i-Dataquest: a heterogeneous information retrieval tool using data graph for the manufacturing industry

Lise KIM, Esma YAHIA, Frédéric SEGONDS, Philippe VÉRON, Antoine MALLET

Abstract

Manufacturing industry needs access to the data in order to realise its activities but also to generate new value-added knowledge. Nevertheless, it is confronted with a large and growing volume of heterogeneous data which limits its ability to exploit them optimally. Moreover, the data are distributed within different heterogeneous information systems, which limits the relationship exploration under the information retrieval process. Usually, the challenge is addressed by trying to manage and normalize the data structure in order to faster searching and exploiting them in a manufacturing context. For their part, the authors present i-Dataquest, an information retrieval system supported by (i) a graph-oriented model built from the structured and unstructured data of the company and (ii) a query system answering 'what' and 'about what' and (iii) generating three different results: a list of items, a list of property values and a list of sentences. The i-Dataquest prototype is built using Neo4J for the graph system generation, ConceptNet for lexical resource management and StandfordNLP for natural language processing. An evaluation of the prototype's performance is conducted through a data set representing a drone manufacturer. The results show that the transformation of specific content such as tables in the graph and the semantic expansion of queries significantly improves the recall and precision measures. The results also suggest improving filtering less relevant results by considering particularly queries looking for a specific value.

Keywords

Graph Database, Query System, Information Retrieval System, Manufacturing Data, Manufacturing Industry

1. Introduction

Access to the right information at the right time is an important issue for companies. It allows the efficient functioning of the various stakeholder activities, but also the acquisition of new knowledge that can generate value. It is a well-known issue for the manufacturing industry, enhanced by products and factories digitalisation, opening up the era of digital engineering and increasing the number of issues concerning information sharing. Since then, the sector has been looking to manage its data throughout the product lifecycle management to find the right information at the right time to be used by the right person. However, despite efforts to manage and standardise information, the manufacturing industry is facing a number of difficulties. First, the volume of data generated then

stored is large and growing. According to the International Data Corporation[1] in its white paper (Reinsel, et al., 2018), the manufacturing industry is estimated at 3,584 exabit, which is more than the volume of the finance and health industries added together. According to the same reference, this volume is expected to grow by 30% by 2025. Second, the data are syntactically heterogeneous with various formats for various business needs. The data are also structured and managed by different information systems such as PLM, PDM[2], ERP[3] or MES[4]. These systems allow, among other things, the creation of explicit relationships between the various elements of information. Data can also be unstructured, for example in geometry, text, image or spreadsheet files stored on the company's various servers. Third, the data are semantically heterogeneous as created by various actors working for different businesses and using different software publisher using their own vocabularies and languages. Finally, all this data are distributed through various databases and under different servers, which makes them disjointed despite the possible relationships between them. These relationships, characterised as 'implicit', are then not exploitable to derive knowledge from them. All of these difficulties slow down the search for information, thus impacting the time dedicated to added value activities.

In order to overcome these difficulties, many solutions are proposed in companies to query and exploit the data (Emmott, et al., 2019). Nevertheless, some of this current solution and works make little use of the relationships between the data. It concerns the solutions supported by column-oriented and document-oriented databases, which are more flexible and quicker to query than those made in SQL[5] databases (Nayak, et al., 2013), but they are less suitable for exploiting the relationships between data. Indeed, manufacturing industry data are mostly relational, implicitly linked as argued above or explicitly linked. A second category of the solutions exploits the relationships between data because they are supported by graph-oriented databases, which are particularly used to model networks and explore relationships between entities (Angles & Gutierrez, 2008). The comparison of graph databases to usual relational databases conducted by the authors of (Batra & Tyagi, 2012) supports this information stating a more flexible modelling and faster interrogation of relational data in the case of graph databases. However, these solutions concern only a limited number of data sources such as in (Schabus & Scholz, 2017) and in (Yoon, et al., 2017) or consider a limited number of challenges such as the relationship enrichment in (Gröger, et al., 2014) or semantic annotation in (Henkel, et al., 2015) for examples.

