Graph Similarity Self Join

A report submitted for the final year project completion

Bachelor of Technology
in
Computer Science and Engineering

by
M.JEEVAN KUMAR (N120429)
V.JYOTHI BABU (N120513)

Under the guidance of
Sadu Chiranjeevi

Department of Computer Science and Engineering
Rajiv Gandhi University of Knowledge Technologies, Nuzvid

April 23, 2018

RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIES
(A.P. Government Act 18 of 2008)
RGUKT-NUZVID, Krishna Dist. - 521202
Tele Fax: 08656 – 235557/235150

CERTIFICATE

This is to certify that the project entitled **"Graph Similarity Self Join"** is the record of bonafide work carried out by **M. JEEVAN KUMAR (N120429) and V.JYOTHI BABU (N120513)** under my guidance and supervision for the partial fulfillment for the degree of **Bachelor of Technology** in Computer Science and Engineering during the acadamic session Augest-2017 to April-2018 at RGUKT-Nuzvid. To the best of my knowledge, the results embodied in this dissertation work have not been submitted to any university or the institute for the award of any degree or diploma.

Mr.S.Chiranjeevi
Project Supervisor,
Asistant Professor,Dept. of CSE,
RGUKT IIIT, NUZVID.

Examiner
Project Examiner,
Lecturer,Dept. of CSE,
RGUKT IIIT, NUZVID.

RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIES
(A.P. Government Act 18 of 2008)
RGUKT-NUZVID, Krishna Dist. - 521202
Tele Fax: 08656 – 235557/235150

CERTIFICATE OF EXAMINATION

This is to certify that the project report entitled **"Graph Similarity Self Join"** is a record of bonafied work carried out by **M.JEEVAN KUMAR (N120429)** and **V.JYOTHI BABU(N120513)** under my guidance and supervision for the partial fulfillment for the degree of **Bachelor of Technology** in the computer science and engineering during the academic session August-2017 to April-2018 at RGUKT Nuzvid. To the best of my knowledge,the results embodied in the dissertation work have not been submitted to any university or institute for the award of any degree or diploma.

EXAMINER:

1.

2.

3.

DATE:

RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIES
(A.P. Government Act 18 of 2008)
RGUKT-NUZVID, Krishna Dist. - 521202
Tele Fax: 08656 – 235557/235150

CERTIFICATE OF PROJECT COMPLETION

This is to certify that the project report entitled **"Graph Similarity Self Join"** is a record of bonified work carried out by **M.JEEVAN KUMAR(N120429)** and **V.JYOTHI BABU (N120513)** under my guidance and supervision for the partial fulfillment for the degree of **Bachelor of Technology** in the computer science and engineering during the academic session August-2017 to April-2018 at RGUKT Nuzvid. To the best of my knowledge,the results embodied in the dissertation work have not been submitted to any university or institute for the award of any degree or diploma.

MR.S.Chiranjeevi,
Asst Professor,
Project Guide,
RGUKT-Nuzvid.

MR.B.Upendar Rao
Asst Professor,
Head of Department,
RGUKT-Nuzvid.

RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIES
(A.P. Government Act 18 of 2008)
RGUKT-NUZVID, Krishna Dist. - 521202
Tele Fax: 08656 – 235557/235150

DECLARATION

We certify that

(A) The work contained in the project report is original and has been done by ourselves under the general supervision of our supervisor

(B) The work has not been submitted to any other institute for any degree or diploma.

(C) We have followed the guidelines provided by institute in writing the report.

(D) We have confirmed to the norms and guidelines given in the ethical code of conduct of the institute.

(E) Whenever we have used materials from other sources,we have given due credit to them by citing them in the text of the project and giving their details in the references.

M.JEEVAN KUMAR (N120429)
V.JYOTHI BABU (N120513)

RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIES
(A.P. Government Act 18 of 2008)
RGUKT-NUZVID, Krishna Dist. - 521202
Tele Fax: 08656 – 235557/235150

ACKNOWLEDGEMENT

This is to acknowledge our immense pleasure to place on record our profound feeling of reverence and gratitude to **Mr.S.Chiranjeevi** for his inspiration,guidance and excellent support provided during the course of our project work in spite of his busy schedule, he spared their valuble time in the preparation of project report. We extremely indebted to their king encouragement,help and perennial approachability during our project work. It is our gratitude to our Director **Dr. V Venkata Dasu** for the best facilities and faculty for the compilation of the project.

