i-DATAQUEST : a Proposal for a Manufacturing Data Query System Based on a Graph

Lise Kim, Esma Yahia, Frédéric Segonds, Philippe Veron, Antoine Mallet

i-DATAQUEST : a Proposal for a Manufacturing Data Query System Based on a Graph

Lise Kim[1*], Esma Yahia[1], Frédéric Segonds[2], Philippe Véron[1] and Antoine Mallet[3]

[1] Arts et Metiers Institute of Technology, LISPEN, HESAM Université, Aix-en-Provence, France
[2] Arts et Metiers Institute of Technology, LCPI, HESAM Université, Paris, France
[3] CapGemini DEMS, Toulouse, France
{lise.kim*, esma.yahia, frederic.segonds, philippe.veron}@ensam.eu
antoine.mallet@capgemini.com

Abstract. During the manufacturing product life cycle, an increasing volume of data is generated and stored in distributed resources. These data are heterogeneous, explicitly and implicitly linked and they could be structured and unstructured. The rapid, exhaustive and relevant acquisition of information from this data is a major manufacturing industry issue. The key challenges, in this context, are to transform heterogeneous data into a common searchable data model, to allow semantic search, to detect implicit links between data and to rank results by relevance. To address this issue, the authors propose a query system based on a graph database. This graph is defined based on all the transformed manufacturing data. Besides, the graph is enriched by explicitly and implicitly data links. Finally, the enriched graph is queried thanks to an extended queries system defined by a knowledge graph. The authors depict a proof of concept to validate the proposal. After a partial implementation of this proof of concept, the authors obtain an acceptable result and a needed effort to improve the system response time. Finally, the authors open the topic on the subjects of right management, user profile/customization and data update.

Keywords: Graph Database · Query System · Information Retrieval · Manufacturing Data · Manufacturing Industry

1 Manufacturing Data Query

Leveraging the data product generated during its life cycle is a main issue for manufacturing industry. Manufacturing can be qualified as either discrete – i.e., the production of discrete items (e.g., cars, aircraft and appliances) – or continuous process, i.e., the production process that lends itself to an endless flow of non-discrete product (e.g., pharmaceutical, food and beverage, chemicals and cosmetics) [1, 2]. In this paper, our proposal is mainly dedicated to discrete manufacturing but could be enlarged to other domains. It requires collecting and managing data that are heterogeneous (including structured and unstructured data), explicitly and implicitly linked. Data are stored in distributed resources. One part is stored in information systems (PLM, ERP,

MES etc.) databases which are supported by different data models and the other part are mainly unstructured and semi-structured data such as target tracking in spreadsheets, note taking in a text editor, reporting written in e-mail, etc. All this data are generated by different actors, using different languages, codifications and vocabulary.

The rapid, exhaustive and relevant acquisition of information in this complex context is difficult. There is no unique point of access to get the requested information and this information is often difficult to access. The time spent searching for information represents a non-negligible part of the employee's working time estimated to 19% [3].

When considering the relational nature of manufacturing data, the authors directed the work towards a graph-database query system. This choice allows the exploitation of all data relationships when searching. Besides, the authors have chosen to prioritize the search in all the textual content of the data (which can come from metadata, text files, but also images and tables). Indeed, the authors consider that textual content is of all other types (3D, schematic, etc.) the one that contains the most useful information for different people. The use of geometric content, images, diagrams, videos or any other type of non-textual content is therefore excluded from this paper to limit the study.

The authors present the main issues to be solved and deduce their main function required for a solution in section 2. A detailed state of the art by function is presented in section 3. The authors detail in section 4 the proposal and its partial validation and conclude with a discussion in section 5.

2 Key Challenges and Associated Functions

A retrieval information system is defined by three main functions which are : data pre-processing, query pre-processing and relevance evaluation of the data according to the query. Beyond the main requirements of a retrieval information system, which are fast time response, completeness and relevance of results, retrieving information in a manufacturing company context involves many other challenges. To answer these challenges, the authors proposed to enrich the three main functions of any query system with four supplementary functions derived respectively from each identified challenge, which are:

1. *Data exploitation despite its syntactic heterogeneity:* Data can be recorded in databases or in documents as text files. In order to optimize search capabilities, the system must be able to exploit the specifications of each type of data and return, as required, either a file list, a record list from a database, data linked to another through a relational database, a metadata value, a table cell value or a specific sentence in a text content. To answer this challenge, the authors define a function that integrates each searchable element in the graph data model. This concerns the metadata of the records in the databases, the metadata of semi and unstructured files, explicit relationship between database tables, the textual content of the text files, image files and tables. This function is called: **"Function A – Process syntax specifics of data"**.

