Noname manuscript No.
(will be inserted by the editor)

Key issues for a manufacturing data query system based on graph

Lise KIM, Esma YAHIA, Frédéric SEGONDS, Philippe VÉRON, Victor FAU

- · · · · · · · · -

Received: date / Accepted: date

Abstract Manufacturing industry data are distributed, heterogeneous and numerous, resulting in different challenges including fast, exhaustive and relevant querying of data. In order to provide an innovative answer to this challenge, the authors consider an information retrieval system based on a graph database. In this paper, the authors focus on determining the key issues to consider in this context. The authors define a three-step methodology using root causes analysis. This methodology is then applied to a data set and queries representative of an industrial use case. As a result, the authors list four main issues: (i) semantic extension of keyword search, (ii) the treatment of syntactic heterogeneity contained in unstructured data, (iii) the results treatment by relevance order and (iv) the detection of relationships between a priori unrelated data. The authors conclude by discussing potential resolutions of these four issues, suggest adapting the methodology used in the paper to evaluate a future proposal, and finally open the possibility of using the results beyond the manufacturing domain.

Keywords Manufacturing data · Information retrieval · Graph database · Query system · Heterogeneous data

1 Introduction

The collection of information across the product lifecycle benefits interactive engineering [1] and manufacturing [2]. The volume of data generated by the manufacturing industry is large and increasing; it represents 3.6 exabytes in 2018 and will increase by 30% in 2025 [3]. The acquisition of sensor data in real time, key tools of the fundamental concepts of Industry 4.0 [4], reinforce this volume. Justified by the need for specialisation of the different businesses, the organisation of companies in silos generates data that is both distributed and heterogeneous. According to the definition of structured and unstructured data given by Kassner et al. [5], part of the data is managed by information systems (PDM[1] , ERP[2], MES[3] . . .) and generate structured data, while the other data are unstructured (text, image, 3D . . .). Moreover, the data can be (i) explicitly linked together as in the parent-child relationships of a digital mock-up or (ii) implicitly linked as between the 3D of a component stored in a database and the user manual of this component stored in another database.

To perform their work, employees have to query the data to retrieve the necessary information. This task becomes complicated and time consuming due to the increasing volume of heterogeneous data stored in distributed resources. The time spent by an employee searching for information has been estimated at 16% of their working time in [6]. To solve these issues, it is necessary to define a data query system that provides exhaustive and relevant data as fast as possible.

To address this challenge, the authors have worked to draw up the list of bare minimum issues to be considered to define the optimal framework. Indeed, until now, different works have listed the issues related to enterprise search based on literature [7] or on employee interviews [8]. The authors, after having chosen an orientation particularly adapted to the context, propose in this paper an experimental process to obtain this list which they then apply to the case of manufacturing

[1] PDM for Product Data Management
[2] ERP for Enterprise Resource Planning
[3] MES for Manufacturing Execution System

data. Thus, this paper is organised as follows: sect. 2 defines the main orientations chosen based on a state-of-the-art analysis. Sect. 3 describes the methodology used to draw up the list of issues. Sect. 4 describes the experimental conditions and sect. 5 presents the results. Sect. 6 discusses the results and sect. 7 concludes and presents prospects for future work.

2 Graph database consideration

Querying information can be achieved through Information Retrieval Systems. These systems must access the data in order to provide the most relevant one. This is achieved by managing the data in NoSQL databases rather than traditional relational databases, first being faster, more efficient and flexible [9]. The main categories of NoSQL databases like column database, key-value store and document-oriented database include indexing and fast access to the information but lack expressing of the relationships between data in their schema. Graph databases address this issue and are therefore the most appropriate in our context. To emphasise the benefit of the graph database, different researches like [10] have shown the importance of analysing data with a strong relational nature.

In addition, many works already propose graph-based systems to represent and access heterogeneous and related data. For example, the authors of [11] apply the graph to represent the spatial and non-spatial manufacturing data of a semiconductor production line in order to facilitate data exchange and analysis. The authors of [12] apply the graph to represent the various elements related to cyber-security in order to facilitate the analysis. The authors of [13–16] apply the graph to represent biological information networks, always with a view to analysing and exploring information. Finally, the authors of [17] present and evaluate a querying system named 'DEX'. This system exploits an entire network composed of heterogeneous data. On the other hand, studies focus on particular capabilities. For example, the authors of [18] propose a graph linking the different versions of engineering data using ontologies. The authors of [19] provide an additional graph to the data warehouse to link data together. The authors of [20] propose the integration of semantic annotations to improve query capabilities. Finally, the authors of [21] propose the detection of relations between data in the design and manufacturing domains.

In conclusion, the graph approach is considered in recent multi-domain studies. The authors of [12–16] show the particular interest of using this approach with heterogeneous and related data, characteristics similar to the data of manufacturing industry. The authors of [17] propose a relatively complete system as much on the heterogeneity of the considered data as on the possibilities of requests. Nevertheless, this system does not integrate specific functionalities handled by other studies such as the detection of links between data [18,19,21] or semantic considerations [20]. Therefore, in this paper, the authors aim to answer the following question : 'What are the minimum issues to be considered for a manufacturing data query system based on a graph database?'.

3 Methodology

In order to define only the bare minimum issues to consider when defining the query system, an iterative process has been implemented. This process is represented in Fig. 1 with the IDEF0 representation method [22].

3.1 Construct the datagraph

The step *construct the datagraph* begins with a data recovery phase. These data are syntactically varied and are then encoded to be schematised into a graph data model. This graph data model contains n nodes and r relationships between nodes. The nodes have properties p_1 ... p_i whose values are expressed by v_{p_i} where i is the number of properties. The nodes and relationships are labelled with l_n and l_r respectively in order to classify them, for example by typology. In order to avoid unnecessary complexity, the transformation rules during the first iteration of the methodology must be simple. Thus, as shown in Fig. 2, two types of transformation are presented according to the type of source data considered :

Transformation of structured data. Structured data is stored in various relational databases. Each data is then represented as a tuple in a table, this tuple is associated with attributes and their values. Each tuple can be associated with other tuples in other tables using a foreign key. All tuples are then represented as a node n whose properties p_1 to p_i are filled in from the attributes of the tuple. The foreign key relationships are represented by a relationship between the nodes. The label of the node corresponds to the name of the table and the label of relationship is by default :in_relation_with.