We would like to take this opportunity to express our profound sense of gratitude to our beloved **Mr.B.Upendar Rao**,Head of the Department of CSE for providing required facilities.

We are very much thankful to the **Faculty** for providing direct and indirect support and special thanks to our parents for their moral support throughout the project.

M.JEEVAN KUMAR (N120429)
V.JYOTHI BABU (N120513)

Contents

1 Introduction **9**
- 1.1 Graph Similarity . . . 9
- 1.2 Isomorphic Graphs . . . 9
- 1.3 Sub Graphs Isomorphism . . . 10
- 1.4 Computationally Efficient Graph Similarity Measures . . . 11
  - 1.4.1 Vertex/Edge Overlap . . . 11
  - 1.4.2 Vertex Ranking . . . 11
  - 1.4.3 Vertex/Edge Vector Similarity . . . 11
  - 1.4.4 Sequence Similarity . . . 11
  - 1.4.5 Signature Similarity . . . 11
- 1.5 Applications . . . 12

2 Related Works **14**
- 2.1 GsimJoin . . . 14

3 Preliminaries **15**
- 3.1 Page Rank . . . 15
- 3.2 Sequence . . . 16
- 3.3 Inverted Index . . . 17

4 Proposed Approach **18**
- 4.1 Abstract View . . . 18
- 4.2 Algorithm . . . 19
- 4.3 Example . . . 19

5 Experimental Evaluation **24**
- 5.1 Datasets . . . 24
- 5.2 Results . . . 24

6 Future Scope **25**

7 Conclusion **25**

8 References **25**

List of Figures

1 Similar Graphs Example 9
2 Isomorphic Graphs Example 10
3 Sub Isomorphic Graphs Example 10
4 Similar news articles 12
5 Drugs with similar chemical structure 13
6 Edit Distance Example 14
7 Page Rank 15
8 Sequence Generation 16
9 Inverted Index 17
10 Abstract View 18
11 Graph Data Set Example 19
12 After Page Rank 20
13 Generated Sequences 20
14 Inverted Index 21
15 Similar Graphs having Similarity greater than threshold(0.2) . . 21
16 Feature Generation Example 22
17 Union Shingles Table 22
18 Feature Generation Example 23
19 Graph Data set 24
20 Results 24

Abstract

Graph theory has turned out be a vast area with innumerable applications. Web graphs play an important role in monitoring the evolution of web and to compute the global properties like Page Rank. Graph similarity helps in comparing the web pages when a search engine acquire content from web. Sequence similarity is one of the similarity measure for graphs. It uses a sequence comparison scheme, shingling in particular to compare two graphs.In this project we addressed the graph similarity join problem by using sequence similarity technique. We designed the methods for finding the similar pair of graphs among the given set of graphs which have the similarity value greater than the given threshold.

1 Introduction

Graphs are widely used and they have wide range of applications.Graphs have been utilized to represent complex information in social networks,chemical information systems etc... due to the existence of inconsistent and noisy data graph similarity problem has gained importance so that graph similarity solves the problem of searching for graphs that are similar in nature.

1.1 Graph Similarity

Graph similarity involves determining degree of similarity between two graphs(which is the number between 0 and 1). intuitively, a node in one graph similar to a node in other graph if their neighbourhoods are similar. again it's neighbours are similar if their neighbourhoods are similar and so on.

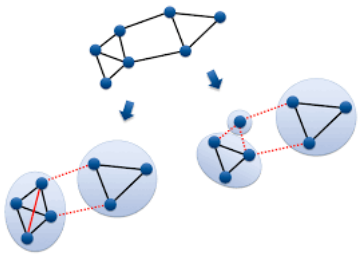

Figure 1: Similar Graphs Example

1.2 Isomorphic Graphs

Definition 1: Given two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, an **isomorphism** between them is any bijection $\phi : V_1 \rightarrow V_2$ such that $(k, l) \in E_1 \longleftrightarrow (\phi(k), \phi(l)) \in E_2, \forall k, l \in V_1$ i.e., there is one-to-one correspondence between

each node of the first graph G_1 and each node of the second graph G_2 that preserves adjacency. Two graphs are said to be **isomorphic** to each other if there exists an isomorphism between them. This is an **NP-Hard** Problem