[1] International Data Corporation (IDC) is a global information technology market research and consulting group. https://www.idc.com/

[2] PDM for Product Data Management

[3] ERP for Enterprise Resource Planning

[4] MES for Manufacturing Execution System

[5] SQL for Structured Query Language

The work of (Kim, et al., 2020) depicted the main milestones in order to define an information retrieval system for manufacturing industry based on pre-processed data under a graph-oriented data model. Indeed, this system must consider several main challenges including the expansion of keywords to semantically close terms, the treatment of syntactic heterogeneity of source data and the display of particularly relevant results.

The expansion of query terms to semantically close terms means that the query is not limited to the user's vocabulary. The semantic heterogeneity is an issue when searching for information.

The treatment of syntactic heterogeneity implies here the retrieval of data elements carrying meaning for the information retrieval. Both structured and unstructured data must be processed; for example, it is the case when retrieving a precise value from a table containing columns and rows or when detecting the arguments sought and contained in a sentence.

A good display of the "particularly relevant" elements compared to the "less relevant" ones should allow the user to quickly access the information sought. Indeed, too many results increase the time of their explorations and reduce the probability of accessing to the desired result.

To improve the capacities of the manufacturing industry to exploit its data while taking into account the previous main challenges, a framework for a graph-based information retrieval system has been proposed (Kim, et al., 2020). The aim of this paper is to present the i-dataquest proposal that fits the framework that has been enriched, implemented and evaluated. More specifically, the contributions of this paper are:

(1) The description of the information retrieval system adapted to the manufacturing industry including the rules of :

- Transformation of structured and unstructured data into a graph-oriented data model
- Expression of user queries according to three distinct typologies of graph research
- Expansion of query terms with a weighted multilingual knowledge graph

(2) The qualitative evaluation of the proposal applied to a set of data and queries representative of the context according to :

- The resolution of each key challenge
- The typology of the considered queries

The research question is formulated as "how to return exhaustive and relevant information in a distributed, heterogeneous and relational context" and the paper is constructed as follows: section 2 details the work in relation to the different choices carried out to construct the proposition, in particular the use of a graph for the information retrieval and the representation of semantic relations between terms, section 3 presents the proposal and its sub-functions, section 4 describe the

experimental process and the evaluation performed on the proposal, section 5 discusses the results obtained and we conclude in section 6.

2. Related work

The use of graphs for the representation of distributed, heterogeneous and relational data as well as for the representation of semantic networks has widely been studied. The following section summarizes the main reference works used and highlights the lack of a complete answer to the authors' research question. The section 2.1 describes the main applications and advantages of using graph-oriented databases in the representation of information networks. The section 2.2 describes more precisely the works involving data graphs to represent heterogeneous and relational data specifically for information retrieval. Finally, the section 2.3 describes current methods using and generating graphs as a representation of a lexical network for the semantic expansion in particular.

2.1 Graph representation orientation

Data models of the NoSQL-type whose terminology emerged in the early 2000s has been popularised by the GAFA[6] (Leavitt, 2010). These models have responded to the problem of storing and managing large amounts of data, particularly on the web. NoSQL databases allow overriding the gaps of usual relational databases that have become too restrictive and slow to search for elements in the light of the multitude and variety of data to be managed (Nayak, et al., 2013) in a big data context (Moniruzzaman & Hossain, 2013). More specifically, column-oriented databases became the models particularly used thanks to its powerful and flexible analysis of a wide range of data (Abadi, 2008). Whereas, graph-oriented databases are typically used to model networks and explore relationships between entities (Angles & Gutierrez, 2008). Among its major uses, there is the analysis of social networks, which allows, for example, the detection (Li, et al., 2020) and tracking (Dakiche, et al., 2019) of communities, but also the personalized recommendation (Zhou & Han, 2019) and also the detection of malicious behaviour (Alassad, et al., 2021). However, the use of graph-oriented data models is not limited to the perimeter of social networks. In fact, it allows representing the elements of any environment composed of initially heterogeneous data, which correspond to our manufacturing context. The graph also allows representing knowledge composed of concepts with ontologies (Dou, et al., 2015) and the semantic web (Sabou, et al., 2017). When this knowledge base is enriched to derive new knowledge, it is then qualified as a knowledge graph (Ehrlinger & Wöß, 2016).

