Towards A Framework For Tensor Ontologies Over Neo4j: Representations And Operations

Georgios Drakopoulos*, Andreas Kanavos‡ , Dimitrios Tsolis† , Phivos Mylonas*, and Spyros Sioutas*
*Ionio University, Department of Informatics
Plateia Tsirigoti 7, Kerkyra 49100, Hellas
e-mail: {drakop, mylonas, sioutas}@ionio.gr
†University of Patras, Cultural Heritage Management and New Technologies Department
Seferi 2, Agrinion 30100, Hellas
e-mail: dtsolis@upatras.gr
‡University of Patras, Computer Engineering and Informatics Department
Patras, Achaia 26504, Hellas
e-mail: kanavos@ceid.upatras.gr

***Abstract*—Ontology has been an active research field connecting philosophy, logic, history, mathematics, and computer science to name a few. Within an ontological context defined over a domain the entities as well as their associated relationships can be represented by the vertrices and the edges of a tree. From the latter new knowledge can be then inferred through a number of techniques including Horn logic from reasoners and RDF triplets. With the advent of the Semantic Web and sophisticated associated software tools including graph databases such as Neo4j, Sparksee, and TitanDB or XML parsers such as Xerces graph mining is done efficiently on the semantic level instead of the combinatorial or algebraic ones. Multilayer graphs, namely graphs whose labeled edges belong to a number of predetermined classes, have been recently introduced in social network analysis in order to represent the different interaction options between netizens. In this work the potential of applying this new type of graphs to an ontological context creating essentially an ontological tensor is outlain and its complexity is assessed. A human readable dataset based on the late 1970s and early 1980s Apple manually constructed from the 2011 officially authorized biography of Steve Jobs and the 1999 film *Pirates of Silicon Valley* serves as a concrete example complete with Neo4j queries.**

***Index Terms*—Neo4j; Cypher; Query sublanguages; Multilayer graph; Tensor algebra; Higher order analytics; Ontological tensor; Semantic Web; Steve Jobs biography**

I. Introduction

A new digital era is starting where machine intelligence is expected to play a prominent role. Ontology, namely the procedure of discovery and formal naming of entities and their associated relationships and properties in a domain, is one of the mainstays of this intelligence. Following the fundamental notions of ontological studies in philosophy from Parmenides to Edmund Husserl and Martin Heidegger, entities are divided in categories hierachically and resursively [1] [2]. With the advent of Semantic Web structured data from Web crawlers in the form of XML, and in the near future possibly of SGML, trees [3] [4]. The latter can then be stored in a graph database like Neo4j and an automated theorem prover or a suite of graph analytics can discover latent non-trivial knowledge. A typical instance of the former is a reasoner driven by Horn logic with definite clauses of the disjunction form

$$\neg t_1 \vee \neg t_2 \vee \ldots \vee \neg t_p \vee g \tag{1}$$

where $\{t_k\}$ are the data clauses and g is the objective clause. A a common instance of graph analytics is community discovery or, equivalently, graph partitioning [5] [6] [7].

Ontologies can be represented as directed graphs where a vertex corresponds to an entity and an edge (u_1, u_2) coupled with a predicate p such as *is.a* or *has.a* denotes that u_1 and u_2 are semantically connected. This is codified in the triplet

$$(u_1, u_2, p) \tag{2}$$

The introduction of multiple edges with pairwise distinct labels, and hence standing for different predicates, connecting the same pair of vertices can lead to new inference possibilities. Alternatively, the same edge can corrspond to a set of predicates instead of a single predicate. This is exactly the class of *multilayer graphs* where the tuples have the form

$$(u_1, u_2, \{p_1, p_2, \ldots, p_n\}) \tag{3}$$

The primary contribution of this conference paper is an ontological scheme whose algorithmic cornerstone are ontological multilayer graphs and their algebraic counterpart the *ontological tensors*. The proposed scheme is applied to a dataset containing key people from the early Apple days and the various relationships between them.

The remainder of this work is organized as follows. In section II the scientific literature regarding ontology and tensor algebra is overviewed. The main properties of graph databases are described in section III. Subsequently, in section IV the proposed framework is presented. The dataset manually created from the official Steve Jobs biography [8] and the film *Pirates of Silicon Valley* [9] is described in section V. Finally, in section VI future research directions are discussed. The notation of this work is summarized in table I. Vectors are symbolized with boldface lower case letters, matrices with capital boldface, and tensors with capital italics.

