ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 5, May 2017

An Intelligent System for Subgraph Matching with Set Similarity in Large Graph Database

Monali Vitthal Divekar , Prof. Shyam S. Gupta

M. E Student, Dept. of Computer Engineering, Savitribai Phule Pune University, Siddhant College of Engineering, Sudumbare, Maval, Pune, India.

Asst. Professor, Dept. of Computer Engineering, Savitribai Phule Pune University, Siddhant College of Engineering, Sudumbare, Maval, Pune, India.

ABSTRACT: In real lifestyles, interpersonal companies, Semantic internet and Natural techniques, social networks, and biological networks, each vertex extra by and large than now not contains important information, which may also be displayed by way of an arrangement of tokens or components. In this paper, a subgraph matching with set similarity (SMS2) query over a large graph database, which retrieves subgraphs that are structurally isomorphic to the query graph, and during this time period it also satisfy the condition of vertex pair matching with the (dynamic) weighted set similarity. This paper designs a novel lattice-based index for data graph to efficiently process the SMS2 query, and lightweight signatures for both query and data vertices. We not only propose an efficient two-phase pruning strategy based on the index and signatures, which including set similarity pruning and structure-based pruning, and also utilizes the unique features of both (dynamic) weighted set similarity and graph topology, but also propose an efficient dominating-set-based subgraph matching algorithm direct by a dominating set selection algorithm to achieve better query performance. Extensive experiments on both real as well as synthetic datasets demonstrate that our method outperforms state of-the-art methods by order of magnitude.

KEYWORDS: subgraph matching, graph database, pruning strategy, vertex pair matching, weighted set similarity, query processing, SMS2 query, dynamic set similarity, structurally isomorphic subgraph, synthetic datasets.

I. INTRODUCTION

In real-world graphs such as social networks, Semantic Web and biological networks, each vertex usually contains rich information, which can be modeled by a set of tokens or elements. We use a subgraph matching with set similarity (SMS2) query over a large graph database, which retrieves subgraphs that are structurally isomorphic to the query graph, and meanwhile satisfy the condition of vertex pair matching with the weighted set similarity. To develop an efficient search engine using subgraph similarity search on large probabilistic graph databases. Based on the index and signatures, we use an efficient two-phase pruning strategy including Vertical and horizontal pruning, which exploits the unique features of both weighted set similarity and graph topology. We use an efficient dominating set based subgraph matching algorithm guided by a dominating set selection algorithm to achieve better query performance.

II. RELATED WORK

In [1] authors used During the online phase, a set of pruning techniques facilitated by the offline data structures are introduced and integrated together to greatly reduce the search space of SMS2 queries. Author uses Structured similarity and set similarity pruning techniques. Author uses design an efficient algorithm to perform subgraph matching based on the dominating set of query graph approach can effectively and efficiently answer the SMS2 queries in a large graph database. In [2] authors used developing reputation of graph databases has generated interesting information administration issues, such as subgraph search, shortest-path question, reach ability verification, and sample healthy. Amongst these, a pattern suit question is extra flexible compared to a subgraph search and extra