Transformation of unstructured data. Unstructured data is generated by different types of software and stored on different servers within the company. Each file contains metadata such as 'type', 'name' and so on. All files are then represented in nodes n and their properties p_1 to

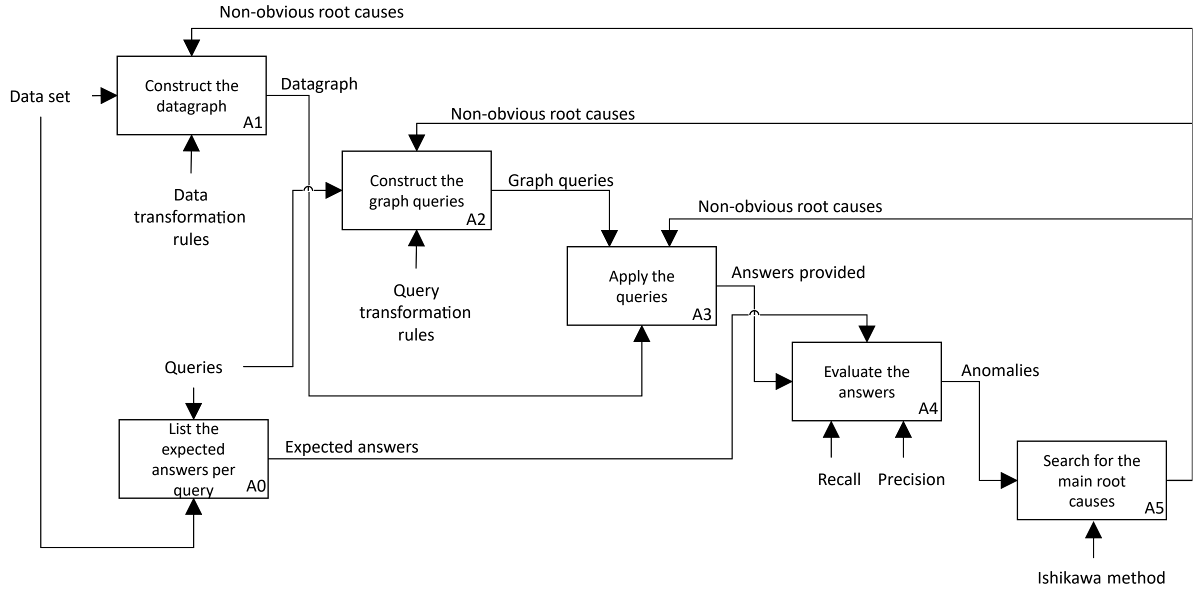
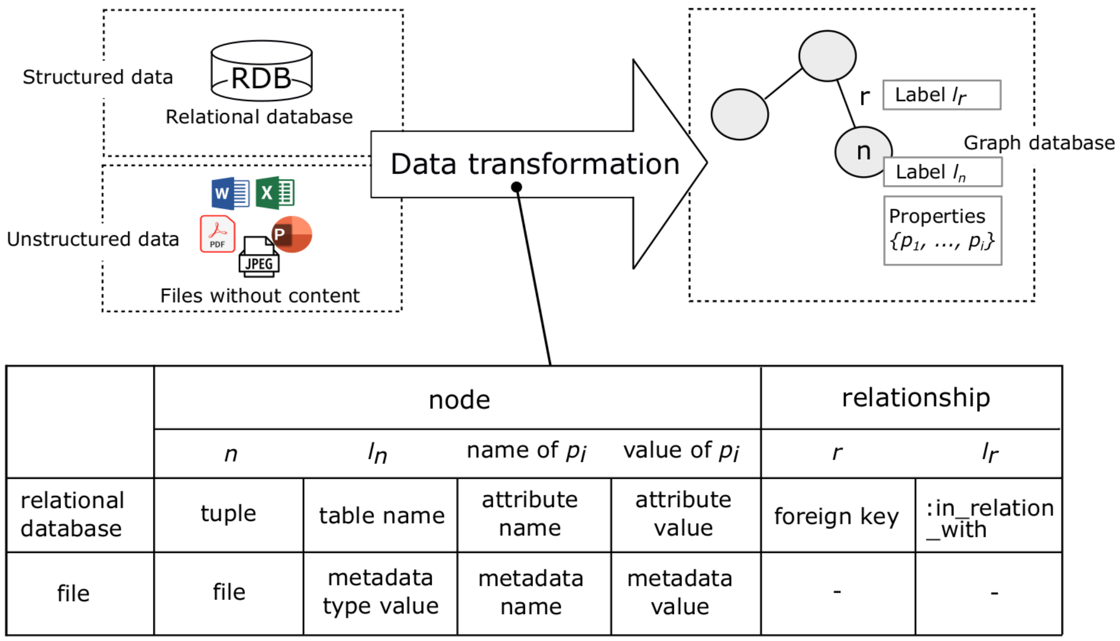


Fig. 1 Process to detect the key issues to be considered in a query system



	node				relationship	
	n	l_n	name of p_i	value of p_i	*r*	l_r
relational database	tuple	table name	attribute name	attribute value	foreign key	:in_relation _with
file	file	metadata type value	metadata name	metadata value	-	-

Fig. 2 Data transformation rules from structured and unstructured data into a graph data model

p_i are the representation of the metadata of the file. The content (text of a word, image of a jpeg ...) is not considered at the first iteration of the methodology.

Thus, the resulting graph G_d, defined by the equation 1, consists of nodes n and relations r. The nodes are defined by labels l_n and the properties p_i with values v_{p_i}. Relations are between two nodes n and m with the label l_r.

$$G_d = (n, r) \leftrightarrow (: l_n\{p_1 : v_{p_1}, ..., p_i : v_{p_i}\}, (n){-}[: l_r] \rightarrow (m)) \quad (1)$$

3.2 Construct the graph queries

User queries can be formulated in different ways: natural language, use of keyword(s) or advanced searches with operators such as 'OR', 'AND' and the selection of specific attributes of the database. Whatever the formulation, the queries must be transformed to browse the graph and find the answer. For completeness of representation of the manufacturing industry's requests, the following three types of queries were considered:

- Search for a list of data (file or tuple) to answer questions such as `which documents ...?`
- Search for a property's value (attribute value or metadata value) to answer questions such as `what is the property of ...?`
- Search for a sentence (contained in an attribute or metadata) to answer questions such as `what is the price or requirement of ...?`

In order to transform these three types of searches into graphs, the following rules have been applied:

The search for a list of data associated with one or more keywords is transformed into a search for a list of nodes containing the keyword(s) in their properties, e.g.: `query = battery` searches all nodes where at least one property contains the term `battery`.