TABLE I
PAPER NOTATION.

Symbol	Meaning
$\triangleq$	Definition or equality by definition
$S = \{s_1, \ldots, s_n\}$	Set containing elements $s_1, \ldots, s_n$
$T = (t_1, \ldots, t_n)$	Tuple containing elements $t_1, \ldots, t_n$
$\|S\|$ or $\|T\|$	Set or tuple cardinality
$\Gamma_i(u, w)$	Inbound neighbor set of u for relationship w
$\Gamma_o(u, w)$	Outbound neighbor set of u for relationship w
$\mathrm{lca}\,(u_1, u_2)$	Least common ancestor of u_1 and u_2
μ	Metric defined on the ontological data
$H(\mathbf{x})$	Discrete entropy of vector $\mathbf{x}$
$\|\mathcal{T}\|_F$	Frobenius norm of tensor $\mathcal{T}$

II. Previous Work

Ontology is a relatively new breakthrough in computer science [10]. However, there is a plethora of applications including the creation of personalized ontologies for content retrieval within a particular context [11], object discovery in images through semantics [12], and query refinement using personalized social content [13]. Moreover, new fields such as bioengineerging also benefit as for instance databases [14], genetic reasoners [15] [16], and mining strategies for medical documents [17] [18] rely on human gene ontology [19] [20].

Graphs constitute the primary and most intuitive ontology representations [3]. Ontological dependencies and alignment are examined in [21], whereas informatioc theoretic metrics are proposed for ontological graphs in [22]. Graph databases such as Neo4j and TitanDB provide graph queries with sublanguages such as Cypher, Gremlin, XPath, and SPARQL, while large graph processing systems such as Apache Giraph and Google Pregel offer inference capabilities [23]. Finally, current programming languages and distributed computing platforms support in memory graph handling with specialized libraries such as NetworkX for Python and GraphX for Apache Spark.

Tensors or multiway arrays are the algebraic counterparts of the combinatorial multilayer graphs as well as the next evolutionary step in linear algebra [24]. Among others, they have been also central in complex network modelling and synthetic graph generation [25] and in representing multimodal social network functionality [26]. Tensor clustering is an NP-hard problem with numerous applications including multilayer graph partitioning [27]. Google TensorFlow works natively with tensors, whereas TensorLab and Tensor Toolbox extend existing MATLAB multidimensional matrix functionality.

III. Graph Databases

Lately there has been a strong engineering interest in the four technological branches of the non-relational databases collectively known as NoSQL databases [28]. Table II contains the aforementioned branches along with the primary data type and some of the most well known databases of that type. Of course, the database list is very far from being exhaustive.

Property 1: NoSQL databases are schemaless [28].
This is the crucial difference between relational and NoSQL databases. However, the latter do not completely lack an abstract data description. For instance, JSON-LD is a graphic and textual such description only imposing light organizational requirements [30].

Property 2: The operational NoSQL requirements are known as BASE [28]:

- **B**asic **a**vailability.
- **S**oft state.
- **E**ventual consistency.

Compared to ACID, BASE are more easily implemented. Moreover, the load for underlying database system is reduced, since less data duplications, file locks, and network packets are required. Also, shorter internal integer descriptors are necessary after data merges, ensuring smoother and quicker overall functionality. The cost for these advantages is reduced consistency, especially for distributed databases.

Property 3: The typical functionality set is CRUD [28]:

- **C**reate.
- **R**ead.
- **U**pdate.
- **D**elete.

The potential for rollback in Neo4j CRUD operations through persisent data structures, adding transactional-like capabilities, was studied in [37].

Neo4j, currently in version 3.1.1, is a scalable and open source graph database implemented mostly in Scala. Graphs can be queried primarily through Cypher, an ASCII art, high abstraction, and declarative sublanguage. Cypher APIs for Python, Java, C#, C++, and Scala exist. Alternatively, Neo4j understands SAIL and SPARQL queries for quartets and triplets respectively.

