Effective Keyword Search over (Semi)-Structured Big Data

Mehdi Kargar

School of Computer Science
Faculty of Science
University of Windsor
How Big is this Big Data?

40 Billion
Instagram Photos

300 Hours of Video
is uploaded every Minute

4.5 Million Entities
3.1 Billion RDF Triples

1.4 Billion
Facebook Users
The Web is Big Data

- **50 Billion** Web Pages
  - News
  - Blogs
  - Business
- How can everyone *easily access* the web?!
The Web is Accessible because of Search Engines

- Web search engines (Google, Bing) index almost the entire Web
- It gives us a **text box** to type whatever we want
- Each answer is a single web-page
  - **unstructured data**

US president with a PhD

After briefly practicing law in Atlanta, Georgia, he received a Ph.D. in political science from Johns Hopkins University in 1886. **(Wilson remains the only U.S. president to earn a doctorate degree)**

Woodrow Wilson - U.S. Presidents - HISTORY.com
www.history.com/topics/us-presidents/woodrow-wilson
What Web Search Engines can’t Find!

• Keywords: "Keanu Reeves" "Laurence Fishburne" "Nicole Kidman"
  – Let’s search over IMDb dataset
• **Google**: a list of web pages
• **Expectation**: Have they starred in the same movie?

```
"Keanu Reeves" "Laurence Fishburne" "Nicole Kidman" site:imdb.com
```

![Search results from Google](image.png)
Outline

• **Keyword Search in Big Graphs**
  – VLDB’11, ICDE’12, TKDE’14, SIGMOD’14, KAIS’15, ICDE’15, CIKM’16

• **Team Formation in Social Networks**
  – CIKM’11, ICDMW’11, PKDD’12, SDM’13, WI’14, EDBT’17

• **Conclusions**
Outline

• **Keyword Search in Big Graphs**
  – VLDB’11, ICDE’12, TKDE’14, SIGMOD’14, KAIS’15, ICDE’15, CIKM’16

• **Team Formation in Social Networks**
  – CIKM’11, ICDMW’11, PKDD’12, SDM’13, WI’14, EDBT’17

• **Conclusions**
(Semi-)Structured vs Unstructured Data

• Structured and semi-structured data has high degree of organization
  – Relational Databases & Social Networks
  – Usually modeled as graphs
• Each answer to a query is a set of pieces
  – A set of connected tuples from different tables
  – A sub-graph of the input graph
• Unstructured data is essentially the opposite!
  – A set of documents or web pages
• Each answer to a query is a single document
Much of the **high quality and valuable big data** are stored as semi-structured data (modeled as graphs):

- Enterprise’s Relational Databases
  - Banks, Insurance, ...
- Social Network’s Graph
  - Facebook, LinkedIn, ...
- XML repositories

1.4 Billion Nodes
400 Billion Edges

400 Million Nodes
80 Billion Edges

1000s of Relations
Millions of Rows
Challenges of Search in Graph-like Databases

• Current enterprise search engines requires:
  – Knowledge of **complex schema**
  – Knowledge of a **query language** (SQL, SPARQL)

• A non-technical user does **not** have this knowledge

```sql
SELECT title FROM conference c,
paper p, author a1, author a2,
write w1, write w2
WHERE c.cid = p.cid AND p.pid = w1.pid AND
p.pid = w2.pid AND w1.aid = a1.aid AND w2.aid = a2.aid AND
a1.name = "Jack" AND a2.name = "Sarah" AND c.name = "VLDB"
```
Challenges of Search in Graph-like Databases

- What about filling in **forms**?
  - Limited access pattern
  - Hard/Expensive to design
  - Hard to maintain on dynamic and heterogeneous data
Web-like Search for Big Graph Data

• Easy to use
  – Just a text box (keyword search)
• Familiar for anyone who ever has used Google/Bing
• Finding interesting or unexpected discoveries

“How are they related?”

“How does Graph Search work?”
Keyword Search in Big Graphs

- Given a graph with a set of query keywords, the goal is to find a sub-graph (e.g., tree), covering all of the keywords.
- **Content node**: a node that contains an input keyword.