The search for a property value associated with one or more keywords is transformed into a search for the property value of nodes containing the keyword(s) and nodes directly related to a node containing the keyword(s), e.g.: `query = price of batteries` searches all the nodes containing the term `battery` and returns the property value named `price` of these nodes and their related nodes.

The search for a sentence , associated with at least two keywords, becomes a search for all sentences containing both keywords in all nodes, e.g.: `query = price of battery` searches for all sentences containing the term `price` and the term `battery`. A specific case has been added to search for a requirement associated with one or more keywords. e.g.: `query = requirements of batteries` becomes a search for all sentences containing verbs or modals expressing the requirement (requires, must etc.) and the term battery. Natural Language Processing (NLP) [23] tools will be used here to find the sentences.

3.3 Application of queries, evaluation of results and search for root causes

In order to obtain a data query system that provides complete and relevant results as fast as possible, the evaluation of the proposal is based on three requirements: the response time between the submission of the query and the display of the result, the completeness of the result using the recall[4] measure and the relevance using the precision[5] measure. These are calculated based on the expected results. The expected results are defined manually, ideally submitted to various user profiles with different prior knowledge of the dataset.

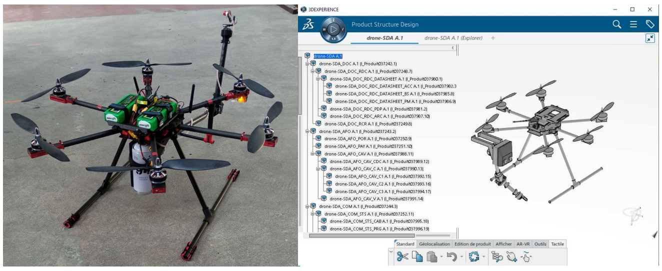
For each of the three requirements, the acceptable limits are defined. When the results are below the accepted limits, the analysis of each error is then performed (excess or missing data) in order to detect the root causes. This root cause analysis is based on the Ishikawa diagram method[6] [24]. Each root cause is scored according to its impact on the results. This score is calculated by dividing the number of errors associated with that root cause by the total number of errors. Once this list of root causes has been classified, it helps to define the main issues to be addressed.

3.4 iteration

The choice to initialise the method with simple data and query transformations can lead to first cycles where the issues are simple to solve. Therefore, the method should be repeated until a list of non-obvious issues is obtained.

4 Experimentation conditions

The expected performance thresholds are less than 1 second for time; this was set according to the findings of a study on the impact of response latency in web search [25]. Precision and recall should be strictly equal to 1 to ensure that all expected results are given and that no false results are given.



drone-SDA A.1
drone-SDA_DOC A.1 (I_Produit037242.1)
drone-SDA_DOC_RDC A.1 (I_Produit037248.7)
drone-SDA_DOC_RDC_DATASHEET A.1 (I_Produit037980.1)
drone-SDA_DOC_RDC_DATASHEET_ACC A.1 (I_Produit037982.3
drone-SDA_DOC_RDC_DATASHEET_BS A.1 (I_Produit037985.8)
drone-SDA_DOC_RDC_DATASHEET_PM A.1 (I_Produit037986.9)
drone-SDA_DOC_RDC_PDP A.1 (I_Produit037981.2)
drone-SDA_DOC_RDC_ARC A.1 (I_Produit037987.10)
drone-SDA_DOC_RCR A.1 (I_Produit037249.8)
drone-SDA_AFO A.1 (I_Produit037243.2)
drone-SDA_AFO_POR A.1 (I_Produit037250.9)
drone-SDA_AFO_PAY A.1 (I_Produit037251.10)
drone-SDA_AFO_CAV A.1 (I_Produit037988.11)
drone-SDA_AFO_CAV_CDC A.1 (I_Produit037989.12)
drone-SDA_AFO_CAV_C A.1 (I_Produit037990.13)
drone-SDA_AFO_CAV_C1 A.1 (I_Produit037992.15)
drone-SDA_AFO_CAV_C2 A.1 (I_Produit037993.16)
drone-SDA_AFO_CAV_C3 A.1 (I_Produit037994.17)
drone-SDA_AFO_CAV_V A.1 (I_Produit037991.14)
drone-SDA_COM A.1 (I_Produit037244.3)
drone-SDA_COM_STS A.1 (I_Produit037252.11)
drone-SDA_COM_STS_CAB A.1 (I_Produit037995.18)
drone-SDA_COM_STS_PRG A.1 (I_Produit037996.19)

Fig. 3 Drone and its digital mock-up used as data set

4.1 Dataset

The study was based on a dataset composed of 686 elements, representing the data of a drone manufacturing company whose digital mock-up can be seen in Fig. 3. The dataset is distributed as following:

- 47% unstructured data including spreadsheets, videos, photos and textual documents
- 21% tree structure data
- 17% of data from relational databases
- 15% of geometrical data.

All these elements represent the data needed to develop a mechanical system (from design to prototyping through logistics, purchasing and project management).

4.2 Queries

18 queries were written in response to 10 innovative use cases characterised by PLM[7] and digital manufacturing business group managers at Capgemini[8]

These use cases are multi-profiled and multi-activity, aiming to cover all phases of product lifecycle management. Examples include a designer looking to identify the requirements or justification for a product, or a salesperson looking for a customer's usage parameters, or a manager looking to identify an available team with the right skills. The list of innovative use cases is presented in Table 1 and the list of queries in Table 2.