Figure 2: Isomorphic Graphs Example

In the above example Fig 2, the two graphs are isomorphic, because they can be arranged to look identical.The isomorphism from the left to the right graph is:
F(A) = E, F(B) = A, F(C) = B, F(D) = C, F(E) = D

1.3 Sub Graphs Isomorphism

Definition 2: Given two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, there is a **sub graph isomorphism** from G_1 to G_2, if there exists a sub graph $S \subset G_2$ such that S and G_1 are isomorphic.
This is an **NP-Complete** Problem

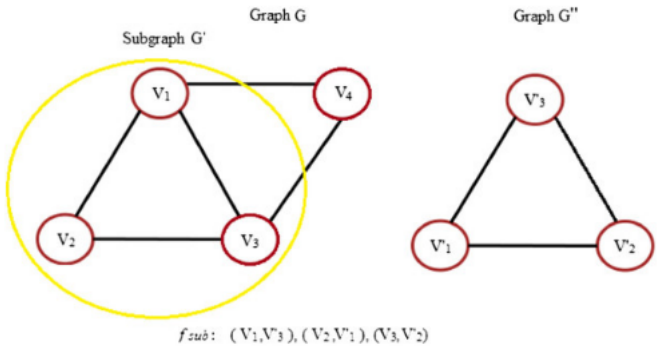

Figure 3: Sub Isomorphic Graphs Example

1.4 Computationally Efficient Graph Similarity Measures

1.4.1 Vertex/Edge Overlap

Two graphs are similar if they share many vertices and/or edges.Two main ways of computing this kind of similarity is graph edit distance and the Jaccard index (as we apply to graphs). Graph edit distance counts the number of some operations on vertices and edges to transform one graph to the other. The operations consist of insertions, deletions, and in the case of labeled graphs, renamings. The Jaccard index is one way to compute the overlap between two graphs. It is defined as the intersections of vertices/edges divided by the union of vertices/edges.

1.4.2 Vertex Ranking

"two graphs are similar if the rankings of their vertices are similar". In this family, the vertices are usually ranked using their quality, and the similarity of rankings is usually computed using a rank correlation method such as Spearman's rho (denoted).

1.4.3 Vertex/Edge Vector Similarity

"two graphs are similar if their node/edge weight vectors are close". For this family, the weights usually represent quality. More formally, if $v_1, v_2, ...$are the vertices shared by two graphs G and $G^{'}$, then we build vectors Q and $Q^{'}$ for these graphs, respectively, where Q[i] is the quality $q(v_i)$ of vertex v_i. Then we compare the two vectors by computing the average difference between all Q[i], $Q^{'}$[i].

1.4.4 Sequence Similarity

"two graphs are similar if they share many sequences of vertices and edges, i.e., short paths" we use a sequence comparison scheme, shingling in particular, to compare two graphs. The main challenge for us is converting the graphs into linear sequences that can then be compared using shingling.

1.4.5 Signature Similarity

"two objects are similar if their signatures are similar". Instead of converting a graph into a sequence, we convert it into a set of features, which are then randomly projected into a smaller-dimensionality feature space (signatures) for comparison.

1.5 Applications

Graphs have a wide range of applications and have been utilized to model complex data in chemical information systems,multimedia,social networks etc.
We can find similar news articles with help of graph similarity, by representing every news article as a graph

Figure 4: Similar news articles

In the above(Fig.4) there are two news articles which are written on two britain politicians. Theresa May and Ed Miliband.
Those two paragraphs articles related to same political situation and both of the articles are representing same political situations. So we can say that aboe two news articles are somewhat similar.

We can find Drugs that have same chemical structure with help of graph similarity by representing every drug as a graph.

