<https://helda.helsinki.fi>

How the Quantum-inspired Framework Supports Keyword Searches on Multi-model Databases

Yuan, G.

ACM

2020-10

Yuan , G 2020 , How the Quantum-inspired Framework Supports Keyword Searches on Multi-model Databases . in CIKM '20 : Proceedings of the 29th ACM International Conference on Information & Knowledge Management . ACM , New York , pp. 3257-3260 , The 29th ACM International Conference on Information and Knowledge Management , Ireland , 19/10/2020 . <https://doi.org/10.1145/3340531.3418508>

<http://hdl.handle.net/10138/329733>

<https://doi.org/10.1145/3340531.3418508>

unspecified

acceptedVersion

Downloaded from Helda, University of Helsinki institutional repository.

This is an electronic reprint of the original article.

This reprint may differ from the original in pagination and typographic detail.

Please cite the original version.

How the Quantum-inspired Framework Supports Keyword Searches on Multi-model Databases

Gongsheng Yuan

University of Helsinki, Helsinki, Finland & Renmin University of China

ABSTRACT

With the trend of increasing vendors to develop various multi-model databases, people have reaped benefits from using a single and unified platform to manage both well-structured and NoSQL data. However, it causes a steep learning curve of mastering a multi-model query language for the specific multi-model database, not to mention various languages for different databases. Therefore, this research discusses the motivations of performing keyword searches on multi-model databases and then presents our current research. Methodologically, we attempt to use the quantum-inspired framework to query and explore multi-model databases. Firstly, we apply non-classical probabilities to estimate the relevance between a keyword query and candidate answers for guaranteeing getting good accuracy. Then we use the Principle Component Analysis (PCA) method to optimize the quantum language model for capturing good scalability. Finally, experiments show that our approaches are effective and our framework outperforms the state-of-the-art approaches.

CCS CONCEPTS

• **Information systems** → **Top-k retrieval in databases**; • **Computer systems organization** → Heterogeneous (hybrid) systems.

ACM Reference Format:

Gongsheng Yuan. 2020. How the Quantum-inspired Framework Supports Keyword Searches on Multi-model Databases. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM '20)*, October 19–23, 2020, Virtual Event, Ireland. ACM, New York, NY, USA, 4 pages. <https://doi.org/10.1145/3340531.3418508>

1 INTRODUCTION

Great reliance on the use of various applications to handle diverse transactions have caused the need of using different suitable data models efficiently dealing with data produced by those applications, these models include structured, semi-structured, and unstructured data. Against these data models, multi-model databases have been developed to manage them in a unified system [8, 9]. However, to access a multi-model database, users have to master the corresponding multi-model query language for the specific multi-model database. For example, we need to use AQL to access ArangoDB [2], use SQL++ for AsterixDB [1], use OrientDB SQL for OrientDB

[4]. This may take lots of energy to grasp these difficult query languages. Besides, understanding complex and possibly evolving multi-model data schemas are basic requirements for using these query languages. This may also bring lots of burden to users. For example, Figure 1 depicts multi-model data in social commerce. If we want to find who has bought Blizzard and given a perfect rating among the friends of Rubeus Hagrid, it won't be an easy job to write a multi-model query involving social network (Graph), order (JSON), and feedback (Relation) for novices.

Therefore, in this research, due to keyword searches having fewer limitations, we consider using it to query and explore multi-model databases. Through reviewing the literature, we could find that most existing works focus on performing keyword searches on a certain single data model, such as relational, XML, and graph, which results in the issue of performing keyword searches on multi-model databases lacking corresponding researches. Considering this point, we think the study about keyword searches on multi-model databases is a promising research topic. And there are still a lot of challenges waiting for solving. This is because simply combining results obtained by doing keyword searches on individual data models separately cannot work well. The reason is that it may miss some relevant results made up of components from different models. Besides, we also need a new measurement system to estimate the divergence between keyword queries and candidate answers for returning the most relevant top- k results which are what the user is interested in. Consequently, we could not directly use traditional keyword searches on multi-model data, and we need to find a novel way to do it.