In regard to the dataset used and described in sect. 4.1, the valuation of the different variables of the table 2 are : *keyword1* = battery ; *keyword2* = hood ; *keyword3* = engine; *keyword4* = blade additive manufacturing ; *keyword5* = recruitment; *keyword6* = filter cleaning ; *keyword7* = drone ; *skill1* = drone ; *skill2* = additive manufacturing ; *parameter1* = 14.8V ; *designation* = 4S5200 ; *employee* = Frederic Segonds ; *customer* = Serge Bernard; *parameter2* = speed

Table 1 List of use cases

Use Case ID	Use Case description
UC1	Identify a system requirement
UC2	Identify the existing products for renewal
UC3	Compare two products according a criteria
UC4	Access justifications for product design choices
UC5	Detecting innovations that respond to a function
UC6	Visualise the process, standard or methodology to apply
UC7	Detecting suppliers with specific skills
UC8	Identify a team available with the desired skills
UC9	Predicting the physical and psychosocial risks of the business
UC10	Propose a personalised configuration to the customer

Table 2 List of queries

Use Case ID	Query ID	Query
UC1	Q1	Finds all sentences expressing a requirement and containing the term *keyword1*
UC2	Q2	Finds all objects mentioning the term *keyword1*
UC2	Q3	Find the references of *keyword1*
UC2	Q4	Find references for *keyword1* and *parameter1*
UC2	Q5	Finds simulation data related to *keyword2*
UC3	Q6	Find the prices of *designation*
UC3	Q7	Find comments mentioning *keyword1*
UC4	Q8	Find all choice justification for *keyword3*
UC5	Q9	Find all patent mentioning *keyword4*
UC6	Q10	Find all process mentioning *keyword5*
UC6	Q11	Find all standard mentioning *keyword6*
UC6	Q12	Find all methodology mentioning *keyword7*
UC7	Q13	Find all suppliers mentioning *skill1*
UC7	Q14	Find all employees mentioning *skill2*
UC8	Q15	Finds all schedule mentioning *employee*
UC9	Q16	Finds symptom and associated job
UC10	Q17	Find configuration files of *customer*
UC10	Q18	Find the *parameter2* related to the *customer*

[4] The recall is defined by the number of relevant documents found with regard to the number of relevant documents in the database

[5] The precision is the number of relevant documents found compared to the total number of documents proposed in the result

[6] Method of analysis used to search for and to represent the different possible causes of a problem

[7] PLM for Product Lifecycle Management

[8] Company of digital services in the manufacturing industry - https://www.capgemini.com/

Table 3 Results of the first cycle

	Cycle 1
Precision [0,1]	0.50
Recall [0,1]	0.01
Response time (s)	5790

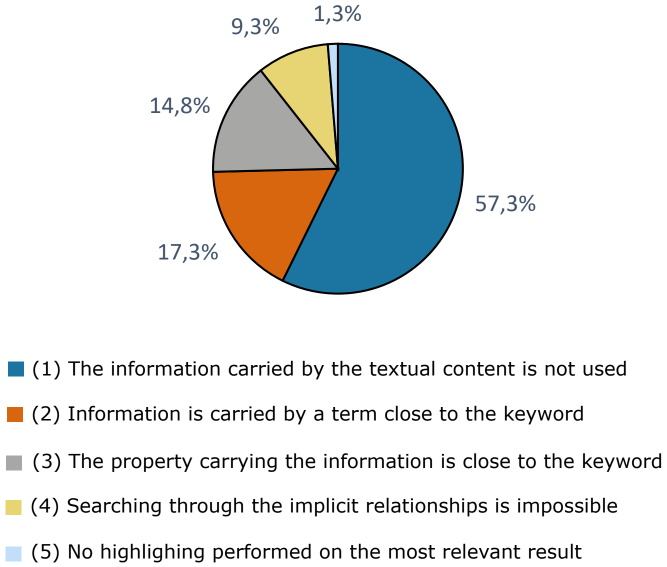


Fig. 4 Root causes of the first cycle and their distribution

4.3 Implementation

The data graph and its querying is supported by Neo4J[9], an open source solution used in many other works of information retrieval from heterogeneous and distributed data, as well as in the field of biology [13], as in the manufacturing industry [11] or in the field of cyber-security [12]. The exploitation of natural language in texts is realised with the StandfordNLP[10] algorithms, a solution allowing a powerful recognition of entity names [26] accessible with a well-documented toolkit [27] and used in many information extraction works [28–30]. The programming language for query transformation is python[11]. The py2neo[12] library has enabled communication between the python language and Neo4J.

5 Results

5.1 Application of the first cycle

The first cycle results of the methodology applied to the dataset and queries described in sect. 4.1 and sect. 4.2 respectively are visible in Table 3. These results indicate that too few expected results are displayed. Indeed, the recall value shows that on average only 1% of the expected results are displayed. The precision value indicates that 50% of the displayed results are not part of the expected results.

Table 4 Results of the second cycle

	Cycle 2
Precision [0,1]	0.44
Recall [0,1]	0.31
Response time (s)	16978

The classification of each anomaly into a list of root causes is listed in Fig. 4. This list shows that more than half of the anomalies are caused by the lack of textual content of the data in the graph database. For example, the search for the battery reference (Q3) does not give any result because the information is carried by the content of an excel named *Bill of Materials*. Another example is the lack of results in the search for suppliers associated with specific skills (Q13) because these skills are mentioned in the textual content of files.

In order to address this issue, a second cycle was therefore launched to integrate the textual content of unstructured data.

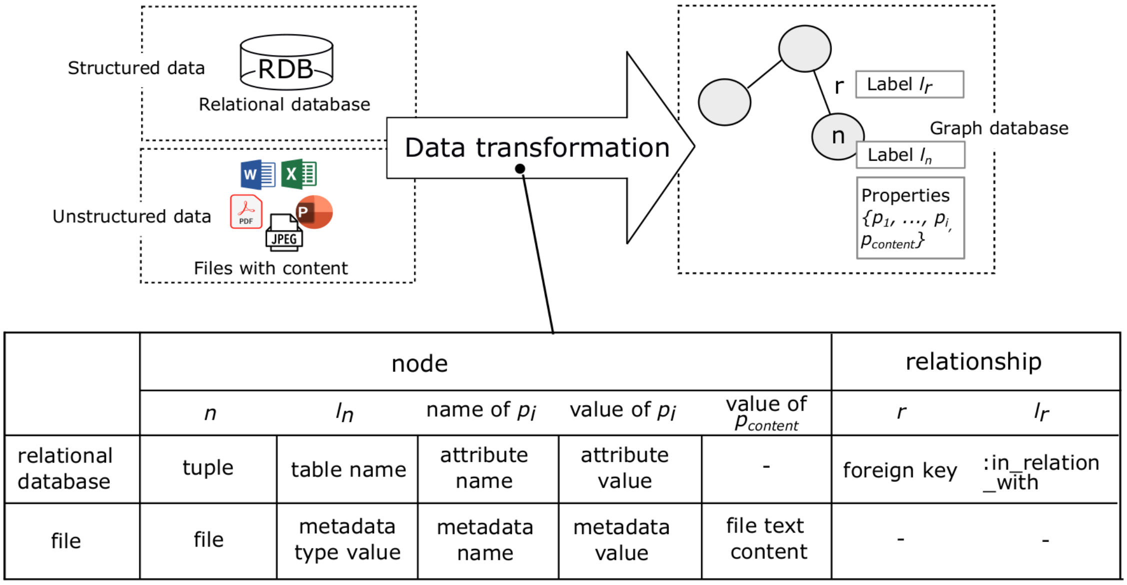
Concerning the average response time, the score is over one hour. Solutions must be proposed to optimise the response time but this topic is not prioritised compared to the previous one. Indeed, the resolution of the previous topic will have an impact on the response time and on the potential leads to follow.