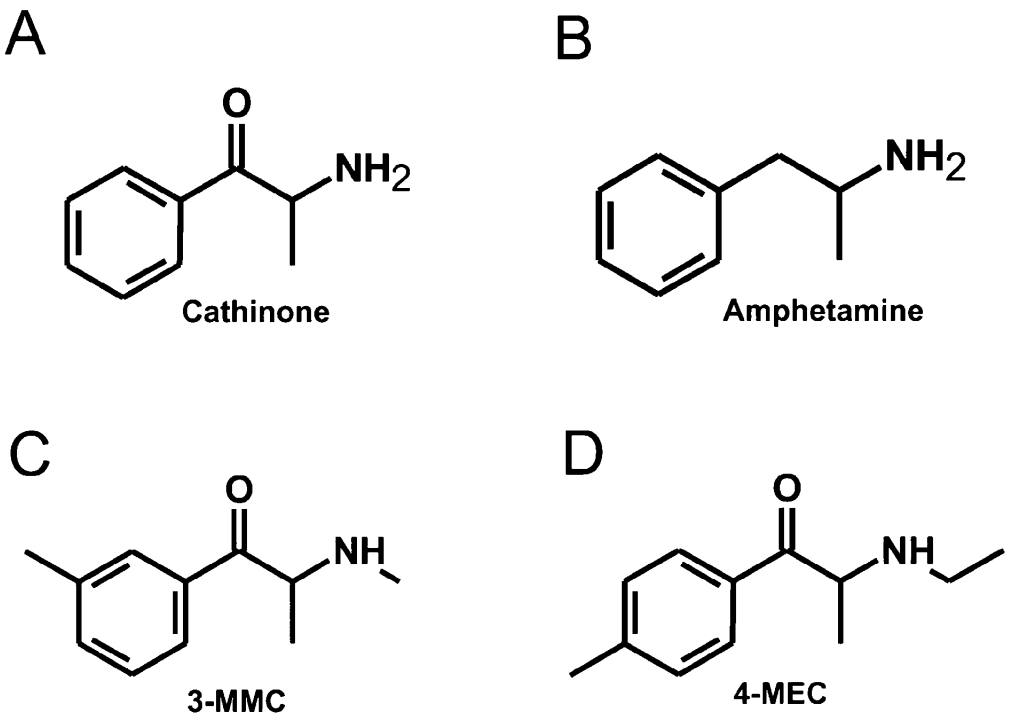

Figure 5: Drugs with similar chemical structure

In the abov figure(5), All the four graphs Cathinone,Amphetamine,3-MMC,4-MEC are having a similar chemical structure. You can clearly observe that from the figure itself.

2 Related Works

2.1 GsimJoin

Efficient Graph Similarity Joins with Edit Distance Construction is a paper combinely published by Xiang Zhao,Chuan Xiao,Xuenin Lin,Wei Wang. In this paper they have an approach to study the graph similarity problem that returns the pair of graphs such that their edit distances are no larger than a given threshold.

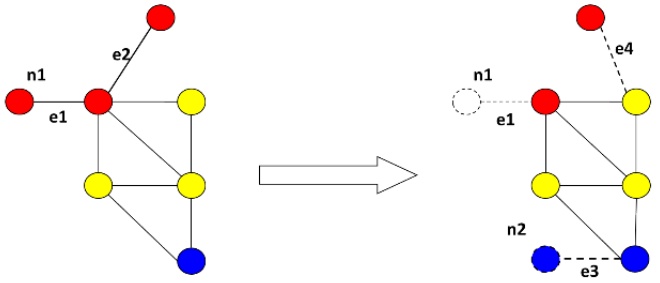

Figure 6: Edit Distance Example

In the above fig(6),First graph is converted to second graph by doing the following
Edit Distatce operations.
On Vertices:
vertex n1 is deleted
vertex n2 is added
On Edges:
edge-e1 is removed
edge-e2 is removed
edge-e3 is added
edge-e4 is added
Number of edit distance operationns = 2(On Vertices) + 4(On Edges).

3 Preliminaries

3.1 Page Rank

Page rank is an algorithm used by Google Search to rank websites in their search engine results. Page Rank is a way of measuring the importance of website pages.
Page Rank works by counting the number of links and quality of links to a page.

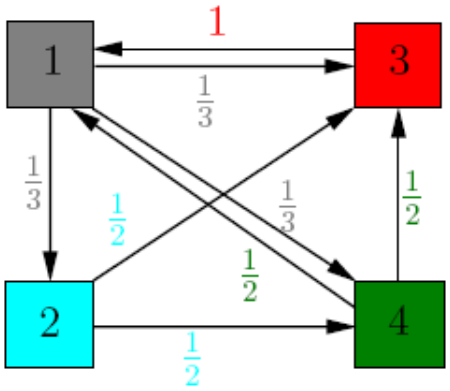

Figure 7: Page Rank

In the above fig(7),initially all the vertices have "1" as their page rank value, The page rank value vertex is devided by the number of outgoing edges and shared equally.
We can clearly observe that from the above figure, Initially page rank value is "1"for vertex "1". It is Divided onto "3" outgoing edges.In the same way all the vertices distribute their page rank.