2. *Data exploitation despite vocabularies and languages heterogeneity:* Data contains textual content defined by a wide variety of actors, especially in an extended enterprise. In order to optimize the query results displayed to the user, it is necessary to return all the relevant results, even if the language or the vocabulary used in the data content differs from the language or vocabulary used in the query. A new function is then needed to extend the search for query keywords to all its semantically related terms. For example, if the term *battery* is used in the query, results must contain those with the battery translation as *batterie* in French and the *battery* synonyms as *accumulator*. This function is called: **"Function B - Search by Semantic Proximity".**

3.*Exploitation of implicit links between data:* One impact of the distributed nature of data is the lack of links between data. For example, if the user searches for the price of a part using its functional reference but the price is associated with the supplier reference in another database, the search system must be able to detect the link between the two references in order to obtain the expected result. To answer this challenge, a function is then needed to detect implicit links between data. This function is called: **"Function C - Detect implicit links between data".**

4. *Displaying results by relevance:* The ranking of results by relevance is essential to any query system that provides a significant number of results. It is therefore necessary to present the result according to a degree of relevance. The authors propose to define a system function called: **"Function D – Rank the results in relevance order".**

3 Related Works

The authors conduct a general state of the art considering several works that proposing query or analysis systems based on graph databases to deal with distributed and heterogeneous data source issues. In **Table 1**, the authors identify some works where each function consideration is evaluated.

Table 1. Related work evaluation according to the expected functions.

References	Function A Process syntax specifics of data	Function B Search by Semantic Proximity	Function C Detect implicit links between data	Function D Rank the results in relevance order
[4]	Yes	No	Yes	No
[5]	Yes	No	No	No
[6]	No	No	Yes	No
[7]	No	Yes	Yes	Yes
[8] [9] [10] [11] [12] [13]	No	No	No	No

The treatment of syntactic heterogeneity is achieved by using a meta-model to be instantiated at each new information source in the work of [4] and by integrating as many plug-ins as necessary transformations in the work of [5]. The works described in [4] and [7] deal with the automatic detection of non-existent links mentioning the use of various analytics such as text mining and the use of Levenshtein distance. The work describes in [6] integrates the detection of links between data to be unified by rules involving the relationships and common properties between the data. Finally, the authors of [7] integrate the notion of semantic search thanks to semantic annotation via bio-ontologies and result classification according to a model specific to biology. The authors of [8-13] don't integrate any of the functions mentioned, notably because the sources of the data involved are known and limited in number, which makes processing ad hoc.

As none of these works encompasses all the manufacturing industry challenges, the authors define a specific state of the art by function. It allows deeply defining the possible orientations transcribed in the scientific literature for each specific function. This study is described in the bullet list below.

- **Function A – The treatment process of data syntactic specificity.** The transformation of relational database schema into a graph database schema is a well-proven field. For some graph database, the transformation is conducted by ETL (Extraction Transformation and Load) tools plug-in[1]. This transformation allows the relationships retrieval between tables (with foreign keys) to generate graph links. Whereas, text extraction from document is possible with parser tools. Supported by natural language processing algorithms [14], the specific sentence detection according to language specificities such as grammatical rules can be done. Besides, property extraction from documents is possible with standard computer operation. Finally, spreadsheet syntax exploitation has already been done in papers like [15] but the transformation is limited by adaptability of the different array formats.

- **Function B – Search by enabling semantic proximity.** Since the Latent Semantic Indexing proposed by the Information Retrieval field [16], the tendency is rather to exploit knowledge resources of semantic relation between terms supported by ontologies [17] or knowledge graph [18]. Two orientations are then taken, query expansion or enrichment of indexed descriptors.