5.2 Application with textual content

The text contained in the documents is extracted using Apache Tika[13]. This tool is a standard and open source parser used in many other heterogeneous document processing works [31–33]. The text contained in an image is extracted using Tesseract[14]. This is a standard Optical Character Recognition Tool used in many other information retrieval studies as [34–36]. The defects highlighted by the study [37] requiring prior training will have been limited by a prior filtering of the scanned files according to their qualities.

The results of this second cycle indicate a clear improvement in the presence of the expected results. Indeed, as shown in Table 4 and the fig. 6, recall has increased by 30 points while precision indicates that on average only 44% of the results displayed are good.

[9] https://neo4j.com
[10] https://standfordnlp.github.io/CoreNLP/
[11] https://www.python.org/
[12] https://py2neo.org/2020.0/
[13] https://tika.apache.org/
[14] https://opensource.google/projects/tesseract



	node					relationship	
	n	l_n	name of p_i	value of p_i	value of $p_{content}$	r	l_r
relational database	tuple	table name	attribute name	attribute value	-	foreign key	:in_relation _with
file	file	metadata type value	metadata name	metadata value	file text content	-	-

Fig. 5 Data transformation rules from structured and unstructured data with their textual content

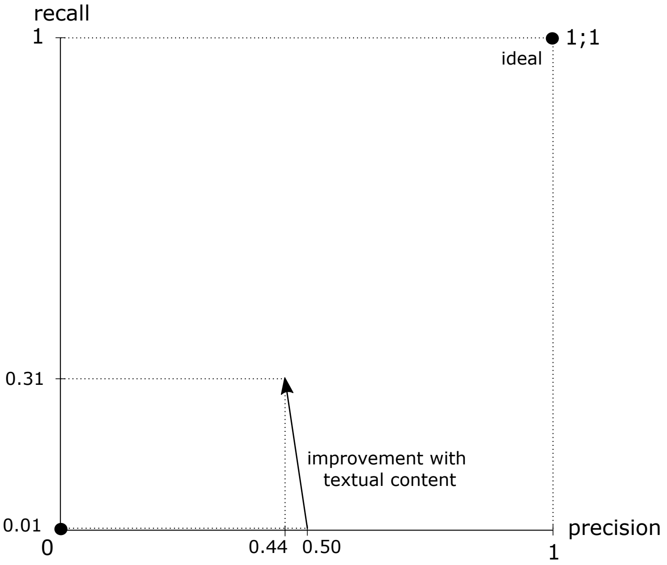


Fig. 6 Improvement and gap with optimum

The analysis and classification of each anomaly divide them into 7 different root causes listed in Fig. 7. The only cause *(6) The OCR algorithm didn't extract the correct characters* is due to insufficient performance of an existing element in the initial architecture. It is then possible to remove this cause from the list of the bare minimum issues to solve.

Also for this cycle, the average response time is over four hours. Solutions should be proposed to optimise the response time but this topic is not prioritised.

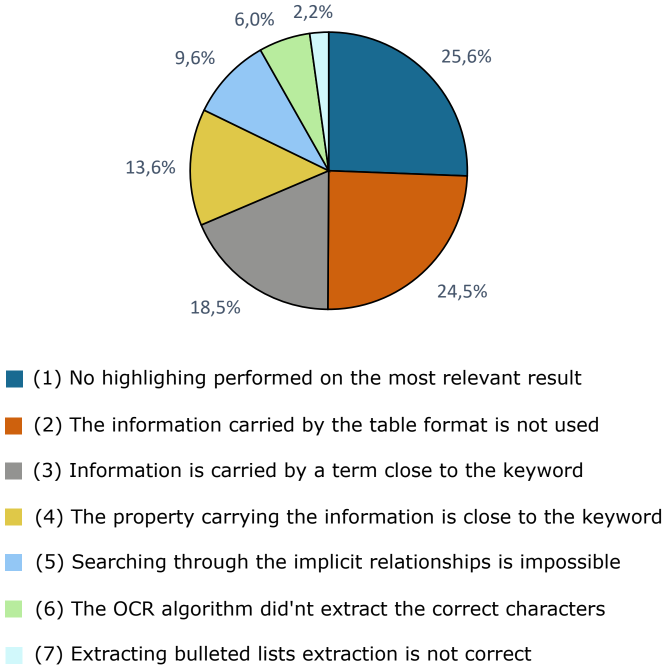


Fig. 7 Root causes of the second cycle and their distribution

5.3 The four key issues to consider

After adding the textual content of the data to the initial system, 6 root causes remain. The authors propose to classify them into 4 main families and in decreasing order of percentages :

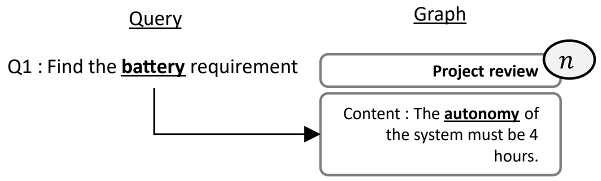
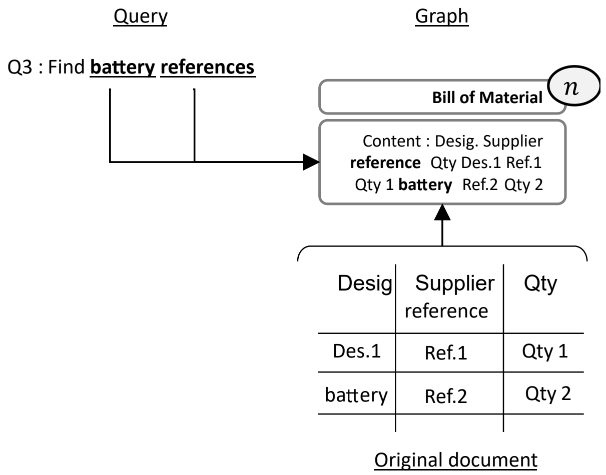

Query
Graph
Q1 : Find the battery requirement
Project review
n
Content : The autonomy of the system must be 4 hours.

