DATA WAREHOUSING & DATA MINING
MAJOR PROJECT

PROJECT-11

Analysing and comparing different similarity measures

MENTOR:
Saksham Singhal

GUIDE:
Dr. Vikram Pudi

TEAM 29

ABHISHEK MUNGOLI	FARAZ ALAM	GUNJIT BANSAL	HASEEB AHMED
201405577	201405591	201405568	201405626

Analysing and Comparing Different Similarity

**Problem Statement - ** To find similarity among research papers belonging to Computer Science domain.

Dataset Used - DBLP Dataset

DBLP Dataset provides a comprehensive list of research papers in computer science domain.

Dataset statistics:

- Nodes (Number of Research papers) = 1632442 (1.6 million)
- Roughly 5000 different communities (Topics)

Algorithm: SimRank

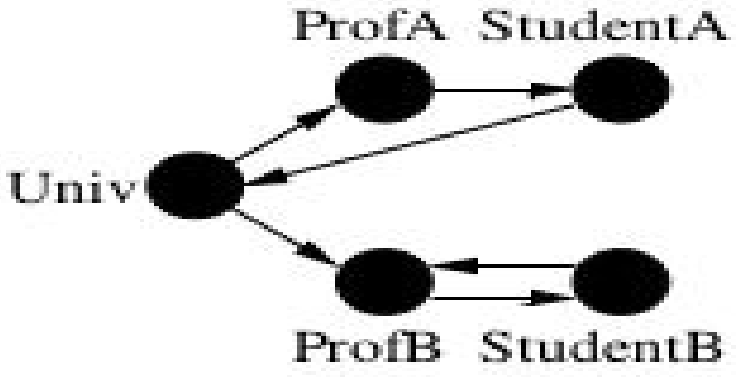
Intro to SimRank -

The generic concept of similarity is **Structural-Context Similarity**, which basically means “Two objects are similar if they are related to similar object.”

Not Domain specific, it basically exploits **object-to-object relationships** and can be applied to almost every domain of interest.

On the **web**, two pages are related if there are **hyperlinks** between them. This statement gives some intuition of its **similarity** with **Google’s PageRank** algorithm.

Illustration with Example



- Prof A and Prof B are similar because they are both referenced by same university.
- Every node is similar to itself & hence university is completely similar to itself.
- Student A & Student B are similar because they are referenced by similar nodes {Prof A, Prof B}.

Similarity with Pagerank

The original PageRank algorithm was described by Lawrence Page and Sergey Brin in their publications. It is given by

$$PR(A) = (1-d) + d\ (PR(T1)/C(T1) + \ldots + PR(Tn)/C(Tn))$$

PR(A) -> PageRank of page A,

PR(Ti) -> PageRank of pages Ti which link to page A,

C(Ti) -> number of outbound links on page Ti and

d -> damping factor which can be set between 0 and 1.

Conclusion - Thus, if an important page ‘x’ refers to page ‘y’, then y is important but if it refers to many other pages as well(apart of y), then y is not that important as other pages are also equally important too.

Basic SimRank Equation

Let's denote the similarity between objects a and b by s(a,b) ∈[0, 1]. Following our earlier motivation, we write a recursive equation for s(a,b). If a = b then s (a, b) is defined to be 1. Otherwise,

$$s(a,b) = \frac{C}{|I(a)||I(b)|} \sum_{i=1}^{|I(a)|} \sum_{j=1}^{|I(b)|} s(I_i(a), I_j(b))$$

where C is a constant between 0 and 1. A slight technicality here is that either a or b may not have any in-neighbors. Since we have no way to infer any similarity between a and b in this case, we should set s(a, b) = 0, so we define the summation in above equation to be 0 when I(a) = ∅ or I(b) = ∅.