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 5, May 2017

informative compared to a shortest-path or reach-ability query. On this paper, we handle sample fit problems over large information graph G. Chiefly, given a pattern graph (i.e., question Q), Author need to find all matches (in G) which have the identical connections as those in Q. In order to shrink the hunt area enormously, Author first turn out to be the vertices into features in a vector area by way of graph embedding tactics, coverting a pattern in shape query into a distance-founded multi-manner become a member of obstacle over the modified vector space. Author additionally endorses a couple of pruning approaches and join order decision process to approach become a member of processing successfully. In [3] authors, previous paper problem is to find all patterns in a colossal data graph that suit a consumer given graph sample. Author suggests a new two-step R-join (reachability become a member of) algorithm with filter step and fetch step situated on a cluster-established become a member of index with graph codes. Author keep in mind the filter step as an R-semijoin, and suggest a brand new optimization approach through interleaving R-joins with R-semijoins. It carried out huge performance experiences, and confirm effectivity of our proposed new approaches. In [4] authors used a novel indexing method that includes graph structural information in a hybrid index structure. This indexing technique attains high pruning power and the index size scales linearly with the database size. In addition, Author propose an innovative matching paradigm to query large graphs. This technique distinguishes nodes by their importance in the graph structure. The matching algorithm firstly matches the important nodes of a query and then progressively extends these matches. In [5] authors use a Torque seeks an identical set of proteins that are sequence much like the query proteins and spana linked vicinity of the network, whilst permitting both insertions and deletions. The algorithm uses on the other hand dynamic programming and integer linear programming for the quest undertaking. Author experiment Torque with queries from yeast, fly, and human, the place Author compare it to the Q Net topology founded process, and with queries from much less studied species, where best topology-free algorithms practice. Torque detects many extra suits than QNet, whilst in both circumstances giving results which are totally functionally coherent. In [6] authors use a novel approximate graph matching technique called SAGA (Substructure Index based Approximate Graph Alignment). SAGA employs a flexible graph distance model to measure similarities between graphs. To expedite query execution on large databases, a novel index is built on the database graphs. Using SAGA for pathway analysis, Author have been able to identify interesting similarities between distinct pathways that could not be found by previous methods. Furthermore, SAGA is orders of magnitude faster than existing methods. The results produced by SAGA can help life sciences researchers discover conserved function units among different pathways, detect potential path ways involved in or affected by a particular disease, and integrate different path way databases to produce more complete data. In addition to pathway analysis, SAGA can be used to compare biomedical documents. In [7] author uses an excessive efficiency graph indexing mechanism, SPath, to address the graph question problem on giant networks. SPath leverages decomposed shortest paths around vertex local as common indexing units, which prove to be each strong in graph search space pruning and enormously scalable in index construction and deployment. Via SPath, a graph query is processed and optimized past thetraditionalvertex-at-a-timetrendtoaextraefficientpath-at-atimeapproach experimental reports exhibit the effectiveness and scalability of SPath, which proves tobe a more functional and efficient indexing process in addressing graph queries on huge networks. In [8] authors use a novel algorithm that supports efficient subgraph matching for graphs deployed on a distributed reminiscence retailer. Rather of counting on tremendous-linear indices, Author use efficient graph exploration and gigantic parallel computing for query processing. Our experimental results reveal the feasibility of performing subgraph matching on internet-scale graph knowledge. In [9] author explores the notion of similarity headquartered on connectivity alone, and proposes a few algorithms to quantify it. Our metrics take abilities of the neighborhoods of the nodes within the quotation graph. Two versions of hyperlink founded similarity estimation between two nodes are described, one headquartered on the separate neighborhood neighborhoods of the nodes, and yet another situated on the joint nearby multiplied from both nodes at the same time. The algorithms are carried out and evaluated on a subgraph of the quotation graph of computer science in a retrieval context. In [10] DBpedia is a group effort to extract structured expertise from Wikipedia and to make this information available on the web. DBpedia enables you to ask sophisticated queries against datasets derived from Wikipedia and to hyperlink other datasets on the net to Wikipedia data. In [11] given a question string, also represented as a set of tokens, a weighted string similarity question identifies all strings in the database whose similarity to the question is better than a consumer designated threshold. Weighted string similarity queries are priceless in purposes like information cleaning and integration for locating approximate suits within the presence of typographical mistakes, a couple of formatting conventions, information transformation blunders, and so on. It show that this challenge has semantic properties that may be exploited to design index buildings that help very effective algorithms for query answering. In [12], algorithm for graph isomorphism and subgraph isomorphism suited to coping with

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 5, May 2017

enormous graphs is expanded here to curb its spatial complexity and to attain a greater performance on enormous graphs; its aspects are analyzed in element with targeted reference to time and memory necessities. The outcome of a testing carried out on a publicly to be had database of synthetically generated graphs and on graphs relative to an actual utility dealing with technical drawings are awarded, confirming the effectiveness of the technique, especially when working with colossal graphs.