Fig. 8 Illustration of the root cause (3)


Query
Graph
Q3 : Find battery references
Bill of Material
n
Content : Desig. Supplier reference Qty Des.1 Ref.1 Qty 1 battery Ref.2 Qty 2
Original document

Desig	Supplier reference	Qty
Des.1	Ref.1	Qty 1
battery	Ref.2	Qty 2

Fig. 9 Illustration of the root cause (2)

Semantically extending the search keywords - 32.1%. Causes (3) and (4) indicate that searching for an exact keyword or property is not enough and that reconciliation between different terms is necessary. As illustrated in Fig. 8, the search for sentences expressing requirements related to the *battery* should also take into account the term *autonomy*. In the case of properties, if the term *reference* is used in the query, the term *Part Number* must also be searched.

The treatment of syntactic heterogeneity contained in unstructured data - 26.7%. Indeed extracting text without format is not enough. Cause (2), illustrated in Fig. 9, indicates that it is necessary to translate the information carried by the table format (rows and columns) in order to use it in query. For example, to detect a reference contained in a specific cell of a bill of materials. Cause (7) indicates that bulleted lists processing is necessary for the performance of the chosen NLP tools. The table format and bulleted lists must be transformed to be used.

The treatment of the results by order of relevance - 25.6%. Indeed, there is no order by relevance in the results, and cause (1), illustrated in Fig. 10, indicates that unexpected results (but potentially relevant) are displayed in the same way as expected results. For example, searching for the *battery reference* provides many results with the terms *reference* and *battery* in the content, but these results are far from the information sought.

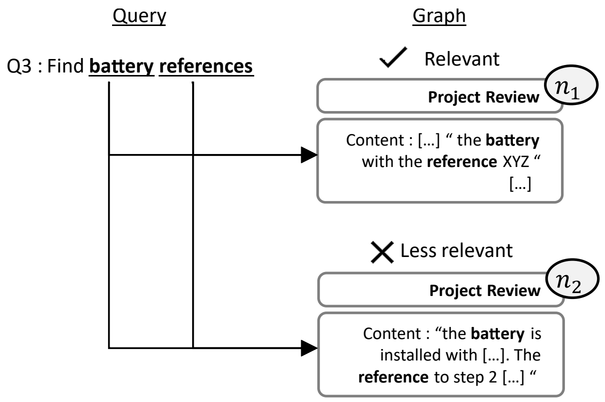

Query
Graph
Q3 : Find battery references
✓ Relevant
Project Review
n_1
Content : [...] " the battery with the reference XYZ " [...]
✗ Less relevant
Project Review
n_2
Content : "the battery is installed with [...]. The reference to step 2 [...] "

Fig. 10 Illustration of the root cause (1)

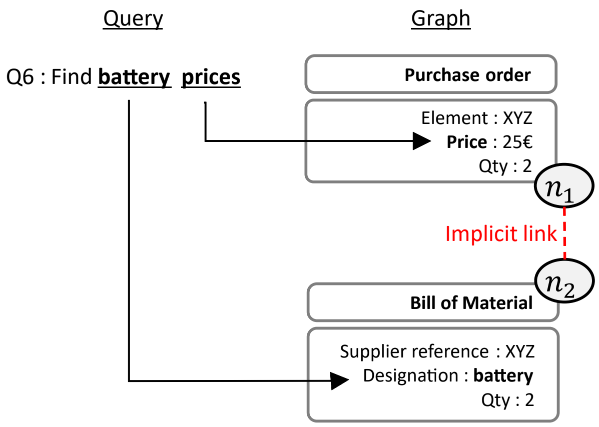


Fig. 11 Illustration of the root cause (5)

The detection of relationships between a priori unrelated data - 9.6%. Indeed, the implicit links between data are not exploitable. Cause (5) highlights some cases where, as illustrated in Fig. 11, related elements such as an element's functional reference and its supplier reference are disjoint in different enterprise systems. These missing but functionally existing relationships are necessary for the full-graph traversal and to provide all the expected results.

6 Discussion

6.1 Pratical discussion

The list of four key issues to consider listed in sect. 5.3 leads to an optimised query system based on a graph database and adapted to the manufacturing data [38].

This list was obtained according to the methodology described in sect. 3 with a heterogeneous, distributed and relational data set and by applying queries in response to the expected uses in the manufacturing industry described in sect. 4.1 and sect. 4.2 respectively.

Several key concepts can be put forward to address these issues.

Firstly, semantic search would allow the keywords of the search to be extended to all semantically related words. The use of a semantic network such as an ontology described in [39] or the semantic annotation described in [20] are possible approaches.

Secondly, the preprocessing of syntactic heterogeneity. In this paper, the lack of consideration of tables and bulleted lists in the transformation of textual content was highlighted. For example, it will be relevant to use specific algorithms for the detection of tables in images as presented in [40] to enrich the data graph. Furthermore, it is possible to extend the consideration to other types of syntax under the condition of validating them with the same process. This extension can include, for examples, geometric information, non-textual images and videos, annotations of 2D and 3D drawings or spatial data as processed in [11]. The pre-processing of syntactic heterogeneity can be included in the *data integration* step, thus generating new data transformation rules and query language enrichment.

And thirdly, record linkage between data can link distributed elements of the various data sources by translating them into relationships between nodes. Thus, these relationships can be exploited when applying queries on the graph. This record linkage can be achieved by supervised or unsupervised probabilistic methods, using blocking techniques to adapt to large volumes of data [41].

In a transversal way, the question of order by relevance and the question of response time must be considered. The answer to these two questions can be an addition of different optimisations, either at the *data integration* step, or at the *query integration* step, or at the query-data matching step.