Our Approach With Modification

1. Since the domain is big, **1.6 million nodes** (Research paper) we need to follow some **Pruning technique**.
2. One **Pruning technique** used is to set the **similarity of nodes far apart to be 0**, and consider **node-pairs only for nodes which are near** each other.
3. **Node-pairs** were considered only **within a radius ‘r’** from each other.
4. Instead of selecting all **in-neighbours** only **‘f’ in-neighbours were selected randomly.**
5. Some **qualitative and quantitative analysis** was done to **get approximately top 20 similar papers in the dataset.**

Implementation

Given two page id, our implementation gives the similarity between the two provided Research papers.

Some qualitative and quantitative analysis was done to get approximately top 20 similar papers in the dataset.

Results -List of top 20 similar papers in the dataset

	Paper Title	Paper_Id
1.	On the Use of Optimistic Methods for Concurrency Control in Distributed Databases. The Performance of Concurrency Control Algorithms for Database Management Systems.	(642113, 62937)
2.	A Proposal for Distributed Concurrency Control for Partially Redundant Distributed Data Base Systems. Parallelism and Recovery in Database Systems.	(62917, 1118236)
3.	A Proposal for Distributed Concurrency Control for Partially Redundant Distributed Data Base Systems. Distributed Concurrency Control in Database Systems.	(62917 , 641988)

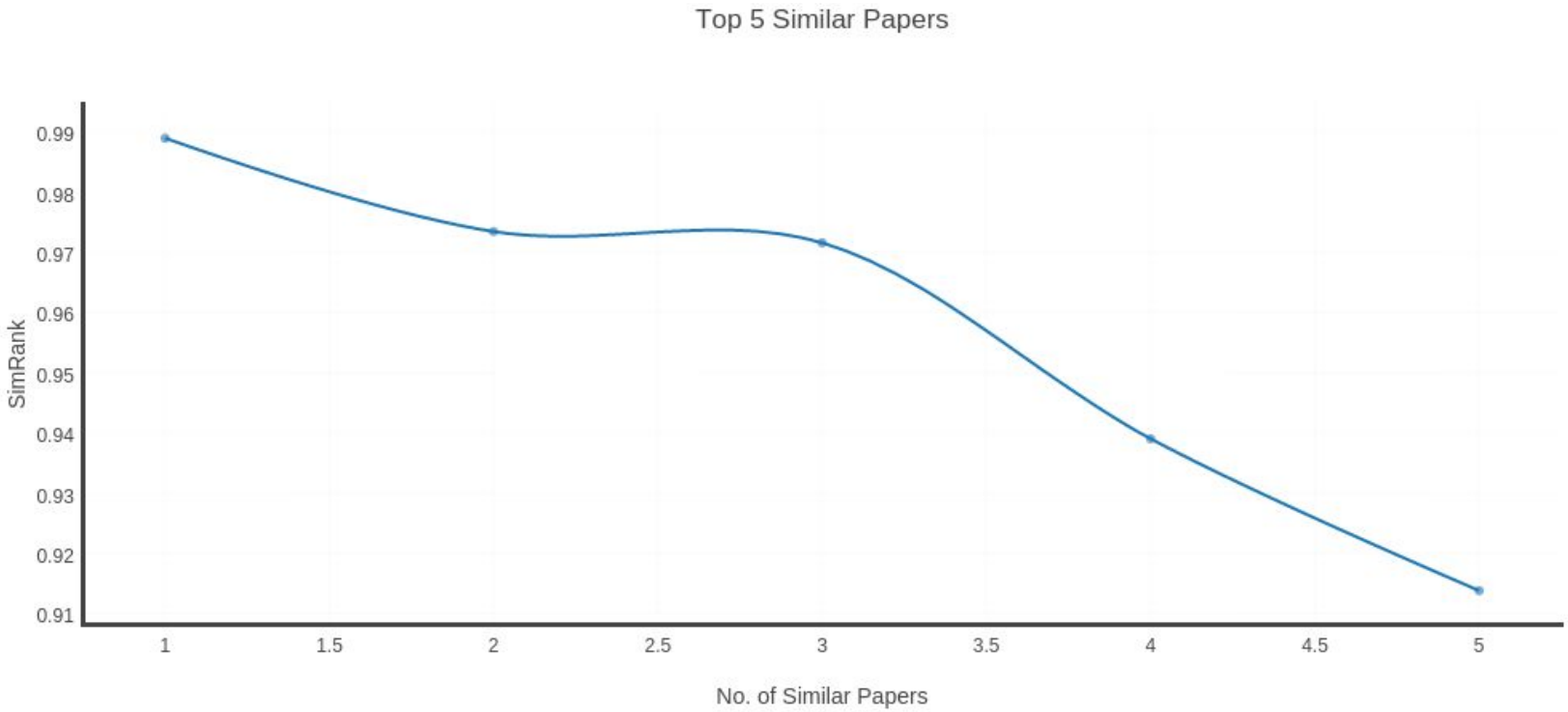
4.	Concurrency Control Overhead or Closer Look at Blocking vs. Nonblocking Concurrency Control Mechanisms. The Vulnerability of Voting Mechanisms	(62915 , 616868)
5.	The R*-Tree: An Efficient and Robust Access Method for Points and Rectangles. Efficient Processing of Spatial Joins Using R-Trees	(598201, 598136)
6.	The Object Database Standard: ODMG-93 (Release 1.1) An Overview of the EXODUS Project.	(2300, 844338)
7.	Architecture of Active Database Systems Datenbanksysteme in Büro, Technik und Wissenschaft (BTW), GI-Fachtagung, Freiburg, 1.-3. März 1999, Proceedings	(2973, 1513215)