The basic Cypher query has the following form [29]:

```
match <pattern>
[where <constraint>]
return <result> [as <expression>]
[order by <field> [desc]]
[limit <constant>]
```

These queries are closely patterned after a property graph, the principal abstract data model of Neo4j. This model is designed for simplicity since the graph can contain information in key-value pairs in both vertices and edges [28]. Since Neo4j is schemaless, this information can differ between any two vertices or any two edges, making thus necessary a number of checks such as the existence of a given key or the type and cardinality of the returned value. Although ontologies are typically highly structured, there are emerging where this many actually not be the case. Consider, for instance, data streaming, deep learning, or big data applications where ontologies may be progressively constructed from highly volatile and quickly evolving data. Moreover, in cross-domain knowledge transfer or ontology alignment scenaria the resulting ontology may well be incomplete for various reasons.

Selecting all edges with a specific label requires the following Cypher pattern

```
match ()-[e:LABEL]-()
return distinct(e)
```

TABLE II
NoSQL DATABASES.

Type	Primary Data Format	Software
Graph	Conceptual graph [29], RDF [28], JSON-LD [30]	Neo4j, Sparksee, TitanDB
Key-value	Associative array [31]	Amazon Dynamo, Riak, Redis, Apache Ignite
Column family	Large columns [32]	Apache Cassandra, Keyspace
Document	JSON [33], BSON [34], XML [35], YAML [36]	MongoDB, CouchDB, OrientDB

Along similar lines computing the number of vertices which are of a give type is done with the query

```
match (v:type)
return count(distinct(v))
```

Over Neo4j a number of graph analytics, most of them of higher order nature, can be developed. Eigenvector-based vertex centrality rankings, which indicate the role of a single vertex plays in total graph connectivity, include PageRank, the eigenvector centrality [29], and Gell point centrality [38]. This ranking class relies on solving

$$\ell(\mathbf{A})\mathbf{g} \;=\; \lambda\mathbf{g} \tag{4}$$

where $\mathbf{A}$ is the graph adjacency matrix and ℓ is a linear mapping. The same objective is accomplished in a different way by matrix series centrality such as Neumann, Mercator, and resolvent centrality [39]. This class of metrics computes the diagonal elements of the matrix power series

$$g(\mathbf{A}) \triangleq \sum_{k=0}^{+\infty} \gamma_k \mathbf{A}^k \tag{5}$$

The Estrada index, which is a metric of graph strength, can be approximated as in [40]. At its core lies the sum

$$s \triangleq \sum_{k=1}^{n} h(\lambda_k) \tag{6}$$

where $\{\lambda_k\}$ are the n eigenvalues of $\mathbf{A}$. In contrast to ℓ, g and h are non-linear mappings in the general case. Finally, from a systemic viewpoint large cardinality estimators like the one in [41] can be used internally to accelerate response times in aggregative queries.

IV. Framework

A. Intuition

Before presenting the proposed ontological framework, certain design principles concerning complexity are given. Although they cannot be directly programmed, they shed light to the fundamental patterns of complexity in large systems.

Principle 1: (Leibniz's warning) But when a rule is extremely complex, what it is in conformity with it passes for irregular.

Principle 2: (Occam's razor or *lex parsimoniæ*) Entities must be not multiplied beyond necessity or in Latin *Non sunt multiplicanda entia sine necessitate*.

Principle 3: (Thorngate's postulate) In order to increase both the generality and the accuracy, the complexity of our theories must necessarily be increased.

The first two principles complement each other and esentially urge against excessive complexity both in rules, namely programming, and in entities, namely the data model. In fact, the first principle connects high complexity levels to randomness and potentially to white noise. The third principle acts in complexity approximately as Heisenberg's principle in quantum mechanics.

B. Representations

An ontology O can be formally defined in terms of ground level objects, classes, attributes, relations, functions, restrictions, rules, axioms, and events. Depending on the underlying domain, one or more of these factors may be trivial or undefined. For the purposes of this work the following simplified defintion will be used throughout the text

Definition 1: An ontology O is the ordered triplet

$$O \triangleq (J, W, T) \tag{7}$$

where J is the ground level object set, W is the relation set, and T is the attribute set.

