Code Retrieval based on Weighted Similarities of Control Flow & Structure

Suraj Pandey	MT18025
William Scott Paka	MT18026
Udit Pant	MT18049

https://github.com/williamscott701/stack-overflow-code-retrieval

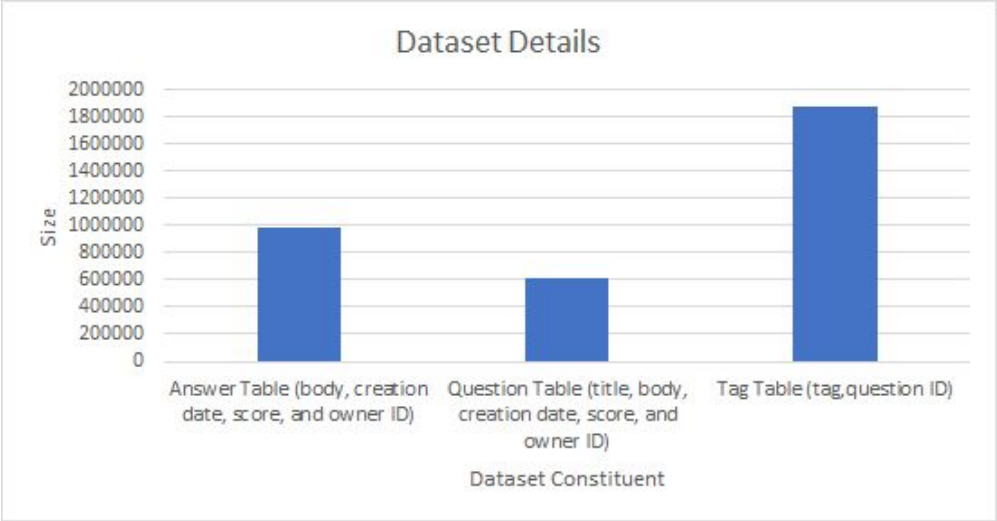
Problem Statement

- Our project aims at developing a robust code retrieval model for python
- Applying information retrieval techniques to suggest relevant code snippets
- Code similarity based on control flow and structure

Dataset Insights*

This dataset is organized into three tables:

- Question Table
- Answers table
- Tags table
- The dataset contains questions all question asked between August 2, 2008 and October 19, 2016



https://www.kaggle.com/stackoverflow/pythonquestions/downloads/pythonquestions.zip/1

Baseline Models

Baseline Models Implemented

- Text Similarity
  - TF with Euclidean Distance Similarity
  - TF-IDF with Euclidean Distance Similarity
  - TF with Cosine Similarity
  - TF-IDF with Cosine Similarity

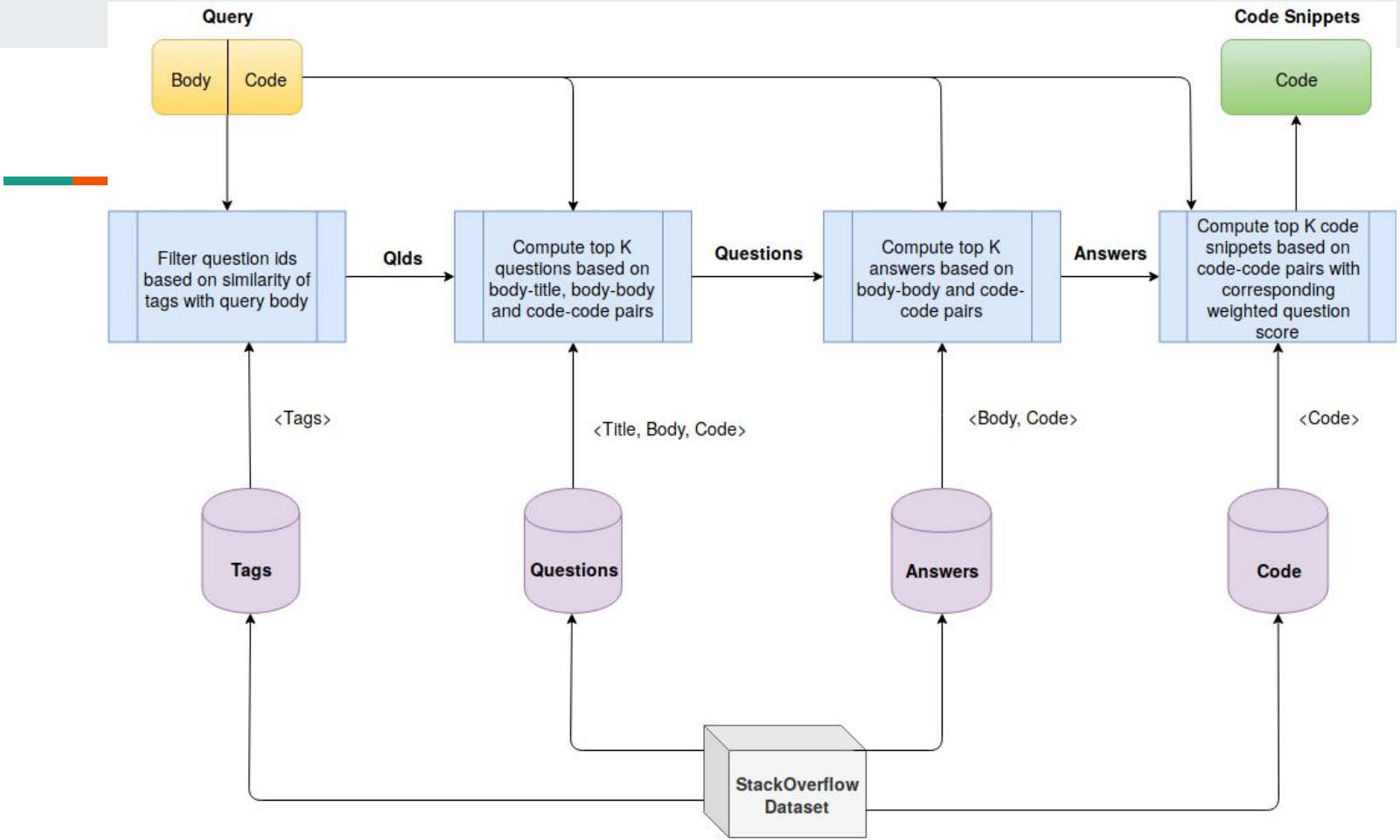
- Code Similarity
  - Line-Line Syntax Matching