8.	Efficient Processing of Spatial Joins Using R-Trees. Multi-Step Processing of Spatial Joins.	(598201, 598202)
9.	Spill Code Minimization Techniques for Optimizing Compilers. Does APL Really Need Run-time Checking?	(1073482, 542431)
10.	Coloring Heuristics for Register Allocation. Rematerialization	(542365 ,542369)
11.	Integrating Register Allocation and Instruction Scheduling for RISCs. Rematerialization	(53613, 542369)

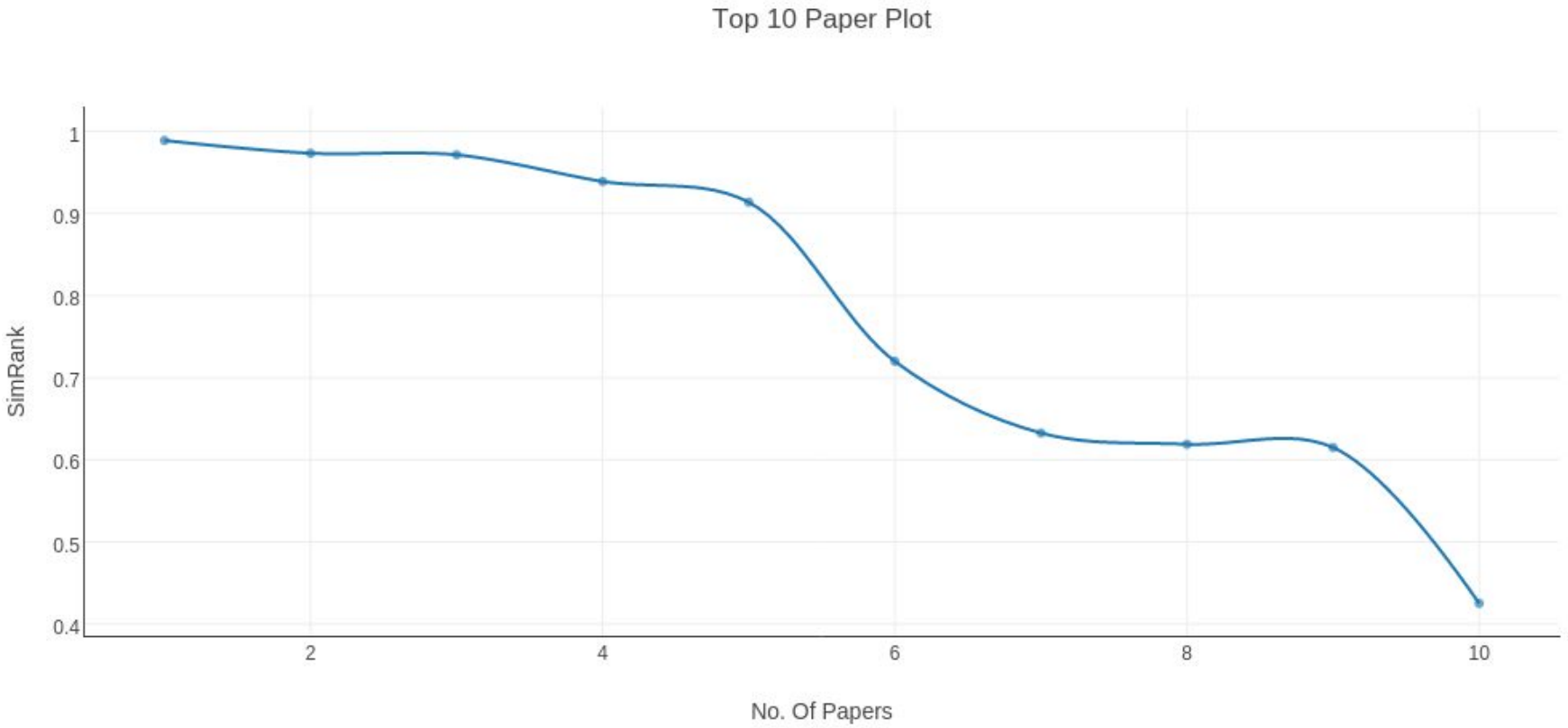
12.	Detecting Equality of Variables in Programs. The History of Language Processor Technology in IBM.	(890179, 545938)
13.	An Indexing Technique for Object-Oriented Databases. Indexing Techniques for Queries on Nested Objects.	(299512, 1112422)
14.	Optimization of Queries using Nested Indices. Indexing Techniques for Queries on Nested Objects.	(176249, 1112422)
15.	A Performance Comparison of Two Architectures for Fast Transaction Processing. An Analysis of Three Transaction Processing Architectures.	(642007, 299464)

16.	Optimization of Queries using Nested Indices. An Indexing Technique for Object-Oriented Databases.	(299512, 176249)
17.	On the Message Complexity of Binary Byzantine Agreement under Crash Failures. Counting Protocols for Reliable End-to-End Transmission.	(841261, 972487)
18.	Composite Registers. Computable Obstructions to Wait-Free Computability.	(841440, 841270)
19.	An algorithm for the asynchronous Write-All problem based on process collision. Spreading Rumors Rapidly Despite an Adversary.	(841416 , 952492)
20.	Efficient Asynchronous Consensus with the Weak Adversary Scheduler. An Optimal Probabilistic Protocol for Synchronous Byzantine Agreement.	(543598, 1059153)