Higher order data can be represented combinatorially as multilayer graphs or algebraically as tensors. An ontology, essentially being a higer order data model from the computer science point of view, satisfies this criterion.

Definition 2: A multilayer graph G is the ordered quintuple

$$G \triangleq (V, E, Q, \Sigma, f) \tag{8}$$

where V is the vertex set, $E \subseteq V \times V \times Q$ the edge set, Q the label set, and Σ the edge value set. The function $f : E \to \Sigma$ assigns to edges a value.

The entities of an ontology are mapped through a bijection to the vertices of the multilayer graph. For each type of connection w, if the entities u_1 and u_2 are related through w, then an edge labeled w connects u_1 and u_2. This edge is denoted by (u_1, u_2, w). Additional domain-specific properties of w can be encoded as positive integers and mapped to (u_1, u_2, w) through f if necessary. Algorithm 1 shows the construction of an ontological multigraph in worst case time $\mathrm{O}\left(\left|J^2\right|(|W| + |T|)\right)$ in serial execution. However, if there are no interdependencies in W or in T, then the double loop in lines 8 to 17 can be trivially parallelized.

Ontological tensors encapsulate multimodality in a natural way, in the sense that each relation has its own separate representation. Formally

Definition 3: A tensor $\mathcal{T} \in \mathbb{S}_1 \times \mathbb{S}_2 \ldots \times \mathbb{S}_p$ of order p is a linear mapping simultaneously connecting p not necessarily distinct linear spaces $\mathbb{S}_k$, $1 \leq k \leq p$.

Algorithm 1 Ontology conversion to multilayer graph

Require: Ontology $O(J, W, T)$, integer encoding scheme K
Ensure: Ontological multilayer graph G is created

create G with $|J|$ vertices and no edges
for all attributes $t \in T$ **do**
  map t to $K(t) \in \mathbb{Z}^+$
end for
for all entities $j \in J$ **do**
  map exactly one entity j to one vertex u_j
end for
for all connections $w \in W$ **do**
  for all ordered pairs $(j_1, j_2) \in J \times J$ **do**
    if for (j_1, j_2) is **true then**
      insert edge (u_{j_1}, u_{j_2}, w)
      if for (j_1, j_2) **and** w attribute t is **true then**
        insert $K(t)$ as property of (u_{j_1}, u_{j_2}, w)
      end if
    end if
  end for
end for
return G

A p-th order tensor is a p-dimensional array containing real or complex values and indexed by a tuple of p integers $(i_1, i_2, \ldots, i_p)$. This is a direct generalization of linear algebra textbook matrices which are two dimensional arrays connecting two linear spaces, namely the row space and the column space, and are indexed by a tuple of two integers.

In the ontology case a third order tensor $\mathcal{G} \in \mathbb{Z}^{|J| \times |J| \times |W|}$ can be constructed. As its dimensions indicate, $\mathcal{G}$ connects the object space to itself and to the relation space. This is analogous to the term-document matrix in information retrieval which connects the term space to the document space. The difference is that $\mathcal{G}$ has more flexibility as it represents $|W|$ distinct relationships that map J to itself. Which exactly these relationships are is determined by the underlying domain. The values of $\mathcal{G}$ are

$$\mathcal{G}[i_1, i_2, i_3] \triangleq \begin{cases} K(t), & (u_{i_1}, u_{i_2}, w_{i_3}) \text{ is true} \\ 0, & \text{otherwise} \end{cases} \tag{9}$$

Now becomes clear why K is restricted to map T to positive integers. Both from a linear algebraic and a knowledge mining viewpoint, the sparsity patterns of $\mathcal{G}$, namely the number and the systematic locations of the zero values, are important. If no attributes exist for the given pair of ground level objects and relationship, then the value 1 can be inserted.

C. Operations

As is the case with any nontrivial data representation scheme, ontological tensors are only as good as their interpretation and knowledge mining power. In turn, the latter relies heavily on the operations which can be rigorously defined on the data described by them. Table III lists the operations which can be applied to an ontological tensor. An algebraic operation is applied to tensor $\mathcal{G}$, whereas a combinatorial operation is applied to multilayer graph G.

TABLE III
ONTOLOGICAL TENSOR OPERATIONS.