III. IMPLEMENTATION DETAILS

This approach includes offline processing and online processing

Offline processing: This construct a novel inverted pattern lattice to facilitate efficient pruning which is based on the set similarity. Since the dynamic weight of each element makes existing indices in efficient for answering SMS queries, the need is to design a novel index for SMS query. Motivated by the anti-monotone property of the lattice structure, then frequent patterns from element sets of vertices in the data graph G, and organize them into a lattice. Also store data vertices in the inverted list for each frequent pattern P, if P is contained in the element sets of these vertices. The lattice together with the inverted lists is called inverted pattern lattice, which can greatly reduce the cost of dynamic weighted set similarity search. To support structure-based pruning, encode each query vertex and data vertex into a query signature and a data signature respectively by considering both the topology and set information, and hash all the data signatures into signature buckets.

Online processing: This propose of finding a cost-efficient dominating set of the query graph , only search candidates for vertices in the dominating set, different dominating sets will lead to different query performances. Thus, a dominating set selection algorithm to select a cost- efficient dominating set DS(Q) of query graph Q. The dominating-set- based subgraph matching is motivated by two observations: (1) finding candidates in SMS queries are much more expensive than that in typical subgraph search, because set similarity calculation is more costly than vertex label matching. As a result, the filtering cost can be reduced by searching dominating vertices of V (Q) rather than all query vertices. (2)Some query vertices may have a large amount of candidate vertices, which leads to many of the unnecessary intermediate results during subgraph matching. Therefore, the subgraph matching cost can also be reduced by decreasing the size of intermediate results. For each vertex $u \in DS(Q)$, This propose a two phase pruning strategy, including set similarity pruning and structure-based pruning. The set similarity pruning includes anti- monotone pruning, horizontal pruning, and vertical pruning, which are based on inverted pattern lattice . Based on the signature buckets, Also the structure-based pruning technique by utilizing novel vertex signatures. After the pruning, This propose an efficient DS-Match subgraph matching algorithm to obtain subgraph matches of Q based on candidates of dominating vertices in DS(Q). DS-match utilizes topological relations between dominating vertices and non-dominating vertices to reduce the scale of intermediate results during subgraph matching, and therefore reduces the matching cost.

IV.ALGORITHM

We have design algorithms to illustrate keywords similarity measurement methods that are already designed. Algorithms are set to Pseudo code language. The result can be displayed via web browser. The methods are evaluated by three criterions (precision, recall, and F-measure)

Precision can be evaluates by the following formula.

precision = TP/(TP + PP) * 100

TP (True Positive) = keywords presented and accepted by experts

FP (False Positive) = keywords presented but not accepted by experts.

Recall can be evaluates by the following formula.

precision = TP/(TP + FN) * 100

TP (True Positive) = keywords presented and accepted by experts

FN (False Negative) = keywords not presented and not accepted by experts

F-measure can be evaluates by the following formula.

F = (2 * precision * recall) / (precision + recall)

From the keywords similarity measurement with JNVA and JNLA.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 5, May 2017

Algorithm for Measuring String Similarity with JNVA(Jaccards N-Gram Vector Average)

Inputting string1 : str1
Inputting string 2 : str2
    Calculating unique string1 : ustr1
    Calculating unique string2 : ustr2
Jaccard(ustr1,ustr2) :
    Calculating union of ustr1 and ustr2 : unstr
    Calculating intersection of ustr1 and ustr2 : instr
    Calculating result of Jaccard : Jacstr = |unstr|/|instr|
Ngram2(str1,str2) :
    Creating data set 1 from str1 with Ngram : ngm1
    Creating data set 2 from str2 with Ngram : ngm2
    Calculating unique ngm1: ungm 1
    Calculating unique ngm2: ungm 2
    Calculating result of Ngram2 : Ngmstr = Jaccard(ungm1,ungm2)
Vector Space(ustr1,ustr2,str1,str2):
    Creating vector space data from ustr1 and ustr2 : vt
    Creating vector space data 1 from vt and str1 : vtstr1
    Creating vector space data 2 from vt and str2 : vtstr2
    Calculating unit vector 1 : uvtstr1 = vtstr1/Len(str1)
    Calculating unit vector 2 : uvtstr2 = vtstr2/Len(str2)
    Calculating result of Vector Space : Vspstr =uvtstr1*uvtstr2
Calculating result of JNVA : JNVA = (Jacstr + Ngmstr +Vspstr)/3