To address this challenge, there are some advanced works to make some attempt, which utilize a graph model to represent heterogeneous data. However, some matters may need attention when using this strategy. Firstly, the constructed graph may be quite huge and complex. Secondly, when trying to find some subgraphs relating to a few keywords to answer queries, we also need to consider the dependencies among keywords and schema information. Thirdly, the problems like graph partition, subgraph matching, and subgraph relevance ranking, etc., have a significant impact on the final returned outcomes.

To avoid these announcements, my Ph.D. research considers how to do keyword searches on multi-model databases for getting good performance. Methodologically, we introduce the “quantum-inspired” framework into the database community. In this framework, we could regard the data in multi-model databases as the information (*statements*) collection, and then we could apply non-classical probabilities to estimate the relevance between a keyword query and candidate answers and help to return top- k the most relevant results, in which each of the results may be made up of multi-model components corresponding to pertinent information.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

CIKM '20, October 19–23, 2020, Virtual Event, Ireland

© 2020 Association for Computing Machinery.

ACM ISBN 978-1-4503-6859-9/20/10...\$15.00

<https://doi.org/10.1145/3340531.3418508>

Figure 1: An Example of Multi-model Data.

About "quantum-inspired" framework, the pioneering work [12] considers quantum theory as a formal language to describe the objects and processes in information retrieval. With this idea, we use this kind of mathematical language of quantum mechanics to describe relational, JSON, and graph data in the database community. Then, we could use the quantum language model to transform keyword searches from a querying-matching work into a calculating-ranking task with non-classical probabilities. And increasing researches begin to believe quantum-like phenomena (non-classical probabilities) could explain the relationship between human natural language and text, cognition and decision making [10]. This motivates us to apply it to our research.

The rest of this paper is organized as follows. Section 2 briefly introduces the basic concepts and quantum language model. Section 3 and 4 present our keyword search framework and optimization strategy, respectively. Section 5 shows the preliminary experimental results. Finally, the last section summarizes this paper and outlines our future work.

2 PRELIMINARIES

Based on Dirac notation, we denote a unit vector $\vec{u}$ as $|u\rangle = (u_1, u_2, \dots, u_n)^T$ and call it *ket*. The conjugate transpose $\vec{u}^H = \vec{u}^T$ is written as $\langle u|$ and called *bra*. For two vectors, the inner product between them is denoted as $\langle u|v\rangle = \sum_{i=1}^n u_i v_i$, the outer product is denoted as $|u\rangle\langle v|$ and called *dyad*. When the vector has unitary length (i.e., $\|\vec{u}\|_2 = 1$), an operator *projector* can be defined as the outer product between the (column) vector and its conjugate transpose (row) vector. For example, if there are $|u_1\rangle = (1, 0)^T$ and $|u_2\rangle = (\sqrt{1/3}, \sqrt{2/3})^T$, the projectors corresponding to them are:

$|u_1\rangle\langle u_1| = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$, $|u_2\rangle\langle u_2| = \begin{pmatrix} \frac{1}{3} & \frac{\sqrt{2}}{3} \\ \frac{\sqrt{2}}{3} & \frac{2}{3} \end{pmatrix}$. Then, we could use the

projector to map $|u\rangle$ to the quantum probability space, in which each rank-one dyad can represent a *quantum elementary event*. For the *superposition event* that is well-defined dyad $|\kappa\rangle\langle\kappa|$ represents a set of words, i.e., $\kappa = \{w_1, \dots, w_p\}$. The mapping function is defined as: $M(\kappa) = M(\{w_1, \dots, w_p\}) = |\kappa\rangle\langle\kappa|$, $|\kappa\rangle = \sum_{i=1}^p \sigma_i |e_{w_i}\rangle$, where the coefficients $\sigma_i \in \mathbb{R}$ satisfy $\sum_i \sigma_i^2 = 1$. A real density matrix is a Hermitian matrix ($\rho = \rho^H = \rho^T$), positive semidefinite ($\rho \geq 0$), and has trace 1 ($Tr(\rho) = 1$). For example, $\rho_1 = \begin{pmatrix} 0.75 & 0 \\ 0 & 0.25 \end{pmatrix}$, $\rho_2 =$

$\begin{pmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{pmatrix}$. By Gleason's theorem, a density matrix can be regarded as the proper quantum generalization of a classical probability distribution. It assigns a quantum probability to each one of the infinite dyads, whose formula is:

$$\mu_\rho(|u\rangle\langle u|) = Tr(\rho |u\rangle\langle u|). \quad (1)$$

For example, ρ_2 assigns a probability value 0.5 to the event $|u_1\rangle\langle u_1|$.