![DBLP Graph][1]

Query 1: *Mehdi Divesh*
Query 2: *Jarek Divesh*
Issues of Previous Works

• **Weak** relationships among content nodes
• **Poor** performance
• **Solution**: Finding \( r \)-cliques

• **Exponential** number of answers
• **Duplicate** answers
  – Some answers have exactly the same set of content nodes
  – Need post-processing
• **Solution**: Enumerating answers in polynomial delay
Finding $r$-cliques

- An $r$-clique is a set of content nodes that together:
  - Contain all of the input keywords
  - The shortest distance between each pair of nodes is no longer than $r$.

- A new weight function is proposed based on the sum of distances between each pair of content nodes
  - The goal is to minimized the weight function

Very Large Databases (PVLDB)
Keyword Search in Graphs: Finding $r$-cliques
M. Kargar, A. An, 2011
Challenges

• Problem:
  – Given a distance threshold $r$, a graph $G$ and a set of input keywords, find an $r$-clique in $G$ whose weight is minimum

• We proved that the problem is NP-hard
  – By reduction from 3-SAT

• Solution:
  – We proposed an approximation algorithm with guaranteed ratio (2-approximation) for finding $r$-cliques
  – We further proposed a faster approximation algorithm with guaranteed ratio ($(t-1)$-approximation) for finding $r$-cliques
    • $t$ is the number of keywords
Challenges

• Problem:
  – Total number of answers is exponential regarding the number of input keywords
  – We want to produce unique set of content nodes (duplication free)

• For big graphs, it is not feasible to generate all answers and then sort them

• Solution:
  – Enumerating top-k answers in polynomial delay
  – Answers are produced in order of their weight
Enumerating Answers in Polynomial Delay

- The **Lawler’s** technique is used for finding the **top-k** answers

- In each iteration, the next *r*-clique is generated by finding the top answer under **constraints**

- The **constraints** result in **duplication free** answers

**Eugene Lawler**
Professor of CS at Berkeley
Constraints and Search Space

- Suppose that the **best (top) answer** contains nodes \{a, b, c\}
  - Best answer is found using our approximation algorithm
- Each search space has **two constraints**
  1. Inclusion set
  2. Exclusion set
- The sub-spaces are guaranteed to be **disjoint** (duplication free)

<table>
<thead>
<tr>
<th>Subspace</th>
<th>Inclusion Set</th>
<th>Exclusion Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>SB₁</td>
<td>{a, b}</td>
<td>{c}</td>
</tr>
<tr>
<td>SB₂</td>
<td>{a}</td>
<td>{b}</td>
</tr>
<tr>
<td>SB₃</td>
<td>{}</td>
<td>{a}</td>
</tr>
</tbody>
</table>

IEEE Transactions on Knowledge and Data Engineering (TKDE)
Efficient Duplication Free and Minimal Keyword Search in Graphs
M. Kargar, A. An, X. Yu, 2014
Overview of the System for Finding top-\(k\) Answers

- **Input** Keywords + Value of \(k\)
- **Find** best \(r\)-clique with no Constraint
- **Insert** the best \(r\)-clique with the search space in priority queue
- **Fetch** the best \(r\)-clique from priority queue and print it

**End**

**Top-k already printed?**

- **YES**
  - **Terminate**
  - **Insert** each answer with the related search space into priority queue

- **NO**
  - **Find** best \(r\)-clique in each sub-space with associated constrains
  - **Divide** the related search space of the top answer into sub-spaces
Outline

• **Keyword Search in Big Graphs**
  – VLDB’11, ICDE’12, TKDE’14, SIGMOD’14, KAIS’15, ICDE’15, CIKM’16

• **Team Formation in Social Networks**
  – CIKM’11, ICDMW’11, PKDD’12, SDM’13, WI’14, EDBT’17

• **Conclusions**
Team Formation in Social Networks

• What does a project need to be successful?
  – **Expertise** of the people
  – Effective **Communication**