Operation	Type
Partition	Algebraic, combinatorial
Least common ancestor	Combinatorial
Concept similarity	Combinatorial
Sparsity	Algebraic, combinatorial
Degree distributions	Algebraic, combinatorial
Frobenius norm	Algebraic

Concept similarity metrics based on concepts have been designed for computing ontological distances and take advantage of the tree structure of the ontology. They include thesaurus-based metrics such as path, Wu-Palmer, Leacock-Chodorow, and combined thesaurus- and corpus-ones such as Resnik, Lin, and Jiang-Conrath metrics [42]. The Wu-Palmer metric is

$$\mu_{wp}(u_1, u_2) \triangleq \frac{2d(\operatorname{lca}(u_1, u_2))}{d(u_1) + d(u_2)} \tag{10}$$

where $d(s)$ is the depth of vertex s in the tree. The Leacock-Chodorow metric is defined as

$$\mu_{lch}(u_1, u_2; b) \triangleq -\log_b \left(\frac{\zeta(u_1, u_2)}{2 \max_s d(s)} \right) \tag{11}$$

where the numerator is the shortest distance length between u_1 and u_2 and the denominator is twice the maximum tree depth. This is related to the entropy $H(\mathbf{x})$ of a data vector $\mathbf{x}$

$$-\sum_{k=1}^{n} \operatorname{prob}\{\mathbf{x}[k] = x_k\} \log_b \operatorname{prob}\{\mathbf{x}[k] = x_k\} \tag{12}$$

where $\operatorname{prob}\{\mathbf{x}[x] = x_k\}$ is the probability that the k-th element of $\mathbf{x}$ has the specific value x_k. The logarithm base b expresses the unit of information measure.

The density ρ_0 of any tensor $\mathcal{T}$ is the ratio of the nonzero elements to its total number of elements. Similarly, the logdensity ρ_0' is the ratio of the logarithm of each size.

For each vertex u it is possible to define a vector $\mathbf{y}\mathbf{u}_i$ of length $|W|$ where each element is the number of inbound neighbors for each $w \in W$ divided by the total number of edges labeled w. The same can be done with the outbound neighbors again for each relationship yileding $\mathbf{y}(u)_o$. The entropy of both vectors is a profile of u regarding its preference to labels. If u has approximately the same number of inbound or outbound neighbors m for each relationship w, then the corresponding entropy should be close to

$$\log_b \left(\sum_{w \in W} |\Gamma_i(u, w)| \right) = \log_b (m|W|) \tag{13}$$

On the contrary, if u has edges of only one label, then the entropy of the corresponding vector is zero.

Finally, for a real valued tensor $\mathcal{T} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_p}$ its Frobenius norm is a complexity measure defined as

$$\|\mathcal{T}\|_F \triangleq \left(\sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \ldots \sum_{i_p=1}^{I_p} \mathcal{T}^2[i_1, i_2, \ldots, i_p] \right)^{\frac{1}{2}} \tag{14}$$

TABLE IV
APPLE DATASET PERSONS (INDICATIVE).

Person	Role
Steve Jobs	Apple co-foudner
Steve Wozniak	Apple co-founder and friend of Jobs
Mike Markkula	Investor and friend of Jobs
Arthur Rock	Investor
John Sculley	Apple CEO (1983-1993)
Regis McKenna	Silicon Valley marketeer

TABLE V
APPLE DATASET RELATIONS (INDICATIVE).

Personal	*likes*	*friend.of*	*trusts*	*dislikes*	*college.with*
Business	*works.for*	*partner.of*	*funds*	*manages*	*rival.of*

However, the values of $\mathcal{G}$ do not corresponnd to edge valence and therefore they cannot be summed in this case.

D. Complexity

Regarding memory requirements, $\mathcal{G}$ occupies a potentially large volume of P RAM pages in a single host where

$$\left\lceil \frac{1}{M} \left\lceil \frac{\rho_0 |J|^2 |W| b_0}{B} \right\rceil \right\rceil \leq P \leq \left\lceil \frac{1}{M} \left\lceil \frac{|J|^2 |W| b_0}{B} \right\rceil \right\rceil \quad (15)$$

In the above equation

$$b_0 \triangleq \max \left\{ 1, \max_{K(t)} \lceil \log_2 K(t) \rceil \right\} \quad (16)$$

Also, B is the length in bits of the integer size used to store $\mathcal{G}$, and that each RAM page can hold up to M such integers. The ceil functions represent overhead due to remainders and possibly memory alignment requirements imposed by the CPUs and the local operating system configuration. An estimation similar to (15) can be derived for disk blocks, provided that overheads due to alignment and serialization are accounted for.