B. Algorithm for Measuring String Similarity using JNLA(Jaccards N-Gram Length Average)

Inputting string1 : str1
Inputting string 2 : str2
    Calculating unique string1 : ustr1
    Calculating unique string2 : ustr2
Jaccard(ustr1,ustr2) :
    Calculating union of ustr1 and ustr2 : unstr
    Calculating intersection of ustr1 and ustr2 : instr
    Calculating Jaccard : Jacstr = |unstr|/|instr|
Ngram2(str1,str2) :
    Creating dataset 1 from str1 with Ngram : ngm1
    Creating dataset 2 from str2 with Ngram : ngm2
    Calculating unique ngm1: ungm 1
    Calculating unique ngm2: ungm 2
    Calculating result of Ngram2 : Ngmstr =
Jaccard(ungm1,ungm2)
    Length(ustr1,ustr2,str1,str2):
    Calculating differences of str1 and str2 : dfstr =abs(Len(str1)-Len(str2))
    Calculating intersection of ustr1 and ustr2 : instr
    Calculating result of Length Lenstr = EXP(-dfstr/Len(instr))
    Calculating result of JNLA : JNLA = ((Jacstr +Ngmstr + Lenstr)/3

IV.MATHEMACTICAL MODEL

S:System;
A system is de_ne dasaset such that:
S= {U,I,P,O}
Where,
    U: Set of users

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 5, May 2017

I: Set of Input.
O: Set of output.
P: Set of Processes.

INPUT SET DETAILS:
(Input Module)
I= Data Set, Query

PROCESS SET DETAILS:
(Query Processing)
P= Data processing, Query Graph Formation, Graph partition, Data Classification Cluster Formation; Result Recommendation

OUTPUT SET DETAILS:
(Result)
O=Statistic, Recommendation Result, Subgraph Result

IV. RESULTS

Important QA-gMatch problem of existing system, which retrieves those consistently matching subgraphs from in consistent probabilistic datagraphs with the guarantee of high quality scores. To tackle the problem, proposed system specifically design effective pruning methods, adaptive label pruning and quality score pruning, for reducing the search space. Further, proposed system build an effective index to facilitate the QA-gMatch processing. Extensive experiments are conducted to verify the efficiency and effectiveness of our approaches. Here, analysis is done in Offline and online mode. User has to enter the query and correct (lies between 0 to 100) threshold and then click on search button.

After Keyword is searched, the number of related link or URL are saved in database and displayed. Most visited link are ranked and displayed in Query Result with rank. After user clicks on showgraph, the links that contain more relevant data are displayed. We use string similarity algorithm for finding the links that contain more relevant data. After user clicks on any of the link,the sublinks will be displayed. The sublinks are formed on the basis of the keywords and nouns that exist in the content of the link. Here we can see that the no. of links displayed are less than that of before. This is because, as the threshold value decreases, the less amount of data which is more accurate and relevant will be displayed. After user clicks on search videos, the videos that are there on first page of youtube will be displayed. After user clicks on search images, the images that are there on the first page of google(images) will be displayed.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 5, May 2017

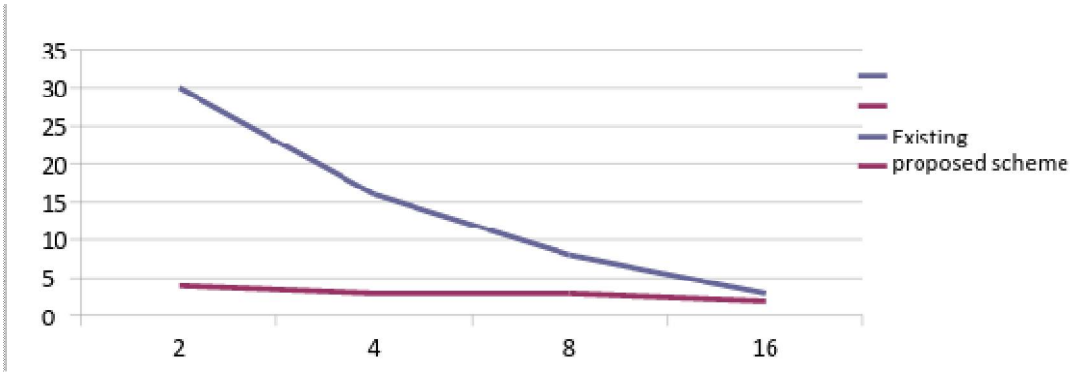
(JNVA and JNLA) can better deliver keywords similarity measurement value prediction and are more stable than Jaccard's, N-Gram, and Vector space.