3.2 Sequence

step-1: Start off by visiting the vertex associated with the highest quality Repeat the following process
step-2: Among the unvisited neighbours for current vertex visit highest quality neighbour
step-3: If the current vertex does not have any unvisited neighbours Jump to nearest the nearest vertex that has the highest quality among all unvisited vertices

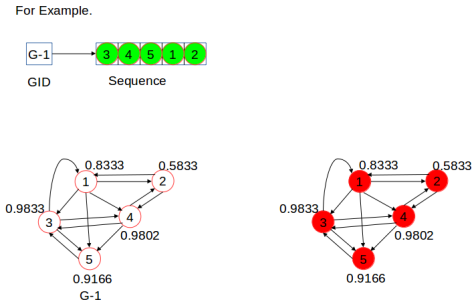

Figure 8: Sequence Generation

In the above figure(8), There is graph "$G1$"with 5 vertices and their computed page ranks.while applying sequence generation algorithm initially highest page rank valued vertex is selected.and it is added to sequence. After that we have to find the highest page rank valued adjacent vertex to "3"(if any). Here we have 1, 4 and 5 among these three, "4" is having highest page rank velue. It is added to the sequence.
Now we have to find highest page rank valued adjucent vertex to node "4". They are "2" and "5" among "2" and "5". "5" is having highest page rank value,So we add "5" to the sequence. Now we have to find highest page rank valued adjacent vertex to vertex "5", but there are not such any vertices. So now we have to find the highest page rank valued unvisited vartex.Here only "1" having highest page rank value.
So we add it to the sequence,and we will find highest page rank valued vertex which is adjacent to vertex "1". Here "2" is the only one remaining and we add it to the sequence. We stop here as all the vertices are visited.

3.3 Inverted Index

An inverted index consists of a list of all the unique words that appear in any document, and for each word, a list of the documents in which it appears. In simple words, it is a hashmap like data structure that directs you from a word to a document or a web page.

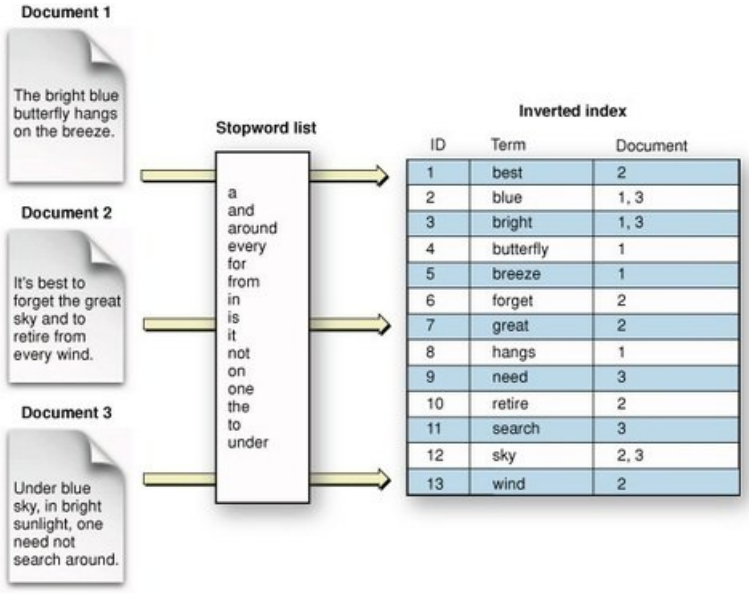

Figure 9: Inverted Index

In the above figure(9), There are 3 documents each having some words. From all the documents a list of union of sop words is created and it's named as "stop word list". The remaining distinct union words are used to find inverted index. For example word "best" is located in document 2, in the same way for all the words a posting list is created.