6.2 Theoretical discussion

The results obtained by applying the iterative process presented in this paper are invariant under the same initial conditions. These conditions are constituted by the choice of the dataset and the list of queries. The choice of the dataset is dimensional because taking into account new syntactic and semantic data could generate new anomalies to be analysed and thus a different list of root causes. This is why it is important to choose a dataset that is representative of the context searched. It would also be appropriate to multiply the number of datasets used. The choice of queries is also important because the results are obtained by cumulating the anomalies on all queries. Thus, if some queries or several queries generate a large number of anomalies of the same type, this one may be overrepresented compared to the others. In order to reduce this discrepancy, it is important not to over-represent the same type of query and to multiply their number. It is also notable that the definition of the expected answers sets is also important. Indeed, the root cause analysis is performed from a list of missing or excess answers. This list of anomalies is obtained by comparing the expected responses defined at step A0 of the process with the results provided at the end of step A3. This list of expected answers thus determines the final result obtained. In order to strengthen the confidence in a unique result, it is then important to establish this list with several people if possible and at least to confront it with third parties in order to estimate the possible margin of error.

It should also be noted that the graph modelling of heterogeneous data allows the network thus created to be analysed using the various network analysis algorithms. For this, an extension of the properties of the nodes into sub-nodes should be considered.

7 Conclusion and future work

In this paper, the authors address the question: 'What are the minimum issues to be considered for a manufacturing data query system based on a graph database?'. To answer it, a methodology has been proposed and applied to a data set representative of the manufacturing industry context. In addition to taking into account the textual content of unstructured data as well as structured data, four main issues were considered: the semantic extension of the search keywords, the treatment of the syntactic heterogeneity contained in the unstructured data, the treatment of the results by order of relevance and the detection of relationships between a priori unrelated data. The resolution of these four challenges, some of which were outlined in sect. 6, can then open up to a proposed query system as illustrated in Fig. 12. The proposal includes a transformation of the structured and unstructured data into a graph (block 1.1), an enrichment of this graph by new relations (block 1.2), a transformation of the queries into a graph including a semantic extension of the keywords thanks to a lexical resource (block 2) and a matching between the transformed query q' and the graph giving the result to the query (block 3). This proposal could then help various business processes such as interactive design and manufacturing. In addition, the methodology described in sect. 3 can be adapted in order to evaluate this future proposal. It will then be ideal to expand the datasets and queries used. Finally, it can be envisaged transposing the conclusions of this paper not only to the field of manufacturing industry but to all enterprises dealing mainly with structured and unstructured textual data with a significant relational character. Moreover, the methodology employed is also applicable to other contexts where the nature and distribution of data may be different. It is then necessary to select datasets and queries adapted to the studied environment.

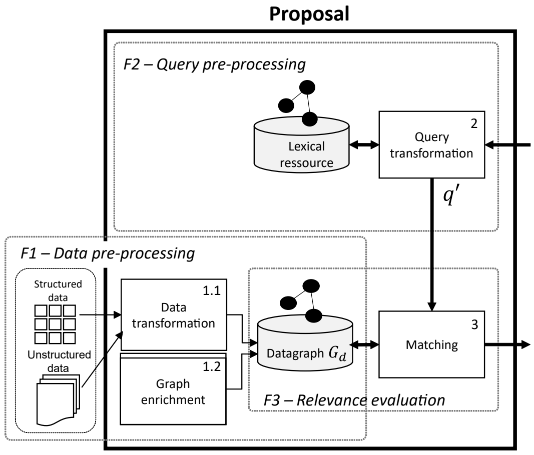


Fig. 12 Proposal for a query system

References

1. Segonds, F., Cohen, G., Véron, P., Peyceré, J.: PLM and early stages collaboration in interactive design, a case study in the glass industry. Int. J. Interact. Des. Manuf. (2014). https://doi.org/10.1007/s12008-014-0217-4
2. X.G. Ming a, J.Q. Yan a, X.H. Wang a, S.N. Li a, W.F. Lu b, Q.J. Peng c, Y.S. Ma :Collaborative process planning and manufacturing in product lifecycle management. Comput. Ind. (2008). https://doi.org/10.1016/j.compind.2007.06.012
3. Reinsel, D., Gantz, J., Rydning, J.: The Digitization of the World From Edge to Core. IDC White Paper. https://www.seagate.com/files/www-content/our-story/trends/files/idc-seagate-dataage-whitepaper.pdf (2018). Accessed 08 september 2020.
4. Lasi, H., Fettke, P., Feld, T., Hoffman, M.: Industry 4.0. Bus. Information Syst. Eng. (2014) https://doi.org/10.1007/s12599-014-0334-4
5. Kassner, L., Gröger, C., Mitschang, B., Westkämper, E.: Product Life Cycle Analytics: Next Generation Data Analytics on Structured and Unstructured Data. Procedia CIRP. (2014) https://doi.org/10.1016/j.procir.2015.06.008
6. Feldman, S., Duhl, J., Marobella, J.R., Crawford, A.: The hidden costs of information work. IDC White Paper. (2005).
7. Mirza, H. T.: Enterprise Information Retrieval: A Survey. In Proc. Tenth Int. Conf. on Enterp. Inf. Syst. - Vol. 2: ICEIS. (2008) https://doi.org/10.5220/0001674201410148
8. Stocker, A., Richter, A., Kaiser, C., Softic, S. : Exploring barriers of enterprise search implementation: a qualitative user study. Aslib J. Inf. Manag. (2015). https://doi.org/10.1108/AJIM-03-2015-0035
9. Nayak, A., Poriya, A., Poojary D.: Type of NOSQL Databases and its Comparison with Relational Databases. Int. J. Applied Inf. Syst. (2013) https://doi.org/10.5120/ijais12-450888
10. Miller, J.: Graph database applications and concepts with Neo4j. SAIS 2013 Poceedings. 24. (2013)
11. Schalbus S., Scholz J.: Spatially-linked manufacturing data to support data analysis. GI_Forum - J. Geogr. Inf. Sci. (2017) https://doi.org/10.1553/giscience2017_01_s126
12. Noel, S., Harley, E., Tam, K. H., Gyor, G.: Big-data architecture for cyber attack graphs representing security relationships in nosql graph databases. IEEE Symp. Technol. Homeland (2015).
13. Lysenko, A., Roznovăţ, I. A., Saqi, M., Mazein, A., Rawlings, C. J.: Representing and querying disease networks using graph databases. BioData Min. (2016). https://doi.org/10.1186/s13040-016-0102-8
14. Yoon, B.-H., Kim, S.-K., Kim, S.-Y.: Use of Graph Database for the Integration of Heterogeneous Biological Data. Genom. Inf. (2017). https://doi.org/10.5808 gi.2017.15.1.19
15. Bonnici, V., Russo, F., Bombieri, N., Pulvirenti, A., Giugno, R.: Comprehensive reconstruction and visualization of non-coding regulatory networks in human. Front. Bioeng. Biotechnol. (2014). https://doi.org/10.3389/fbioe.2014.00069
16. 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 Syst. Biology (2018). https://doi.org/10.1186/s12918-018-0616-4
17. Martínez-Bazan, N., Muntés-Mulero, V., Gomez-Villamor, S., Nin,J., Sánchez-Martínez, M., Larriba-Pey, J. L.:DEX: High-Performance Exploration on Large Graphs for Information Retrieval. Processing Sixteen ACM Conference Information Knowledge Management (2007). https://doi.org/10.1145/1321440.1321521
18. Mordinyi R., Schindler P., Biffle S.: Evaluation of NoSQL graph databases for querying and versioning of engineering data in multi-disciplinary engineering environments. 2015 IEEE 20th Conference on Emerging Technologies & Factory Automation (ETFA) (2015). https://doi.org/10.1109/ETFA.2015.7301486
19. Groger C., Schwarz H., Mitschang B.: The Deep Data Warehouse: Link-based Integration and Enrichment of Warehouse Data and Unstructured Content. In Proceedings of the 18th IEEE International Enterprise Distributed Object Computing Conference (2014). https://doi.org/10.1109/EDOC.2014.36
20. Henkel, R., Wolkenhauer, O., Waltermath, D.: Combining computational models, semantic annotations and simulation experiments in a graph database. Database (2015). https://doi.org/10.1093/database/bau130
21. Peukert, E., Watner, C.: Taking the LEAP: The Methods and Tools of the Linked Engineering and Manufacturing Platform (LEAP) (2016). https://doi.org/10.1016/C2015-0-02474-1