V. APPLICATION: APPLE DATASET

A. Dataset Synopsis

Some of the persons of the current version of the Apple dataset are listed in table IV and were taken from [8] and [9][1]. Readers will probably be familiar with the most of these names which represent a major part of the Silicon Valley history. For instance, Regis McKenna conceived and executed marketing campaigns for many of the most well known US high technology companies and Arthur Rock has been one of the early investors in technology companies, helping among others the so-called *traitorous eight*.

Some of the relations of the Apple dataset are listed in table V. They are divided into two categories, business and personal. Also, there are 31 persons in total with 499 directed edges connecting them. Thus, only slightly more than a quarter of the total connections are used. Finally, $K(t)$ was always 1.

[1]It is worth mentioning that, according to actor Noah Wyle, Steve Jobs personally called him and congratulated him for his performance.

TABLE VI
ENTROPY FOR $\mathbf{y}(u)_o$ AND $\mathbf{y}(u)_i$ (PERSONAL AND BUSINESS).

$H(\mathbf{x})$	**min**	**avg**	**max**	**min**	**avg**	**max**
Personal	0.1711	1.3131	3.4406	0.2135	1.2835	2.9313
Business	0.4429	1.7346	4.1284	0.1184	0.9979	1.2464

TABLE VII
PAIRWISE μ_{lch} (PERSONAL AND BUSINESS).

μ_{lch}	**min**	**avg**	**max**	**min**	**avg**	**max**
Personal	0.0031	0.3415	0.7633	0.1241	0.3636	0.7992
Business	0.0012	0.2533	0.4417	0.0000	0.2340	0.6102

B. Results

For each of the persons in the dataset the entropy of the vectors $\mathbf{y}(u)_i$ and $\mathbf{y}(u)_o$ was computed. The number of each inbound neighbors for relationship w was computed as

```
match (s)-[:w]->(u)
return id(u), count(distinct(s))
```

The total number of edges labeled w was computed with the query of section III. In order to compute $\mu_{lch}(u_1, u_2)$ for relationship the shortest path length between u_1 and u_2 was computed with

```
match p=(u1)-[:w*]->(u2)
return u1, u2, length(p)
order by length(p)
```

The maximum depth was the maximum of the longest paths.

The values of of both tables VI and VII indicate that for the *business* relationship the inbound and outbound degree distributions differ, whereas for the *personal* relationship they coincide to a great extent. This is expected, since the majority of current business models establish hierarchical operational paths, whereas personal human interaction typically has a high degree of reciprocity.

This pattern is also visible in μ_{lch}. While for the personal relationships the values for the inbound and outbound similarities are approximately the same, the sitution is much different in business relationships. In fact, in the latter there is a considerable number of vertex pairs who has zero similarity. This is attributed to the asymmetric nature of ontologies, with the relatonship between employer and employee been the most asymmetric. Also, the values for the *business* relationship are consistently lower that those of the *personal* relationship. Again, this is a result of the stronger assymetries found in modern business environments.

VI. FUTURE WORK

This conference paper presents the fundamental concepts which lay the groundwork for representing an ontology in two ways. The first is a multilayer graph, namely a graph with labeled edges which can be naturally stored in Neo4j. This combinatorial representation allows for multiple connections between any given vertaix pair, enabling the coexistence of more than one relationships between the ground level entities without the need for elaborate coding schemes. The algebraic counterpart is an ontological tensor, which has the same expressive power but a different operation set. Both representations were applied to an open dataset with persons and relationships extracted from the offical Steve Jobs biography and the 1999 film *Pirates of Silicon Valley*.

Temporal information is absent from the proposed methodology. In certain domains, mostly in Web 2.0 event-driven and deep learning ones, this in a significant factor. This can be remedied by inserting a time dimension resulting in a very large fourth order tensor. This presents certain challenges as the need for distributed processing and big data techniques.