4 Proposed Approach

4.1 Abstract View

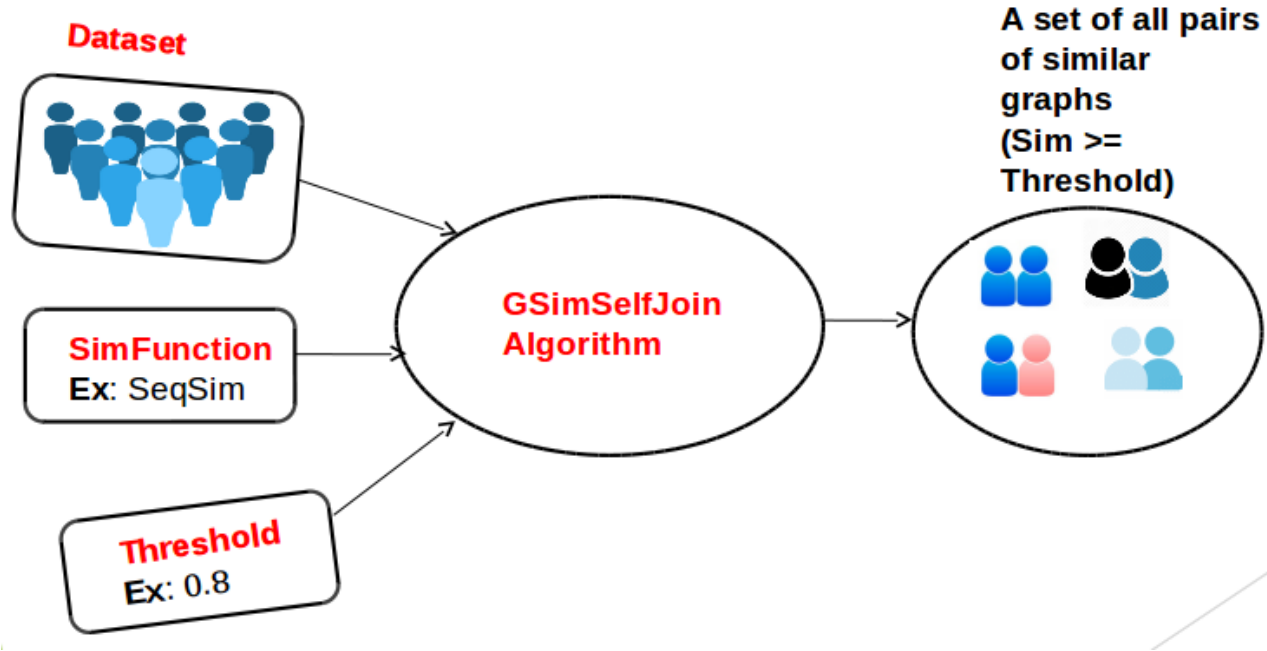

Figure 10: Abstract View

In the above figure(10), Our proposed approach is graphically shown. In our proposed approach a data set of graphs, a similarity measure and the similarity threshold will be given as input to our algorithm it produces(output) all similar pair of graphs having similarity value greater than or equal to given threshold.

4.2 Algorithm

step-1: Apply the PageRank on all graphs

step-2: generate the linear sequence for every graph that can be compared by using shingling

step-3: Construct inverted index (for efficiency)

step-4:Find the similarity among all the graphs

step-5: Returns the pairs of such that their similarity value greater than the given threshold.