22. Ross, D. T.: Structured Analysis (SA): A Language for communicating ideas. IEE Transactions on software engineering (1997). https://doi.org/10.1109/TSE.1977.229900
23. Chowdhury G. G.: Natural language processing. Annual Review of Information Science and Technology (2003). https://doi.org/10.1002/aris.1440370103
24. Barsalou M. A. Root Cause Analysis: A Step-By-Step Guide to Using the Right Tool at the Right Time (2014). https://doi.org/10.1201/b17834
25. Arapakis I., Bai X., Cambazoglu B.: Impact of Response Latency on User Behavior in Web Search. SIGIR '14: Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval (2014). https://doi.org/10.1145/2600428.2609627
26. Schmitt, X., Kubler, S., Robert, J., Papadakis, M., LeTraon, Y.: A Replicable Comparison Study of NER Software: StanfordNLP, NLTK, OpenNLP, SpaCy. 2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS) (2019). https://doi.org/10.1109/SNAMS.2019.8931850.
27. Manning, C., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S., McClosky, D.: The stanford corenlp natural language processing toolkit. Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations (2014). https://doi.org/10.3115/v1/P14-5010
28. Pinquié, R., Véron, P., Segonds, F., Croué, N.: Requirement Mining for Model-Based Product Design. Int. J. Prod. Lifecycle Manag. (2016). https://doi.org/10.1504/IJPLM.2016.10001870
29. Rajbabua, K., Srinivasb, H., Sudhab, S.: Industrial information extraction through multi-phase classification using ontology for unstructured documents. Comput. Ind. (2018). https://doi.org/10.1016/j.compind.2018.04.007
30. Cordeiro, F., Galhardas, H., Leblay, J., Manolescu, I., Merabti, T.: Keyword Search in Heterogeneous Data Sources. Technical report (2020)
31. Alhabashneh, O,. Iqbal, R., Shah, N., Amin, S., James A. : Towards the development of an integrated framework for enhancing enterprise search using latent semantic indexing. Conceptual Structures for Discovering Knowledge. Lecture Notes in Computer Science (2011). https://doi.org/10.1007/978-3-642-22688-5_29
32. Totaro, G., Bernaschi, M., Carbone, G., Cianfriglia, M., Di Marco, A.: ISODAC: A high performance solution for indexing and searching heterogeneous data. J. Syst. Softw. (2016). https://doi.org/10.1016/j.jss.2015.11.043
33. Quix, C., Hai, R., Vatov, I.: GEMMS: A generic and extensible metadata management system for data lakes. CAiSE Forum (2016).
34. Vuong, T., Jacucci, G., Ruotsalo, T.: Proactive Information Retrieval via Screen Surveillance. SIGIR '17: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval (2017). https://doi.org/10.1145/3077136.3084151
35. Moreno-Schneider, J., Martinez, P., Martinez-Fernandez, JL.: Combining heterogeneous sources in an interactive multimedia content retrieval model. Expert Syst. App. (2017). https://doi.org/10.1016/j.eswa.2016.10.049
36. Jackson, R., Kartoglu, I., Stringer, C. et al.: CogStack - experiences of deploying integrated information retrieval and extraction services in a large National Health Service Foundation Trust hospital. BMC Med. Inf. Decis. Mak. (2018). https://doi.org/10.1186/s12911-018-0623-9
37. Lenadora, D., Wickramarachchi, A., Meedeniya, D., Mallawaarachchi, V., Perera, I.: An Adapter Architecture for Heterogeneous Data Processing in Bioinformatics Pipelines. 2019 Moratuwa Engineering Research Conference (MERCon) (2019). https://doi.org/10.1109/MERCon.2019.8818781.
38. Kim, L., Yahia E., Segonds F., Véron P., Mallet A.: iDATAQUEST a proposal for a manufacturing data query system based on graph. In proceedings of the 17th International Conference on Product Lifecycle Management, PLM 2020, Rapperswil, Switzerland, July 5–8, (2020).
39. Li, Y., Thomas, M., Osei-Bryson, K.: Ontology-based data mining model management for self-service knowledge discovery. Inf. Syst. Front. (2017). https://doi.org/10.1007/s10796-016-9637-y
40. Shafait, F., Smith, R. : Table detection in heterogeneous documents. DAS '10: Proceedings of the 9th IAPR International Workshop on Document Analysis Systems (2010). https://doi.org/10.1145/1815330.1815339
41. G. Papadakis, J. Svirsky, A. Gal, T. Palpanas: Comparative analysis of approximate blocking techniques for entity resolution. Proc. VLDB Endow. (2016). https://doi.org/10.14778/2947618.2947624