Project Workflow

Data Preprocessing	Tag Detection for User Query	Relevant Question Detection	Relevant Answer Detection	Evaluation
1. Performed lemmatization 2. Converted text to the lowercase 3. Removed punctuations like $, <, >, ?, @, `, \ 4. Removed the stop words other than set S where S = {'what', 'which', 'if', 'while', 'for', 'between', 'into', 'from', 'up', 'down', 'in', 'out', 'on', 'off', 'over','then','not','how','do'}	1. Find the relevant tags related to query of user. 2. Done by transforming tagged questions as documents with sections - title, body & code. 3. Finding similarity between these tag documents with user query.	1. Finding the question Ids for relevant tags using query_body - question_title, query_body - question_body, query_code - question_code with weightages.	1. Finding the answer Ids for relevant answers using query_body - answer_body, query_code - answer_code with weightages.	1. Calculated the precision, recall, accuracy (partial and absolute)

Methodology

- Sampling Data based on irrelevant code
- Finding Relevant Tags
- Finding Relevant Questions
- Finding Relevant Answers


Query
Body
Code
Code Snippets
Code
Filter question ids based on similarity of tags with query body
QIds
Compute top K questions based on body-title, body-body and code-code pairs
Questions
Compute top K answers based on body-body and code-code pairs
Answers
Compute top K code snippets based on code-code pairs with corresponding weighted question score
<Tags>
<Title, Body, Code>
<Body, Code>
<Code>
Tags
Questions
Answers
Code
StackOverflow Dataset