- **Function C – Detect implicit links between data.** Record Linkage is the task of finding two records referring to the same entity through different sources. Two objects can be associated if they have a common key. The typographical variations problems in this key are to be considered and can be solved by the comparison of common character strings (Levenshtein distance for example [19]). The link can be accentuated according to the number of common labels, and according to the labels involved, which can be weighted based on a revelation criterion [20]. On the other hand, it is also possible to use a knowledge graph and the user browsing history to

[1] https://neo4j.com/developer/neo4j-etl/

detect link between data [21]. Besides, link prediction is an important research area in social network analysis where the existing link analysis allows to the new links prediction [22].

- **Function D – Rank the results in relevance order.** The results ranking is carried out using several criteria. In the Information Retrieval field, the vector model used the number of terms occurrences. Other methods used by web search engines in the field of Search Engine Optimization should be considered [23]. These methods use for example the popularity of the sites (are they often browsed or searched?), the page freshness (are they regularly updated?) and the pages authority (do other sites refer to them?) etc.

4 I-DATAQUEST, a Proposal for a Manufacturing Data Query System

4.1 Proposal Description

The authors define the proposal for a manufacturing data query system based on a graph in Fig. 1. The main object of the proposal is to offer to any manufacturing stakeholder the possibility to query all the company data. The back end of the proposed system contains three areas presenting the system main functions: data pre-processing, query pre-processing and relevance evaluation of the data according to the query. Each of these parts was then enriched by sub-blocks allowing to respond to the specific functions listed in section 2 and thus to the key challenges related to the study context. The details of the proposal implementation are briefly presented below.

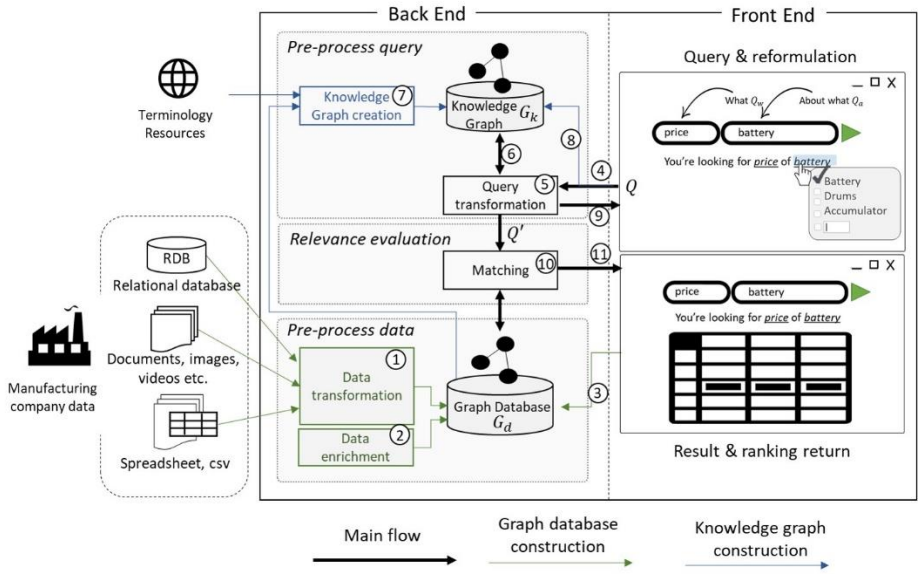

Fig. 1. I-DATAQUEST – Architecture proposal for a manufacturing data query system

1. *Heterogeneous data transformation*. As in the ETL transformation rules and the spreadsheet-specific transformation rules set out in Part 3, heterogeneous data are transformed to the unified graph data model of G_d as shown in Fig. 2. (a) The node represents a document, a relational database record or a table row. (b) The nodes have relationships between them carried by the foreign keys of relational databases. (c) The nodes carry labels with a name and value that represent properties of documents, metadata of relational databases or columns value of a table content. (d) The label 'contents' integrating the textual content of the documents is added to each node concerned. (e) The 'relevance' label including a relevance score is added to each node.