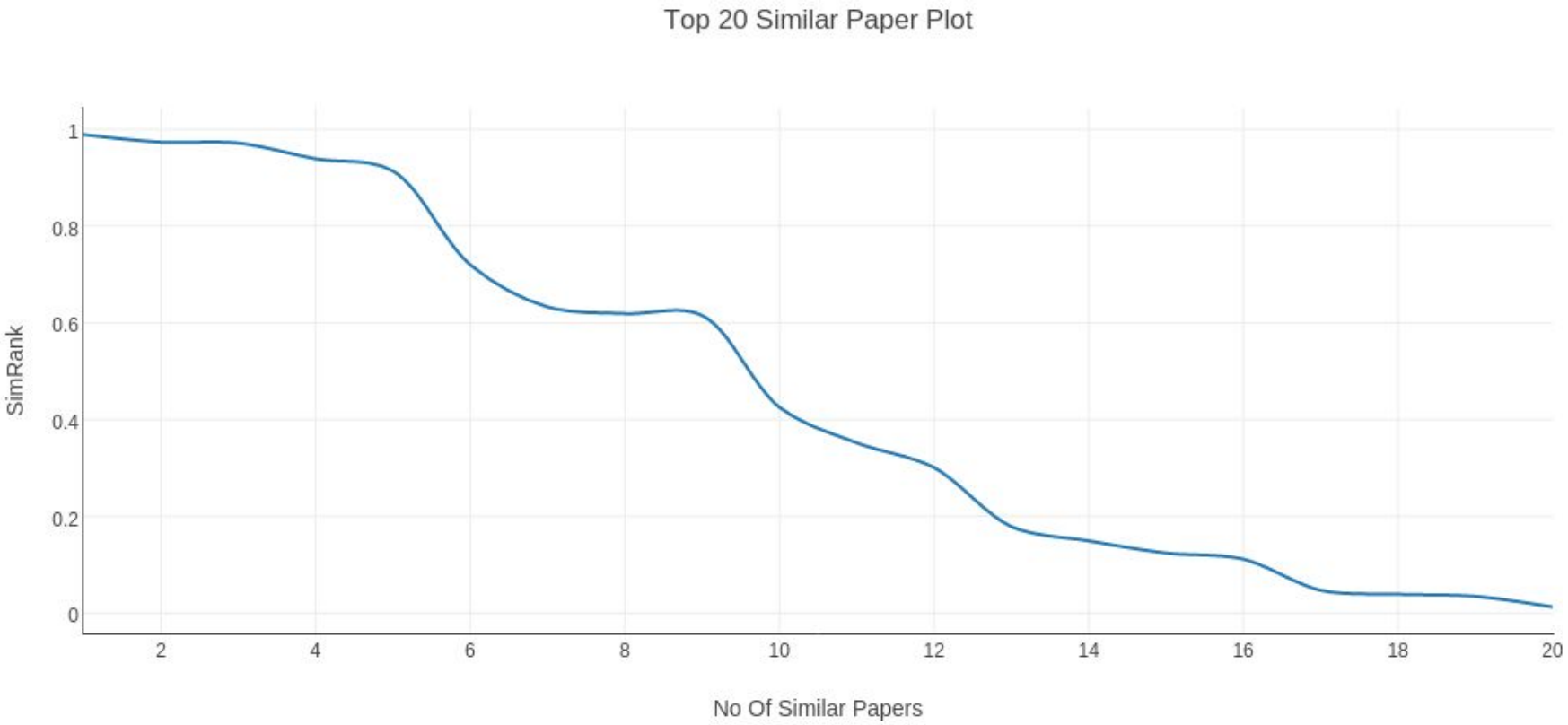
Plot


Top 5 Similar Papers
SimRank
0.99
0.98
0.97
0.96
0.95
0.94
0.93
0.92
0.91
1
1.5
2
2.5
3
3.5
4
4.5
5
No. of Similar Papers

Plot


Top 10 Paper Plot
SimRank
1
0.9
0.8
0.7
0.6
0.5
0.4
2
4
6
8
10
No. Of Papers

Plot


Top 20 Similar Paper Plot
SimRank
1
0.8
0.6
0.4
0.2
0
2
4
6
8
10
12
14
16
18
20
No Of Similar Papers

Conclusion

SimRank exploits **object-to-object relationships very well** and finds out the similarity between two objects.

We implemented a basic version of **SimRank** and also found out the top 20 similar Research papers from the **DBLP dataset** containing comprehensive list of research papers in computer science domain.

SimRank is a generic approach and its **basic idea** can also be applied to **other domain of interests**.