[6] GAFA for Google, Apple, Facebook, Amazon describes as the giants of the web

2.2 Graph-oriented data models for information retrieval systems

In the proposal, the authors have chosen a graph-oriented data model for the representation of the manufacturing data. The data and the relationships between them are represented by the nodes and edges of the graph allowing exploiting them in the search for information. This choice to unify heterogeneous elements in a graph-oriented data model is the one made in many domains such as cyber security (Dawood, 2014) where data representing infrastructure elements or attack events, are integrated (Noel, et al., 2016), for the representation of government data (Lin, 2020), for the data journalism (Berven, et al., 2020), for the representation of users and their points of interest (Qiao, et al., 2020) or for the biology field where protein or genome databases are integrated for example (Yoon, et al., 2017). All these works consider as input heterogeneous data whose unification under a graph-oriented data model becomes revealing of new information. These approaches use graph databases such as Neo4J[7], DEX[8] or Titan[9] to manage the data (Patil, et al., 2018). To take into account specificities linked to a context or particular stakes, other approaches customize their approaches.

In the context of the manufacturing industry, the graph model is used in many different applications. Indeed, the graph theory is used in different domains to interrogate the relationships between several artefacts distributed in the enterprise throughout its life cycle (hedberg Jr, et al., 2020); to model the continuous traceability of data interconnecting objects (Kuhn & Franke, 2021), to combine event information captured with their service activity processes (Kammler, et al., 2019); to establish the cost of reconfiguring provisioning chains (Guo, et al., 2018); to target production resources from product descriptions (Beisheim, et al., 2018); to represent the relationships between parts, resources and activities in the industrial complex network (Leng & Jiang, 2019); to version engineering data from different disciplines (Mordinyi, et al., 2015) or for the representation of a semiconductor manufacturing environment including the particular typology that is the spatial representation of the different elements (Schabus & Scholz, 2017).

Because we aim to meet the challenges listed by (Kim, et al., 2020), we are looking for an information retrieval system solution based on a graph constructed from the enterprise's heterogeneous data exploiting both the textual and tabular content of text and image documents as well as database records while semantically extending the terms of the query. While commercial solutions for information retrieval in companies are numerous but they do not offer a solution that meets all the criteria sought for the manufacturing context. Besides, the graph database management systems such as DEX or Neo4J offer functionalities such as native data transformation (Martínez-Bazan, et al., 2007) and facilitate querying but they need to be customized to fully meet the identified millstones.

[7] https://neo4j.com/
[8] DEX became Sparksee since 2014 – https://sparisity-technologies.com/#sparksee
[9] https://titan.thinkaurelius.com/

2.3 Knowledge graphs

To meet the challenge of semantic search, the choice of a knowledge graph. Several methods are used to create this graph. Among the trends highlighted by the review of all these techniques in (Azad & Deepak, 2019), there is the use of hybrid sources of data and the use of relevance feedback that have been included in our proposal. Taking up the classification of (Raza, et al., 2019), we distinguish two main approaches for constructing the compound graph of semantic relations between terms. The first one is the global analysis; it includes the use of external resources such as ontologies for e.g. sentiment analysis (Meškelė & Frasincar, 2020), thesauri or similar knowledge graphs. Ontologies are a recurrent support in the knowledge representation and sharing in engineering (Yang, et al., 2019) as recently for complex product development (Wu, et al., 2018). These resources can be monolingual such as WordNet[10], a lexical database of English, or multilingual such as ConceptNet[11], a knowledge graph and some proposal can combine these two resources for semantic enrichment (Huet, et al., 2021). It also includes the analysis of terms in the collection of documents. Besides, it includes the analysis of the user's browsing logs. Finally, it includes the use of existing relationships between pages in the web context. The second approach is the local analysis where one can retrieve indications from the user based on the relevance of the obtained results list or deduced from the results obtained in relation to the query. To answer the expansion query issue, ConceptNet has shown great potential for improving search results for difficult queries (Kotov & Zhai, 2012).