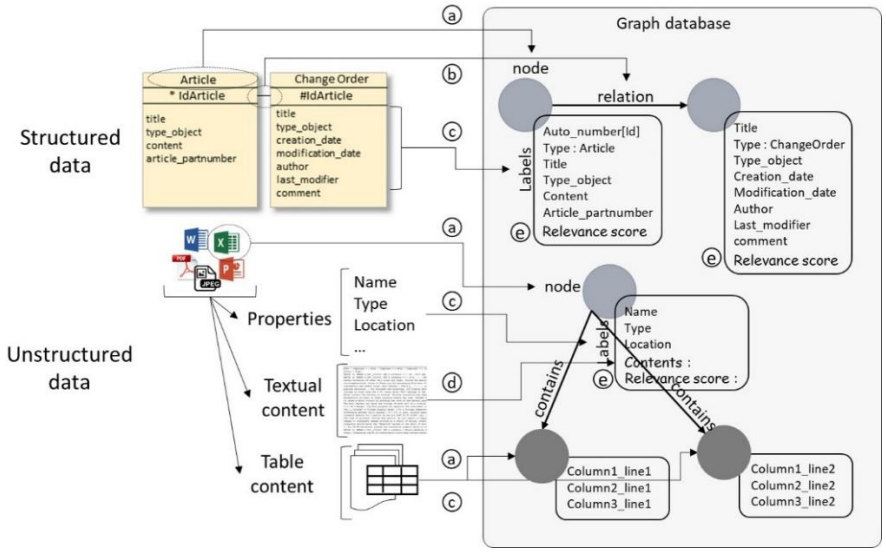

Fig. 2. Transformation of heterogeneous data to the graph data model

2. *Data enrichment*. Implicit links between nodes are added to G_d. In line with the record linkage domain, these relationships are detected if a main label value (e.g. title, reference, name etc.) of a node A is mentioned in a label of a node B. The weighting of this relationship is added according to the presence of versioning label values (e.g. version, revision, maturity etc.) of node A in the labels of node B.
3. *Relevance return*. As in popularity ranking methods for websites, when the user selects a query result, the value of the label 'relevance score' of the associated node in G_d is increased.
4. *Query Q integration*. The query Q integration is carried out by a graphical user interface represented in Fig. 1. The user expresses his need for information in two parts: $Q = Q_w \cdot Q_a$ with Q_w terms representing the information type sought ('what') and Q_a terms contained in the information sought ('about what'). In addition to the 'about what' field, the use of the 'what' field allows searching for the value of a specific node label (for example the search for the 'price' label of a node whose title is 'battery' is thus allowed when $Q_w = 'price'$ and $Q_a = 'battery'$).
5. *Transformation of the query Q to Q'*. The Q query is translated into the query language adapted to the graph database and it is adapted according to the values of Q_w and Q_a. For example, if Q_w *is not null*, Q' will integrate the query of all named or approximately named Q_w and return the values as results. In addition, Q' will query all textual content and return specific sentences based on the terms of Q and using natural language processing algorithms. For example, in a requirement search query, it is then possible to search for all sentences expressing a requirement because they contain specific verbs and modals as 'require', 'must' etc. The query is also divided into several sub-queries to evaluate the ranking of the results. For example, the sub-query querying only the labels named 'title' will return results of high relevance.

6. *Query expansion*. The terms used in query are extended to the relevant terms. The relevant terms defined with a knowledge graph G_k and according to a relevance score. This relevance score is calculated based on the frequency of the term use in previous queries. For example, in a car manufacturer context, if the user uses the term 'batterie' in French, which means both the musical instrument and the energy storage element, the English translation 'battery' and not 'drums' will by preferred because it is more often used.
7. *Knowledge graph G_k creation*. Merging both ontology and knowledge graph concepts, G_k is built using external resources such as [24] to obtain synonymous, related and translating terms used in G_d .
8. *Knowledge graph G_k enrichment*. When the user submits a query, the relevance score S_r between the query terms t_q and the terms selected in reformulation contained in G_k is increased. Conversely, when terms are deselected, the relevance score S_r is decreased.
9. *Query reformulation.* A cross-reference of the terms used in Q and all terms in relationship in G_k are proposed to the user as shown in Fig. 1. The user can select or deselect the terms to be included and thus reformulate the query.
10. *Graph database querying*. The query Q' is applied to the graph database G_d.
11. *Results displaying*. The results of the query of G_d according to Q' are displayed to the user in an order of relevance established by steps *2* and *9*.

4.2 Proposal Validation

The authors checked that necessary functions defined in section 2 are taken into account in the proposal. The authors have transcribed the result in **Table 2**.

Table 2. Functions consideration in the proposal.