Complexity can be reduced by pruning superfluous relationships, either existing or discovered, with techniques such as those in [43]. Additionally, advanced semantics such as Kripke models [44] [45] would increase the expressive strength of the initial ontology.

Reproducible Research Note

In order to assist researchers in their work, the Apple dataset will be uploaded in human readable form as RDF triplets on the corresponding Researchgate site as supplementary data.

References

[1] M. Theunissen, *The other: Studies in the social ontology of Husserl, Heidegger, Sartre and Buber*. MIT Press, 1984, vol. 10.

[2] M. Furth, "Elements of Eleatic ontology," *Journal of the History of Philosophy*, vol. 6, no. 2, pp. 11–132, 1968.

[3] G. Antoniou and F. Van Harmelen, *A semantic Web primer*. MIT press, 2004.

[4] N. Shadbolt, T. Berners-Lee, and W. Hall, "The semantic Web revisited," *IEEE intelligent systems*, vol. 21, no. 3, pp. 96–101, 2006.

[5] A. Kanavos, G. Drakopoulos, and A. Tsakalidis, "Graph community discovery algorithms in neo4j with a regularization-based evaluation metric," in *WEBIST*, April 2017.

[6] E. Kafeza, A. Kanavos, C. Makris, and D. K. W. Chiu, "Identifying personality-based communities in social networks," in *Advances in Conceptual Modeling - ER*, 2013, pp. 7–13.

[7] E. Kafeza, A. Kanavos, C. Makris, and P. Vikatos, "T-PICE: Twitter personality based influential communities extraction system," in *International Congress on Big Data*, 2014, pp. 212–219.

[8] W. Isaacson, *Steve Jobs*. Simon and Schuster, October 2011.

[9] Pirates of Silicon Valley, "http://www.imdb.com/title/tt0168122/," June 2017.

[10] A. Maedche and S. Staab, "Ontology learning for the Semantic Web," *IEEE Intelligent systems*, vol. 16, no. 2, pp. 72–79, 2001.

[11] D. Vallet, P. Castells, M. Fernández, P. Mylonas, and Y. Avrithis, "Personalized content retrieval in context using ontological knowledge," *IEEE Transactions on circuits and systems for video technology*, vol. 17, no. 3, pp. 336–346, 2007.

[12] T. Athanasiadis, P. Mylonas, Y. Avrithis, and S. Kollias, "Semantic image segmentation and object labeling," *IEEE Transactions on circuits and systems for video technology*, vol. 17, no. 3, pp. 298–312, 2007.

[13] I. Anagnostopoulos, G. Razis, P. Mylonas, and C.-N. Anagnostopoulos, "Semantic query suggestion using Twitter Entities," *Neurocomputing*, vol. 163, pp. 137–150, 2015.

[14] L. M. Schriml, C. Arze, S. Nadendla *et al.*, "Disease Ontology: backbone for disease semantic integration," *Nucleic acids research*, vol. 40, no. D1, pp. D940–D946, 2012.

[15] M. Bada, D. Turi, R. McEntire, and R. Stevens, "Using reasoning to guide annotation with gene ontology terms in GOAT," *SIGMOD*, vol. 33, no. 2, pp. 27–32, 2004.

[16] S. Myhre, H. Tveit, T. Mollestad, and A. Lægreid, "Additional gene ontology structure for improved biological reasoning," *Bioinformatics*, vol. 22, no. 16, pp. 2020–2027, 2006.

[17] A. R. Aronson, "Effective mapping of biomedical text to the UMLS Metathesaurus: The MetaMap program," in *Proceedings of the AMIA symposium*. , 2001, p. 17.

[18] R. Baud, A.-M. Rassinoux, and J. R. Scherrer, "Natural language processing and semantical representation of medical texts," *Methods of Information in Medicine*, vol. 31, no. 2, pp. 117–125, 1992.

[19] Gene Ontology Consortium, "The gene ontology (GO) database and informatics resource," *Nucleic acids research*, vol. 32, no. suppl 1, pp. D258–D261, 2004.

[20] D. Botstein, J. Cherry, M. Ashburner, and thers, "Gene Ontology: Tool for the unification of biology," *Nature Genetics*, vol. 25, no. 1, pp. 25–29, 2000.