Concerning the general application of knowledge graphs in the manufacturing industry, there are many recent and diversified works on the subject such as for the management of CAD design rules (Huet, et al., 2021), additive manufacturing (Ko, et al., 2021), resource allocation (Zhou, et al., 2021) and the planning of assembly (Zhou, et al., 2021), or more generally the profiling of employees (Munir, et al., 2020) or the proposal of knowledge graphs around industry 4.0 (Bader, et al., 2020). Moreover, (Buchgeher, et al., 2021) reveals that the technical literature on the subject is growing fast. The review highlights a plurality of applications of knowledge graphs in this context, classifying the different papers into five manufacturing domains such as machinery, chemical or transport equipment manufacturing; six use cases of which the most represented is that of knowledge fusion and eight system kinds of which search-based application is strongly represented. Following this classification, our use of knowledge graphs is general to the manufacturing domain, its aims the generation of a system kind of search-based application.

[10] https://wordnet.princeton.edu/
[11] http://conceptnet5.media.mit.edu/

2.4 Synthesis

Related work in section 2 shows the relevance of using graph-oriented databases to represent and search for relational information in different fields. In particular, in the context of information retrieval systems using a graph of data, the literature lacks a solution considering both structured and unstructured data while providing an answer to the issues listed by (Kim, et al., 2020) including the treatment of syntactic specificities, the semantic research and finally the display of particularly relevant results. Concerning the specific issue of semantic extension, the state of the art has highlighted the possibilities offered by lexical networks, which can take the form of a multilingual knowledge graph such as ConceptNet. The choice of using a knowledge graph rather than a multilingual dictionary is made because of the variety of relationships allowed, such as that to the usage of the initial word in addition to its synonyms, contexts and types. Furthermore, with a knowledge graph it is also possible to refine the search by focusing on a user's usual context, for example. For all these reasons, the authors propose in the following section an information retrieval system using a graph-oriented database to represent the manufacturing industry data and a multilingual knowledge graph for semantic extension.

3. i-Dataquest system

In this section, the authors describe the proposal that answers the identified issues and fits into an enterprise environment with multiple data sources from different information systems publishers and multiple collaborative partners. The connection to the various databases including access identification tasks for example must be done beforehand and is not covered here. The i-DATAQUEST proposal, shown in Fig. 1, is composed of the following three modules: data pre-processing, query pre-processing and relevance evaluation and feedback.

(A) The data pre-processing module is composed of the block “Generation of the data graph”. This block, detailed in section 3.1, allows transforming syntactically heterogeneous data coming from multiple data models into a single graph-oriented data model named G_d. The data structured in tables distributed in the enterprise are retrieved and transformed. Each table row becomes a node whose properties are the names and values of the columns. In the case of relational databases, the relations between elements expressed by foreign keys between tables are transformed into links between nodes. Each document containing unstructured information is then transformed into a node whose properties are the document metadata. The extracted textual content is stored in one of the node properties except in the case of tables where the structured data transformation rules are applied.

(B) The query pre-processing module is composed of sub-block (B1) “query transformation” detailed in section 3.2 and the sub-block (B2) “knowledge graph creation” detailed in section 3.3. The sub-block B1 "query transformation" has as input a query Q expressed by the user from a graphical interface according to two variables valued by words: the what (Qw) and the about what (Qa). This valorization helps to express four types of requests: the search for an element mentioning information like the search for all the elements related to a product, the search for a certain type of element like the search for an activity process, the search for a precise information like the search for a product property and the search for a sentence expressing specific notions like the search for a product requirement. Once the query is integrated, block 1 queries a multilingual knowledge graph to extend the query terms (Qw and Qa) to all semantically related terms. This extension is performed to simple synonyms but also to terms with more complex relationships such their context (the battery in the electronic context), their use (the battery is used for charging or for autonomy) or their type (the battery is voltaic). The multilingual knowledge graph is created with sub-block B2 from external resources. This graph contains the words in nodes and the semantic relations between each word by the relations between the nodes. The nodes are then weighted according to the use of the words in the query. The user can manually select the words to be extended or not, although a pre-selection is suggested by default according to this weighting. Block B1, once all the terms have been retrieved, transforms the query into a graph language, e.g. "search for nodes or sentences in properties like X that meet the conditions Y ”.