Although we do not know density matrices of query q and candidate answer c , we could use *Maximum Likelihood* (MaxLik) estimation to learn them. Then through the value of negative Von-Neumann Divergence (VND), we could know the relevance between the query and candidate answer [11]. Its formalization is:

$$-\Delta_{VN}(\rho_q || \rho_c) \stackrel{rank}{=} \sum_i \lambda_{q_i} \sum_j (\log \lambda_{c_j} \langle q_i | c_j \rangle^2). \quad (2)$$

3 OVERVIEW OF OUR APPROACH

Figure 2 shows an overview of our framework consisting of two parts. They are offline transformation and online keyword searches separately. The former part uses quantum language to represent heterogeneous data in a uniform format, density matrix. The later one applies quantum probabilities to answer queries.

To represent heterogeneous data in density matrices, we regard each tuple with schema in tables as a *statement* (a piece of information) for the relational data. For the JSON, we observe it in a coarse granularity to avoid information explosion and treat each object as an integral *statement*. For the graph, we put each node information and all of its neighbor node information in a *statement* to keep complete neighbor relationships. Of course, only this is not enough. We need to take join into consideration. For this problem, we could use a method like [3] to find the relationship among different data models and join them. Or, based on expert knowledge, we could do equi-join operations on meaningfully related domains, which is a straightforward way to find meaningful joins and cut down the search space by avoiding generating useless intermediate results. Since this is not the focus of this paper, we use the later one to continue our work. Next, we transform these statements into density matrices. In this process, to choose proper compounds κ from a given statement and determine the value of each coefficient σ_i in κ ,

Figure 2: The Framework of Quantum-inspired Keyword Search

we introduce a special spatial pattern concept called *co-location pattern* [5]. After these, we eliminate heterogeneous data and convert the querying-matching problem into calculating-ranking.

When a query comes, there are two data flows about it. One is about constructing the query density matrix from input keywords, which is from users to the constructor. In this step, we will consider the relationship among input keywords instead of treating them as mutually independent entities. And we will use the iterative schema of *Maximum Likelihood* (MaxLik) estimation to get query density matrix [11]. Another one is about reducing the query scope by our index structure, from users to the index structure.

4 OPTIMIZATION

Table 1: Queries employed on the small data set

ID	Queries
Q ₁	Glogy Heimi friend CCM ice hockey feedback perfect
Q ₂	Aerobed Guest Choice Inflatable Bed
Q ₃	Jose Lima SiteLite Ultra Mag Laser Professional Boresighter

Table 2: The breakdown time for queries in milliseconds

	Q ₁	Q ₂	Q ₃
Constructing matrices	4022	1088	3970
Others (mainly calculating)	25	139	172
Total time	4047	1227	4142

Table 2 gives the breakdown time of queries (Table 1) on a small data set (JSON: 26 objects, Relation: 2 tuples, Graph-entity: 9, Graph-relation: 6) which has economic data like Fig. 1. The time in Table 2 shows that we need optimization instead of using the quantum language model directly on the database community due to the database system requirements. When we use Π to represent a word or a compound, the probability of quantum event Π is $Tr(\rho\Pi)$. Through analysing $Tr(\rho\Pi)$, we could get the dot product of two vectors, i.e., $Tr(\rho\Pi) = \vec{\rho} \cdot \vec{\Pi}^2$, where $\vec{\rho} = (\beta_1, \beta_2, \dots, \beta_n)$, we call it *density vector*, $\vec{\Pi}^2 = ((\sum_{i=1}^n u_{i1}v_{i1})^2, (\sum_{i=1}^n u_{i1}v_{i2})^2, \dots, (\sum_{i=1}^n u_{i1}v_{in})^2)$.