4.3 Example

Let's take a data set containing four groups for simplicity

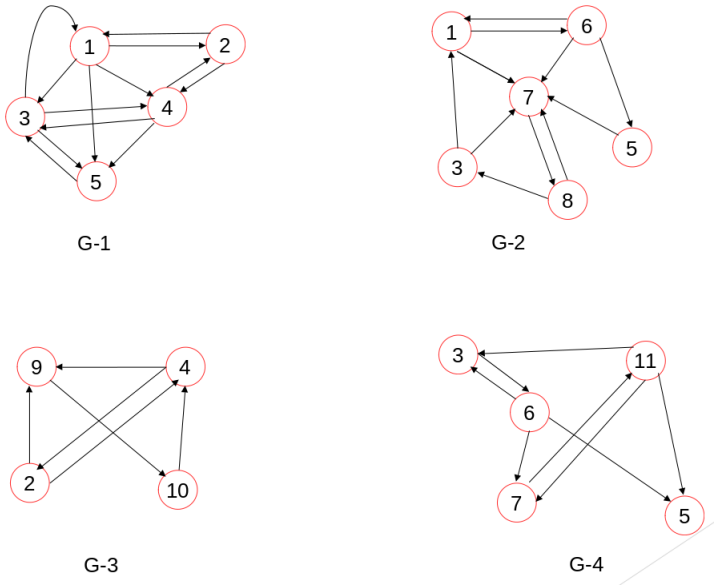

Figure 11: Graph Data Set Example

step-1: Apply page rank on all graphs

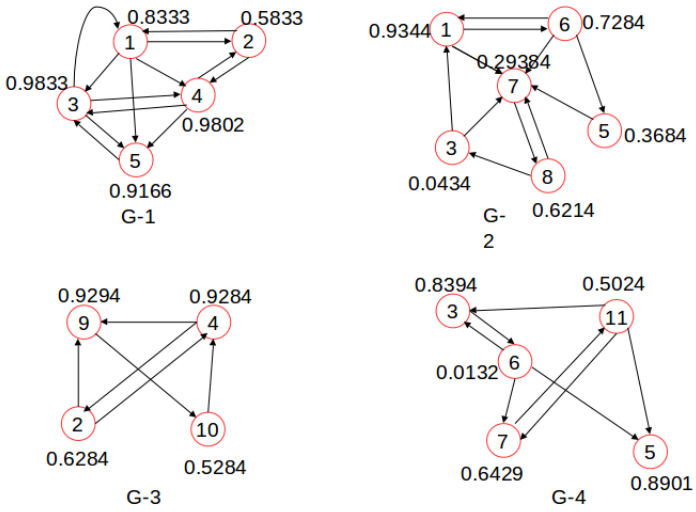

Figure 12: After Page Rank

step-2: generate the linear sequence for every graph that can be compared by using shingling

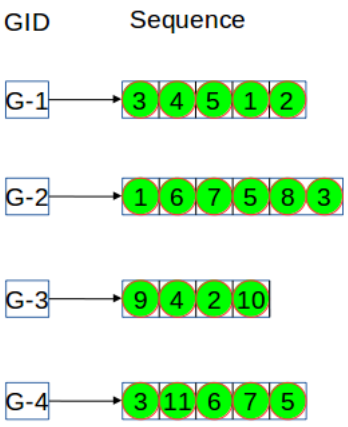

Figure 13: Generated Sequences

step-3: Construct inverted index (for efficiency)

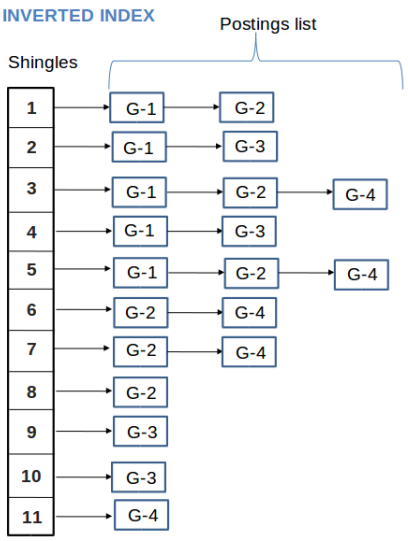

Figure 14: Inverted Index

step-4:Find the similarity among all the graphs

step-5: Returns the pairs of such that their similarity value greater than the given threshold(assume 0.2).

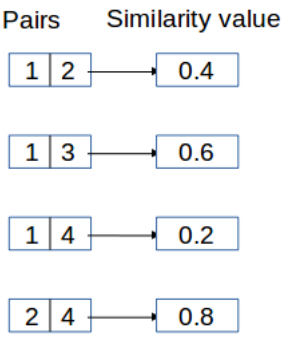

Figure 15: Similar Graphs having Similarity greater than threshold(0.2)

In the this example we have clearly mentioned that how our proposed approach is going to work. Step by step by mentioning what happens after every step. After constructing inverted index we can find all the similar graphs having their similarity value greater than 0(as you we are eliminating all the pairs of graphs which doesn't have any similarity by using inverted index technique). After finding the pairs (by using inverted index) which need to be compared to find the similarity between them, we compare every pair of graphs by comparing their sequences (generated by the sequence similarity algorithm). We compare the sequences of both the graphs by comparing their signatures. Signatures are generated by using a technique called MinHashing. A signature is a collection of features. Finally similarity is calculated by finding the number of common features divided by total number of features.