• Social networks among professionals
  – LinkedIn
  – DBLP

• They form a **graph**
  – Each node is an expert
  – The edges determine previous collaboration
Example

- Project = \{AI, DB, DM, IR\}

- Smaller edges represents better communication

- This is **quite similar** to the graph keyword search problem, isn’t it?!
Our Contribution

Introducing the **cost** of the project

**Expertise**

**Communication**

**Personnel Cost**

---

*Effective Keyword Search over (Semi)-Structured Big Data*
Affordable Team Formation

• Find a team of experts that minimizes:
  – Communication cost
  – Personnel cost

• This is a bi-objective optimization problem

• So, how to solve it?!
Solving Bi-Objective Optimization Problems

1. **Combining** the two objective functions into a single one
   - Using a *trade off parameter* $\lambda$ between the communication and personnel costs

2. Finding a team of experts with a *bounded budget*

3. Finding *Pareto-optimal* teams
Combining the Two Objective Functions

- $\lambda$ is the **tradeoff** between the communication and personnel costs
  - $\text{CombFunc} = (\lambda).(\text{ComCostFunc}) + (1-\lambda).(\text{PersonCostFunc})$
- We proved that optimizing the new function is **NP-hard**
- We proposed:
  - An **approximation** algorithm with the ratio of 2
  - Two **greedy** algorithms
Finding Teams of Experts with Bounded Budget

• Give us your **personnel cost budget** (e.g., $20K)
  – We find the **most collaborative team** within your budget

• We proved the problem is NP-hard

• **$(\alpha, \beta)$-approximation** algorithm is used to solve the problem
  – $\alpha$ is the bound on first objective (personnel cost)
  – $\beta$ is the bound on second objective (communication cost)

• We propose a **$(\log n, 2)$-approximation** algorithm
  – $n$ is the number of required skills
Example: Best Team within Budget

- **Max Budget:** $300
- **Max Budget:** $100
- **Max Budget:** $50
Finding Pareto-Optimal Teams

• Pareto-optimal teams are a set of optimal solutions that are not dominated by others
• User is presented with a set of Pareto teams and choose one of them
• We proposed an approximation algorithm for finding Pareto teams
Example: Pareto-Optimal Teams

Communication Cost
- Team D: 62
- Team H: 78

Personnel Cost
- Team D: $43
- Team H: $62
Future Work – Team Formation

• Considering more constraints
  – Expertise of skill holders

• Adding one or more experts to an existing team to increase performance
  – The new member(s) should be able to communicate with existing members

• Due to a cut in the budget, we have to fire some team members
  – Who to fire?

• Assuming that a team lacks a particular skill, which of these approaches are more efficient?
  – Train an existing team member
    • Which one?
  – Hire a new one with the required skill
    • Who to hire?
  – Outsource the project
Effective Keyword Search over (Semi)-Structured Big Data

Outline

• Keyword Search in Big Graphs
  – VLDB’11, ICDE’12, TKDE’14, SIGMOD’14, KAIS’15, ICDE’15, CIKM’16

• Team Formation in Social Networks
  – CIKM’11, ICDMW’11, PKDD’12, SDM’13, WI’14, EDBT’17

• Conclusions
Effective Keyword Search over (Semi)-Structured Big Data

Collaborators

• AT&T Labs Research
  – Divesh Srivastava

• York University
  – Aijun An
  – Parke Godfrey

• University of Waterloo
  – Lukasz Golab

• University of Ontario Institute of Technology
  – Jarek Szlichta

• University of Toronto
  – Morteza Zihayat

• School of Information Technology, York University
  – Xiaohui Yu
Conclusions

- Accessibility of graph-like big data is an important area of research
- We have done some (hopefully) interesting work in this area
  - Keyword Search in Big Graphs
  - Team Formation in Social Networks
- Collaboration in Big Data Analytics related topics is of paramount importance
- A lot more research needs to be done!
Effective Keyword Search over (Semi)-Structured Big Data

mkargar@uwindsor.ca