[21] P. Mitra, G. Wiederhold, and M. Kersten, "A graph-oriented model for articulation of ontology interdependencies," in *EDBT*, 2000, pp. 86–100.

[22] J. N. Liu, Y.-L. He, E. H. Lim, and X.-Z. Wang, "A new method for knowledge and information management domain ontology graph model," *IEEE Transactions on systems, man, and cybernetics*, vol. 43, no. 1, pp. 115–127, 2013.

[23] G. Malewicz, M. H. Austern, A. J. Bik *et al.*, "Pregel: A system for large-scale graph processing," in *SIGMOD*. ACM, 2010, pp. 135–146.

[24] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Review*, vol. 51, no. 3, pp. 455–500, 2009.

[25] J. Leskovec, D. Chakrabarti, J. Kleinberg, C. Faloutsos, and Z. Ghahramani, "Kronecker graphs: An approach to modeling networks," *J. Mach. Learn. Res.*, vol. 11, pp. 985–1042, 2010.

[26] G. Drakopoulos, "Tensor fusion of social structural and functional analytics over Neo4j," in *IISA*. IEEE, 2016.

[27] G. Drakopoulos, F. Stathopoulou, G. Tzimas, M. Paraskevas, P. Mylonas, and S. Sioutas, "A genetic algorithm for discovering linguistic communities in spatiosocial tensors with an application to trilingual Luxembourg," in *MHDW*. EANN Workshop, 2017.

[28] I. Robinson, J. Webber, and E. Eifrem, *Graph Databases*. O'Reilly, June 2013.

[29] G. Drakopoulos, A. Baroutiadi, and V. Megalooikonomou, "Higher order graph centrality metrics for Neo4j," in *IISA*. IEEE, 2015.

[30] JSON-LD specifications, "https://json-ld.org," June 2017.

[31] G. DeCandia, D. Hastorun, M. Jampani, G. Kakulapati, A. Lakshman, A. Pilchin, S. Sivasubramanian, P. Vosshall, and W. Vogels, "Dynamo: Amazon's highly available key-value store," *SIGOPS operating systems review*, vol. 41, no. 6, pp. 205–220, 2007.

[32] A. Lakshman and P. Malik, "Cassandra: A decentralized structured storage system," *SIGOPS operating systems review*, vol. 44, no. 2, pp. 35–40, 2010.

[33] JSON specifications, "www.json.org," June 2017.

[34] BSON specifications, "www.bsonspec.org," June 2017.

[35] XML definition, "https://www.w3.org/xml," June 2017.

[36] YAML specifications, "http://yaml.org," June 2017.

[37] S. Kontopoulos and G. Drakopoulos, "A space efficient scheme for graph representation," in *ICTAI*. IEEE, 2014.

[38] H. Tong, B. A. Prakash, T. Eliassi-Rad, M. Faloutsos, and C. Faloutsos, "Gelling and melting large graphs by edge manipulation," in *CIKM*. ACM, 2012, pp. 245–254.

[39] E. Estrada and D. J. Higham, "Network properties revealed through matrix functions," *SIAM Review*, vol. 52, no. 4, pp. 696–714, 2010.

[40] M. Benzi and P. Boito, "Quadrature rule-based bounds for functions of adjacency matrices," *Linear Algebra and its Applications*, vol. 433, no. 3, pp. 637–652, 2010.

[41] G. Drakopoulos, S. Kontopoulos, and C. Makris, "Eventually consistent cardinality estimation with applications in biodata mining," in *SAC*. ACM, 2016.

[42] C. Leacock, G. A. Miller, and M. Chodorow, "Using corpus statistics and WordNet relations for sense identification," *Computational Linguistics*, vol. 24, no. 1, pp. 147–165, 1998.

[43] G. Drakopoulos and V. Megalooikonomou, "On the weight sparsity of multilayer perceptrons," in *IISA*. IEEE, 2015.

[44] D. Kozen, "On the duality of dynamic algebras and Kripke models," *Logic of Programs*, pp. 1–11, 1981.

[45] C. Stirling, *Modal and temporal logics*. LFCS, Department of Computer Science, University of Edinburgh, 1991.