To understand the feature generation lets take a simple examples with 2 graphs as shown below

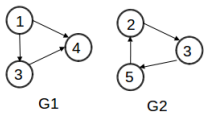

Figure 16: Feature Generation Example

After constructing union shingles for the above two graphs it will look like below table. Graph-1 column represent boolean shingles vector for Graph1, Graph-2 column represent boolean shingles vector for Graph2. Where 0 represent corresponding shingle is present in that graph and 1 represent that corresponding shingle is not present in that graph.

Shingle	Graph-1	Graph-2
1	1	0
2	0	1
3	1	1
4	1	0
5	0	1

Figure 17: Union Shingles Table

For generating the features we need hash functions. Number of features depends on number of hash functions. For simplicity lets take two hash functions

H(x) = x mod 5
G(x) = (2x+1) mod 5

Features are generated by using MinHashing technique, in this technique initially infinity is considered as the feature value for all the features. And feature value will be minimized after computing hash value for every shingle (if that shingle is present in that graph) for that corresponding graph. Below picture clearly depicts feature generation for the above example.

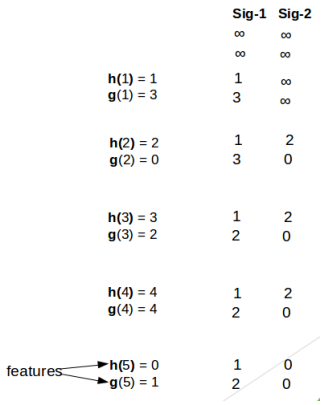

Figure 18: Feature Generation Example

For the above example:
features for Graph1 are (1,2)
features for Graph2 are (0,0)
Number of common features is 0. Total number of features is 2.
similarity 0/2 = 0.

5 Experimental Evaluation

5.1 Datasets

Let's take a data set with four graphs for simplicity as shown below

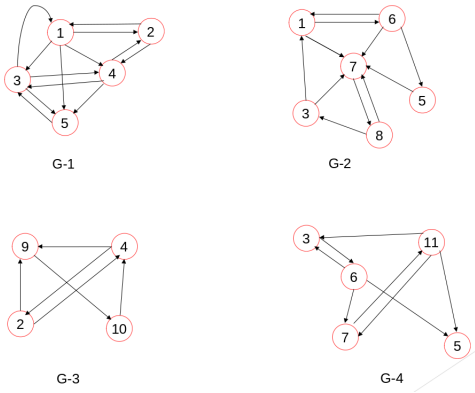

Figure 19: Graph Data set

5.2 Results

For the above dataset after finding the graph similarity self join number of shingles and the similar pairs(having the similarity value greater than given threshold).

```
All Shingles:1
2
3
4
5
6
7
8
9
10
11
DEBUG:hashCoeffs1804289384 846930887
DEBUG:hashCoeffs1681692778 1714636916
DEBUG:hashCoeffs1957747794 424238336
DEBUG:hashCoeffs719885387 1649760493
DEBUG:hashCoeffs596516650 1189641422
#Shingles: 11
1 2 0.400
1 3 0.600
1 4 0.200
2 4 0.800
```

Figure 20: Results

6 Future Scope

In this project we have not used inverted index during the initial development phase.To improve the efficiency we implemented methods for using inverted index to find similar graphs.Inverted index has helped to reduce the number of comparisons.Inverted index is very useful as in real life almost all the graphs are sparse.There is huge scope to decrease the number of comparisons even now by applying some other information retrieval like TF-IDF etc.

7 Conclusion

In this project we have implemented the methods to address the problem of graph similarity self join by using a computationally efficient similarity measure called "sequence similarity". we designed the methods for finding the sequences(based on sequence similarity technique) by applying Page Rank algorithm on all the graphs in the given Data set.To improve the efficiency and to reduce the number of comparisons we designed the methods to implement inverted index for all the graphs in the given data set,it helped in reducing the number of comparisons significantly.

8 References

1. Xiang Zhao,Chuan Xiao,Xuenin Lin,Wei Wang.Efficient Graph Similarity Joins with Edit Distance Constraints in ICDE 2012.
2. Panagiotis Papadimitriou,Ali Dasdan.Web Graph Similarity for Anomaly Detection in 2009.
3. Introduction to information retrieval by Christopher D.Manning , Prabhakar Raghavan,Hinrich Schutze.
4.wwW.GOOGLE.com