Step	Query pre-processing	Data pre-processing	Relevance evaluation	Function A	Function B	Function C	Function D
1,2,4	X						
7,8,9		X					
5			X				
7				X			
3,10,11					X		
8						X	
6							X

A validation concerning the main requirements of a query system such as the fast response time, completeness and relevance of results must be conducted. This validation has been so far partially achieved. Indeed, only the first four functions in **Table 2** has been developed for the moment (including step 1,2,5,6,7) and by excluding the automation of data transformation step.

The proof of concept developed by the authors uses different tools among the Python language, Apache Tika[2] to extract the textual content of documents, Tesseract[3] for the characters recognition in images documents and Neo4J[4] as graph databases. Py2neo[5] was selected to conduct the query to Neo4j with Python language and DashByPloty[6] generates a web interface suitable for data analysis. Fig. 3 shows the different tools used for each layer of the proposition.

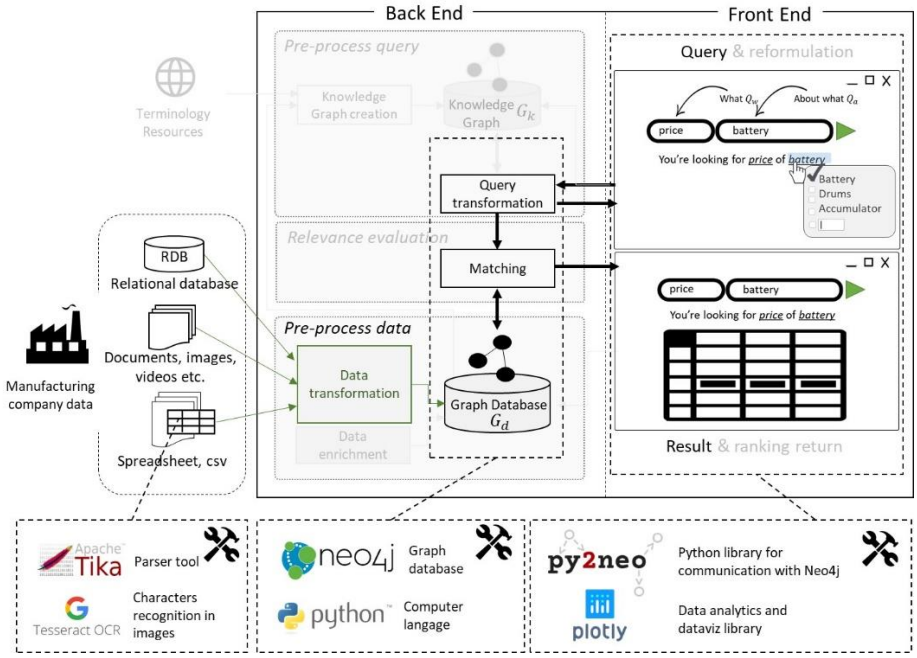

Fig. 3. Proof of concepts tools

The dataset used to validate the author's approach, comes from a digital model of a drone project and enriched. It contains 686 elements, 47% of which are unstructured data, 22% tree structure data and 17% from a relational database. The authors identified 19 queries to meet the requirements of innovative uses in the manufacturing company and from its various departments. The expected answers per query were evaluated manually by browsing through each object in the dataset. For time measurement, the aim is to achieve less than one second between the submission of the query by the user and the display of the result according to [25].

The measures used to evaluate the Information Retrieval system are the recall (number of relevant documents found according to the number of relevant documents in the

[2] https://tika.apache.org
[3] https://opensource.google/projects/tesseract
[4] https://neo4j.com
[5] https://py2neo.org/v4
[6] https://plot.ly/dash/

database) and the precision (number of relevant documents found according to the total number of documents display in result). A system returning all relevant results without displaying irrelevant results will have these two measurements equal to 1.

The results obtained are in **Table 3**. The results are compared to a first step where only node and labels without processing of texts and tables content, have been integrated in the graph database.

Table 3. Partial evaluation of the proposal

Stage	Response time (s)	Precision	Recall
First step : Only metadata	25	1	0.01
Second step : With function A	79	0.60	0.45

Evaluation of the results shows a degraded query time. An important waste of time is due to the search for sentences in the textual content. There are ways to improve time such as the search for sentences on demand only but also by the integration of text content descriptors rather than the entire content, the reduced selection of labels to be used in a search, and the table content extracts on demand. It will therefore be necessary to evaluate each option according to the degradation of the other requirement values. However, the recall measurement improves and precision decreases with the increase in the number of results.