Definition 4.1. Density vector

A density vector is a real vector $\vec{\rho} = (\beta_1, \beta_2, \dots, \beta_n)$, noted by $\langle \rho |$, where $\beta_i \geq 0$, and $\beta_1 + \beta_2 + \dots + \beta_n = 1$.

For example, density vector $\langle \rho_3 | = (0.2, 0.1, 0.7)$.

Now, we could simply original quantum language model and represent each word $w_i \in \mathcal{V}$ by $|e_i\rangle$, where $|e_i\rangle$, the standard basis vector, is an one-hot vector. As for compound, we represent $|\kappa\rangle = \sum_{i=1}^p \sigma_i |e_{w_i}\rangle$, where $\kappa = \{w_1, \dots, w_p\}$. Next, with the PCA method, we transform these statements into eigenvalues (density vectors) and eigenvectors (mapping directions). Then, we use density vector as the uniform format instead of density matrices to represent heterogeneous data. And we use

$$Probability_{\Pi} = \langle \rho | \Pi^2 \rangle. \quad (3)$$

to get the quantum probability of quantum event $|\Pi\rangle$, use

$$-\Delta_{VN}(\langle \rho |_q || \langle \rho |_c) \stackrel{rank}{=} \sum_i \beta_{q_i} \log \beta_{c_i} \quad (4)$$

where $\beta_{q_i} \in \langle \rho |_q$, $\beta_{c_i} \in \langle \rho |_c$, to get the relevance.

5 EXPERIMENT

Table 3: The number of records/objects in different data models

	Relational	JSON	Graph-entity	Graph-relation
IMDB	494 295	84 309	113 858	833 178

We use IMDB datasets [7] in Table 3 to evaluate our approach, Quantum-inspired framework with PCA (QIPCA). The IMDB dataset is crawled from website¹ by OMDb API. Through extracting inner relationships and potential information, we generate several data models to represent the original data. We use keyword queries in Table 4 to evaluate QIPCA, in which each keyword query includes one or more data models to test the ability of QIPCA in catching potential semantics of queries and search accuracy. Besides, we do the corresponding AQL query for each Q_i in ArangoDB databases and use the output answers as standards to evaluate the performance of QIPCA and EASE [6]. Here, we do all operations in memory and conduct the standard NLP pre-processing, such as dropping the stop words and stemming in advance. In the experiments, we use the top-20 returned results to estimate search accuracy. Table 5 shows the comparison of the average precision, recall, and F-measure of QIPCA with EASE's. It demonstrates that our proposed methods

¹<https://www.omdbapi.com/>

Table 4: Queries employed in the experiments

ID	Queries
Q_1	Lock, Stock and Two Smoking Barrels actors
Q_2	Forrest Gump
Q_3	The title and imdbVote of films of Bruce Willis
Q_4	The films of director Robert Zemeckis
Q_5	films in 1997 Genre Action, Adventure, Family
Q_6	The Legend of 1900 awards
Q_7	Scent of a Woman imdbRating
Q_8	The film of Dustin Hoffman with Tom Cruise
Q_9	Morgan Freeman friends
Q_{10}	Aamir Khan films

Table 5: Presion, Recall, F-measure on IMDB

$min_threshold$	QIPCA					EASE
	0.2	0.4	0.6	0.8	nonSpatial	
Precision	0.753	0.753	0.758	0.758	0.758	0.548
Recall	0.782	0.782	0.784	0.784	0.784	0.466
F-measure	0.657	0.657	0.661	0.661	0.661	0.269

Figure 3 presents the executing time of QIPCA and EASE under the situation of $min_threshold$ being 0.6 for constructing compounds with spatial pattern method. Here, due to the different complexities of queries involving different numbers of data models, there is a different fluctuation in the scales of time. But in general, QIPCA has a good time performance.

Figure 3: Executive time of Q_i

Figure 4: Scalability

Figure 4 manifests a good linear growth trend of the scalability as the loading proportions of metadata into QIPCA increase, where for each query, correct results increase as loading more data too.

6 CONCLUSION AND FUTURE WORK

In this research, we explore introducing quantum probability mechanism into keyword searches of databases. We proposed new representations of words and compounds, present new concepts density vector, and use the spatial pattern and PCA to reduce the complexity of quantum-inspired framework and improve query performance.