Time complexity of proposed system and existing system shown here.

Key Set Count	proposed scheme query graph count	Existing System query graph count
2	30	4
4	16	3
8	8	3
16	3	2

Table 1.1

Graph for comparison between exiting and proposed system are shown below.



Graph

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 5, May 2017

V. CONCLUSION

In this paper, we study the problem of subgraph matching with set similarity, which also exists in a very large number of applications. To tackle this problem, we propose efficient pruning techniques by considering both vertex set similarity and graph topology. A novel inverted pattern lattice and structural signature buckets are designed to facilitate the online pruning. Finally, we propose an efficient dominating- set based subgraph match algorithm to find subgraph matches. Extensive experiments have been conducted to demonstrate the efficiency and effectiveness of our approaches compared to state-of-the-art subgraph matching methods.

ACKNOWLEDGEMENT

I dedicate all my works to my esteemed guide, Prof. Shyam Gupta, whose interest and guidance helped me to complete the work successfully. This experience will always steer me to do my work perfectly and professionally. I also extend my gratitude to Prof. Deepak Gupta (H.O.D. Computer Department) who has provided facilities to explore the subject with more enthusiasm. I express my immense pleasure and thankfulness to all the teachers and staff of the Department of Computer Engineering, for their co-operation and support. Last but not the least, I thank all others, and especially my friends who in one way or another helped me in the successful completion of this paper

REFERENCES

1. Liang Hong, Lei Zou, Xiang Lian, Philip S. Yu, "Subgraph Matching with Set Similarity in a Large Graph Database", IEEE Transactions on Knowledge and Data Engineering, (Volume:PP , Issue: 99),12 January 2015
2. L.Zou,L.Chen,and M.T.Ozsu, "Distance-join: Pattern match query in a large graph database" ,PVLDB,vol. 2,no. 1,2009.
3. J.Cheng,J.X.Yu,B.Ding,P.S.Yu,andH.Wang, "Fast graph pattern matching", in Data Engineering, 2008. ICDE 2008. IEEE 24th International Conferenceon. IEEE,2008,pp. 913922.
4. Y. Tian and J. M. Patel, "Tale: A tool for approximate large graph matching", in ICDE,2008.
5. Eftychios A. Pnevmatikakis, Petros Maragos "An Inpainting System For Automatic Image Structure-Texture Restoration With Text Removal", IEEE trans. 978-1-4244-1764, 2008
6. S. Bruckner1, F. Huffner, R. M. Karp, R. Shamir, and R. Sharan, "Torque: topology-free querying of protein interaction networks", Nucleic Acids Research, vol. 37, no. suppl 2, pp. W106–W108, 2009.
7. P. Zhao and J. Han, "On graph query optimization in large networks", PVLDB, vol. 3, no. 1-2, 2010.
8. Z. Sun, H. Wang, H. Wang, B. Shao, and J. Li, "Efficient subgraph matching on billion node graphs", PVLDB, vol. 5, no. 9, 2012.
9. W. Lu, J. Janssen, E. Milios, N. Japkowicz, and Y. Zhang, "Node similarity in the citation graph, Knowledge and Information Systems", vol. 11, no. 1, pp. 105–129, 2006.Mr. Rajesh H. Davda1, Mr. Noor Mohammed, "Text Detection, Removal and Region Filling Using Image Inpainting", International Journal of Futuristic Science Engineering and Technology, vol. 1 Issue 2, ISSN 2320 – 4486, 2013
10. S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, and Z. Ives, "Dbpedia: A nucleus for a web of open data", in ISWC, 2007.
11. M. Hadjieleftheriou and D. Srivastava, "Weighted set-based string similarity", in IEEE Data Engineering Bulletin, 2010.
12. L. P. Cordella, P. Foggia, C. Sansone, and M. Vento, "A (sub)graph isomorphism algorithm for matching large graphs", PAMI, vol. 26, no. 10, 2004