5 Perspectives

In order to address the issue of information retrieval in the context of manufacturing industry, the authors have highlighted the key challenges related to the manufacturing context and specifies the functions to be developed. These challenges have been defined according to the nature of the manufacturing data. These characteristics may be common to other domains, so it will be interesting to study the applicability of the solution to other domains in the future. A graph-based query system proposal was presented in this paper with three main functions and four specific functions. A proof of concept has been conducted for 4 of the 7 functions. This proposal responds to key challenges by first transforming the syntactic specificities of the data into a graphical data model, by extending the query terms to semantically close terms, by detecting and transforming the implicit links between the data into explicit links, and finally by classifying the results according to their relevance. It is therefore necessary to develop the integration of the remaining functions in order to complete the evaluation of the proposal. In addition to this work, some topics can be discussed.

The user rights management is an important security issue. Indeed, unifying multiple data sources, the system integrates a variety of heterogeneous rights matrices, particularly in the extended enterprise context. The main orientation is then the creation of a specific rights management to the query system. This right management required the same complexity as the existing one, which is a greatest remaining challenge.

Beyond the management of user rights, the user profile information (who, what profession, with what history of use) is an important asset to improve the results relevance. One of the questions is whether knowledge graphs by profile or user is needed. If so, its feasibility will have to be assessed while still meeting the requirements.

Updating source data is also a feature to be considered in this proposal. This paper does not present the automation of data extraction from their sources but one orientation was taken in the creation and the update by batch. So the features to be considered are the creation, modification and deletion of nodes and relationships.

Finally, the authors wish to consolidate the detection of implicit links between data. This allows a better follow-up in change management as presented in [26] with graphs and ontology.

References

1. International Organization for Standardisation (ISO): ISO 15531-1:2004 Industrial Automation Systems and Integration – Industrial Manufacturing Management Data – Part 1: General Overview, Geneva (2004).
2. Pinquié, R., Rivest, L., Segonds F., Philippe V.: An illustrated glossary of ambiguous PLM terms used in discrete manufacturing. International Journal of Product Lifecycle Management, vol. 8, no. 12, pp. 142-171, (2015). doi: 10.1504/IJPLM.2015.070580
3. Chui, M., Manyika, J., Bughin, J., Dobbs, R., Roxburgh, C., Sarrazin, H., Sands, G., Westergren, M.: The social economy: Unlocking value and productivity through social technologies. McKinseyGlobal Institute (2012). (Access Online) http://www.mckinsey.com/industries/high-tech/our-insights/the-social-economy 2020/03/27
4. Gröger, C., Schwarz, H., Bernhard, M.: The Deep Data Warehouse: Link-Based Integration and Enrichment of Warehouse Data and Unstructured Content. In: 2014 IEEE 18th International Enterprise Distributed Object Computing Conference (2014). doi: 10.1109/EDOC.2014.36
5. Martínez-Bazan, N., Muntés-Mulero, V., Gómez-Villamor, S., Nin, J., Sánchez-Martínez, M.-A., Larriba-Pey, J.-L.: DEX: High-performance exploration on large graphs for information retrieval. In : International Conference on Information and Knowledge Management, Proceedings (2007). doi: 10.1145/1321440.1321521
6. Touré, V., Mazein, A., Waltemath, D., Balaur, I., Saqi, M., Henkel R., Pellet J., Auffray C.: STON: Exploring biological pathways using the SBGN standard and graph databases. BMC Bioinformatics, vol. 17, no. 1, (2016). doi: 10.1186/s12859-016-1394-x
7. Henkel, R., Wolkenhauer, O., Waltermath, D.: Combining computational models, semantic annotations and simulation experiments in a graph database. Database, pp. 1-16 (2015). doi: 10.1093/database/bau130
8. Lysenko, A., Roznovăţ, I. A., Saqi, M., Mazein, A., Rawlings, C. J.: Representing and querying disease networks using graph databases. BioData Mining, vol. 9, no. 11, p. 23 (2016). doi: 10.1186/s13040-016-0102-8
9. Yoon, B.-H., Kim, S.-K., Kim, S.-Y.: Use of Graph Database for the Integration of Heterogeneous Biological Data. Genomics & informatics, vol. 15, pp. 19-27 (2017). doi: 10.5808/gi.2017.15.1.19
10. Bonnici, V., Russo, F., Bombieri, N., Pulvirenti, A., Giugno, R.: Comprehensive reconstruction and visualization of non-coding regulatory networks in human. Frontiers in Bioengineering and Biotechnology, vol. 2, p. 69 (2014). doi: 10.3389/fbioe.2014.00069

