they can be ranked using cosine similarity. The online mode involves finding the term frequency for the input query and then calculating the TF-IDF from the IDF generated using the offline mode. We have created a website for displaying the overall functionality of the search engine.

References

1) New term weighting schemes with combination of multiple classifiers for sentiment analysis
<http://repository.taibahu.edu.sa/en/bitstream/handle/123456789/14629/New>

Appendix

<https://dl.acm.org/citation.cfm?id=2742831>

ORIGINALITY REPORT

11%

SIMILARITY INDEX

6%

INTERNET SOURCES

8%

PUBLICATIONS

%

STUDENT PAPERS

PRIMARY SOURCES

1	Sumit Bhatia, Prasenjit Mitra, C. Lee Giles. "Finding algorithms in scientific articles", Proceedings of the 19th international conference on World wide web - WWW '10, 2010 Publication	3%
2	www.cosy.informatik.uni-bremen.de Internet Source	1%
3	shodhganga.inflibnet.ac.in Internet Source	1%
4	www.papercamp.com Internet Source	1%
5	www.cis.udel.edu Internet Source	1%
6	www.studymode.com Internet Source	1%
7	iieng.org Internet Source	1%

Mohamed Abdel Fattah. "New term weighting

8 schemes with combination of multiple classifiers for sentiment analysis", Neurocomputing, 2015
Publication

9 www.geo.ed.ac.uk
Internet Source

10 Tuarob, Suppawong, Sumit Bhatia, Prasenjit Mitra, and C. Lee Giles. "AlgorithmSeer: A System for Extracting and Searching for Algorithms in Scholarly Big Data", IEEE Transactions on Big Data, 2016.
Publication

11 Kewen Chen, Zuping Zhang, Jun Long, Hao Zhang. "Turning from TF-IDF to TF-IGM for term weighting in text classification", Expert Systems with Applications, 2016
Publication

12 Ko, Youngjoong. "A new term-weighting scheme for text classification using the odds of positive and negative class probabilities", Journal of the Association for Information Science and Technology, 2015.
Publication

13 Dionysios P. Xenos, Georgios M. Kopanos, Matteo Ciccotti, Nina F. Thornhill. "Operational optimization of networks of compressors considering condition-based maintenance",

14

Carberry, Sandra, Stephanie Elzer, Richard Burns, Peng Wu, Daniel Chester, and Seniz Demir. "Information Graphics in Multimodal Documents", Multimedia Information Extraction Advances in Video Audio and Imagery Analysis for Search Data Mining Surveillance and Authoring, 2012.

Publication

<1%

15

Sandra Carberry, Stephanie Elzer, Seniz Demir. "Information graphics", Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '06, 2006

Publication

<1%

Exclude quotes On

Exclude matches

< 3 words

Exclude bibliography On