The general plan to complete my Ph.D. dissertation is to focus on the two parts shown in Figure 2. First, due to the different workloads and queries, we may introduce the machine learning or quantum phenomena-based method to automate choosing appropriate join to construct meaningful statements. And make the density matrices include more information. Second, we will optimize our ranking system for returning more accurate answers based on density matrices. Thirdly, we want to present a suitable index structure based on density matrices instead of using an inverted list. Finally, we will use more experiments to estimate our approaches from different perspectives. Another extension is to investigate using quantum theory to found the unified multi-model query language.

ACKNOWLEDGMENTS

The author would like to thank his doctoral advisor of Prof. Jiaheng Lu. The author would also like to thank all the reviewers for their valuable comments and helpful suggestions. The work is supported by the China Scholarship Council and CIMO Fellowship.

REFERENCES

- [1] Sattam Alsubaiee, Yasser Altowim, Hotham Altwaijry, Alexander Behm, Vinayak Borkar, Yingyi Bu, Michael Carey, Inci Cetindil, Madhusudan Cheelang, Khurram Faraaz, Eugenia Gabrielova, Raman Grover, Zachary Heilbron, Young-Seok Kim, Chen Li, Guangqiang Li, Ji Mahn Ok, Nicola Onose, Pouria Pirzadeh, Vassilis Tsotras, Rares Vernica, Jian Wen, and Till Westmann. 2014. AsterixDB: A Scalable, Open Source BDMS. *Proc. VLDB Endow.* 7, 14 (Oct. 2014), 1905–1916. <https://doi.org/10.14778/2733085.2733096>
- [2] ArangoDB. Accessed 1 July 2020. *Three major NoSQL data models in one open-source database.* <http://www.arangodb.com/>
- [3] Sonia Bergamaschi, Elton Domnori, Francesco Guerra, Raquel Trillo Lado, and Yannis Velegrakis. 2011. Keyword search over relational databases: a metadata approach. In *SIGMOD '11*.
- [4] OrientDB Developers. Accessed 1 July 2020. *OrientDB, Hybrid Document-Store and Graph NoSQL Database.* <https://orientdb.com/>
- [5] Yan Huang, Shashi Shekhar, and Hui Xiong. 2004. Discovering colocation patterns from spatial data sets: a general approach. *IEEE Transactions on Knowledge and data engineering* 16, 12 (2004), 1472–1485.
- [6] Guoliang Li, Jianhua Feng, Beng Chin Ooi, Jianyong Wang, and Lizhu Zhou. 2011. An effective 3-in-1 keyword search method over heterogeneous data sources. *Information Systems* 36, 2 (2011), 248–266.
- [7] Jiaheng Lu, Jun Chen, and Chao Zhang. 2016. Helsinki Multi-Model Data Repository.
- [8] Jiaheng Lu and Irena Holubová. 2019. Multi-Model Databases: A New Journey to Handle the Variety of Data. *ACM Comput. Surv.* 52, 3, Article 55 (June 2019), 38 pages. <https://doi.org/10.1145/3323214>
- [9] Jiaheng Lu, Irena Holubová, and Bogdan Cautis. 2018. Multi-Model Databases and Tightly Integrated Polystores: Current Practices, Comparisons, and Open Challenges. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management (Torino, Italy) (CIKM '18)*. Association for Computing Machinery, New York, NY, USA, 2301–2302. <https://doi.org/10.1145/3269206.3274269>
- [10] Dawei Song, Mounia Lalmas, Keith Van Rijsbergen, Ingo Frommholz, Benjamin Piwowarski, Jun Wang, Peng Zhang, Guido Zucco, Peter Bruza, Sachi Arafat, et al. 2010. How quantum theory is developing the field of information retrieval. In *2010 AAAI Fall Symposium Series*.
- [11] Alessandro Sordani, Jian-Yun Nie, and Yoshua Bengio. 2013. Modeling term dependencies with quantum language models for IR. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*. 653–662.
- [12] C. J. van Rijsbergen. 2004. *The Geometry of Information Retrieval*. Cambridge University Press. <https://doi.org/10.1017/CBO9780511543333>