11. Messina, A., Fiannaca, A., La Paglia, L., La Rosa, M., Urso, A.: BioGraph: a web application and a graph database for querying and analyzing bioinformatics resources. BMC Systems Biology, vol. 5, no. 112, pp. 75-89 (2018). doi: 10.1186/s12918-018-0616-4
12. Noel, S., Harley, E., Tam, K. H., Gyor, G.: Big-data architecture for cyber attack graphs representing security relationships in nosql graph databases. In : IEEE Symposium on Technologies for Homeland, Boston (2015).
13. Schabus, S., Scholz, J.: Spatially-Linked Manufacturing Data to Support Data Analysis. Journal for Geographic Information Science, vol. 15, no. 11, pp. 126-140 (2017). doi: 10.1553/giscience2017_01_s126
14. Chowdhury, G. G.: Natural Language Processing, Annual Review of Information Science and Technology, pp. 51-89 (2003). doi: 10.1002/aris.1440370103
15. Zhu, Z., Zhou, X., Shao, K.: A novel approach based on Neo4j for multi-constrained flexible job shop scheduling problem. Computers and Industrial Engineering, vol. 130, pp. 671-686 (2019). doi: 10.1016/j.cie.2019.03.022
16. Dumais, S. T.: Latent semantic indexing (LSI) and TREC-2. In The Second Text Retrieval (TREC2), pp. 105-116 (1994).
17. Kyriakakis, A., Koumakis, L., Kanterakis, A., Iatraki, G., Tsiknakis, M.: Enabling Ontology-based Search: A Case study in the Bioinformatics Domain. In: IEEE 19th International Conference on Bioinformatics and Bioengineering (BIBE) Enabling (2019). doi: 10.1109/bibe.2019.00048
18. Xiong, C., Power, R., Callan, J.: Explicit Semantic Ranking for Academic Search via Knowledge Graph Embedding. In: 26th International World Wide Web Conference, WWW, pp. 1271-1279 (2017). doi: 10.1145/3038912.3052558
19. Navarro, G.: A Guided Tour to Approximate String Matching. ACM Computing Surveys, vol. 33, no. 11, pp. 31-88 (2001). doi: 10.1145/375360.375365
20. Fellegi, I. P., Sunter, A. B.: A theory for Record Linkage. Journal of the American Statistical Association, vol. 64, no. 1328, pp. 1183-1210 (1969). doi: 10.1080/01621459.1969.10501049
21. Hakkani-Tür, D., Heck, L., Tur, G.: Using a knowledge graph and query click logs for unsupervised learning of relation detection. In: ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing – Proceedings (2013). doi: 10.1109/icassp.2013.6639289
22. Cukierski, W., Hammer, B., Yang, B.: Graph-based Features for Supervised Link Prediction. In Proceedings of International Joint Conference on Neural Networks, San Jose, California, USA (2011). doi: 10.1109/ijcnn.2011.6033365
23. Xing, W., Ghorbani, A.: Weighted PageRank algorithm. In Proceedings. Second Annual Conference on Communication Networks and Services Research (2004). doi: 10.1109/dnsr.2004.1344743
24. Speer, R., Chin, J., Havasi, C.: ConceptNet 5.5: An Open Multilingual Graph of General Knowledge. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17) (2017).
25. Arapakis, I., Bai, X., Cambazoglu, B.: Impact of Response Latency on User Behavior in Web Search. SIGIR 2014 - Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 103-112 (2014). doi: 10.1145/2600428.2609627
26. Mordinyi, R., Shindler, P., Biffl, S.: Evaluation of NoSQL graph databases for querying and versioning of engineering data in multi-disciplinary engineering environments. IEEE International Conference on Emerging Technologies and Factory Automation, ETFA (2015). doi: 10.1109/etfa.2015.7301486