Similarity Measure

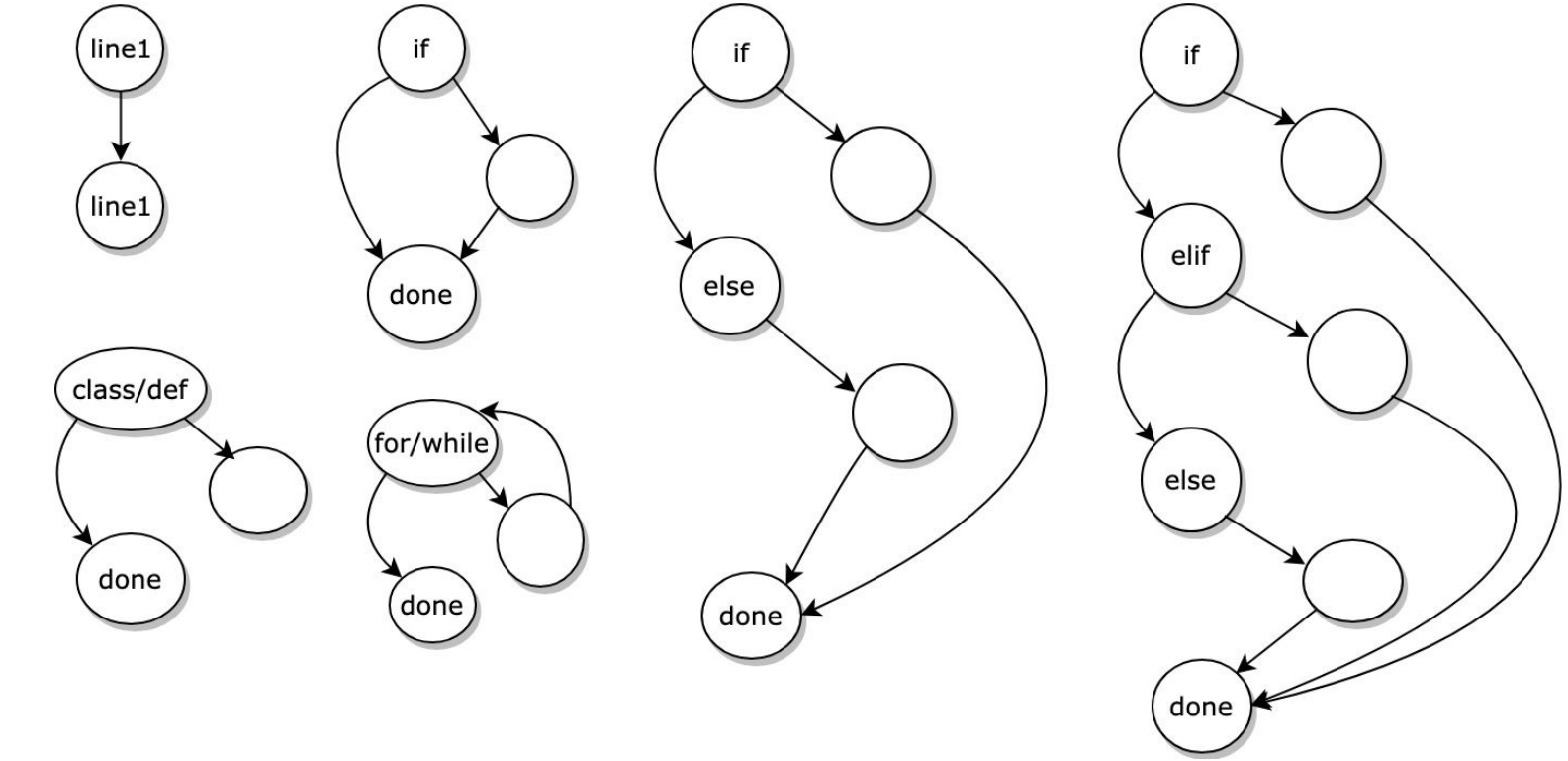
Computing Text Similarity

- TF-IDF with Cosine Similarity

Computing Code Similarity

- Control Flow Graph
- Sliding Window Syntax-Matching Heuristic

Control Flow Graphs



Code Similarity - Control Flow Graph

- Isomorphism
- Eigenvalues Similarity
- Steady State Similarity
- Neighbour Similarity

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - \tilde{y}_i)^2$$

$$x_{ij}^{k+1} \leftarrow \frac{s_{in}^{k+1}(i,j) + s_{out}^{k+1}(i,j)}{2}.$$

Query Results 1

.....User Query.....
Enter the Question Body : how to install numpy in linux
Enter the Question Codepip install --upgrade numpy error

Answer 1
base on path in error message, look like intend instal numpi for python 2.7 alreadi installed. get python 2.6 instead. suggest problem environ variable. adjust variabl in exec ut for python 2.7 appear system version in .

Code
PATH
****
**

Answer 2
alreadi mention in comments, if intent use 2d-array, creat as: then access element like:

Code
m = array([[0, 64], [0, 79], [0, 165], [0, 50]])

****
**

Query Results 2

.....User Query.....
Enter the Question Body : how to convert numpy into panda pandas
Enter the Question Codedata = array([['','Col1','Col2'],['Row1',1,2],['Row2',3,4]])\n df = pd.DataFrame(data,index=data[:,0]),

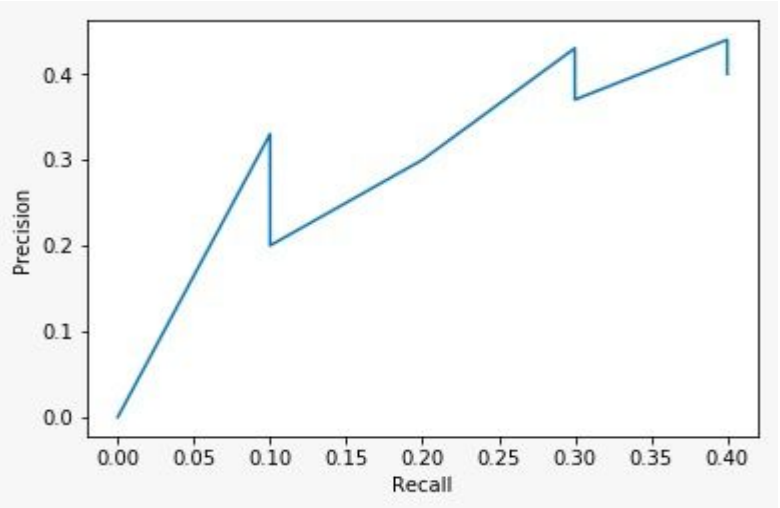
Answer 1
convert panda datafram numpi array:

Code
import numpy as np
np.array(dataFrame)

****

Results

Metric	Score
Precision @ 3	0.33
Precision @ 5	0.20
Recall @ 3	0.10
Recall @ 5	0.10
Precision	0.28
MAP	0.281



Results calculated for 50 samples.