Finally, (C) *the relevance evaluation and feedback module* match the query and the data to provide the answer to the initial query. Thanks to the query transformation into graph language performed by block B1, three types of answers are displayed to the user according to his query: the list of nodes representing a list of documents and records in a data table, the list of node property values and the list of sentences contained in node properties.

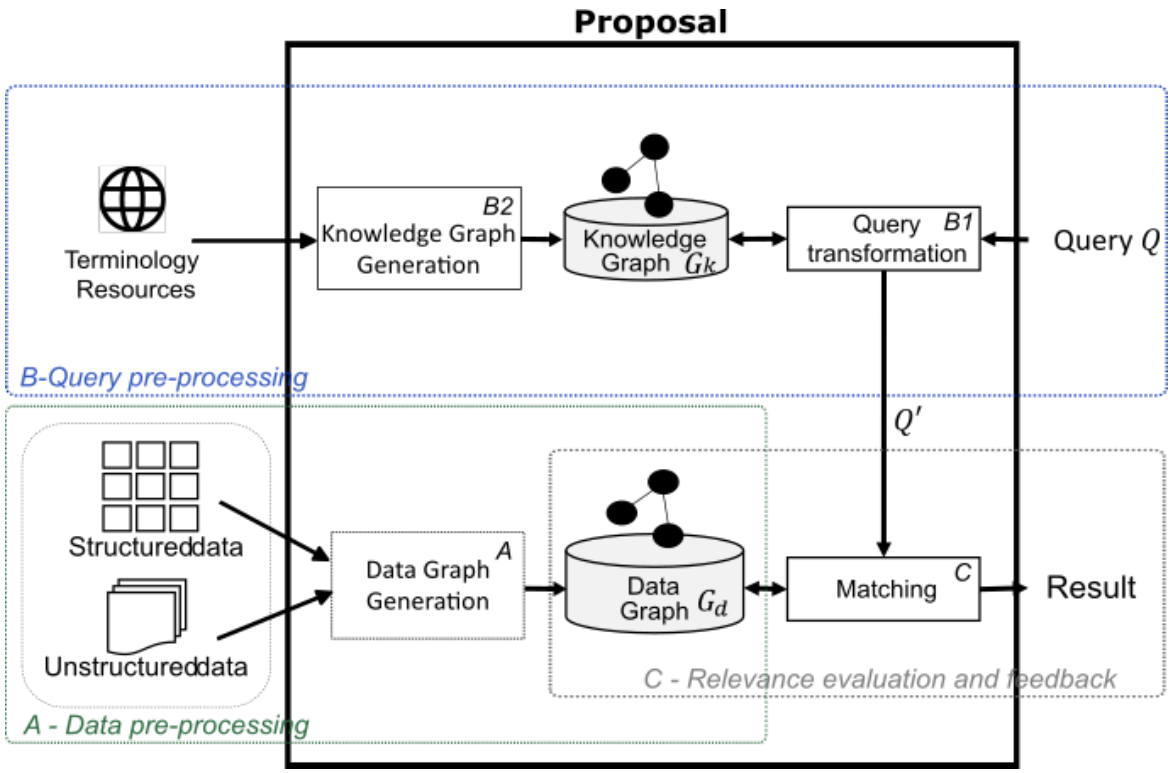


Fig. 1- i-DATAQUEST. A retrieval system proposal and its sub-blocks

The Fig. 2 illustrates an example dealing with the battery of a drone. The two graphs used in the proposal: (1) the data graph representing the set of heterogeneous and distributed data of the company and (2) the knowledge graph used for the semantic extension of the query terms. Thus, for a query "what is the price of the battery", the proposal browses the data graph looking for battery but also for the accumulator (terms found with the knowledge graph) and for a property named as price. For a query "what are the requirements of the battery", the proposal browses the data graph looking for sentences expressing requirements and mentioning the battery or the autonomy (terms found with the knowledge graph).

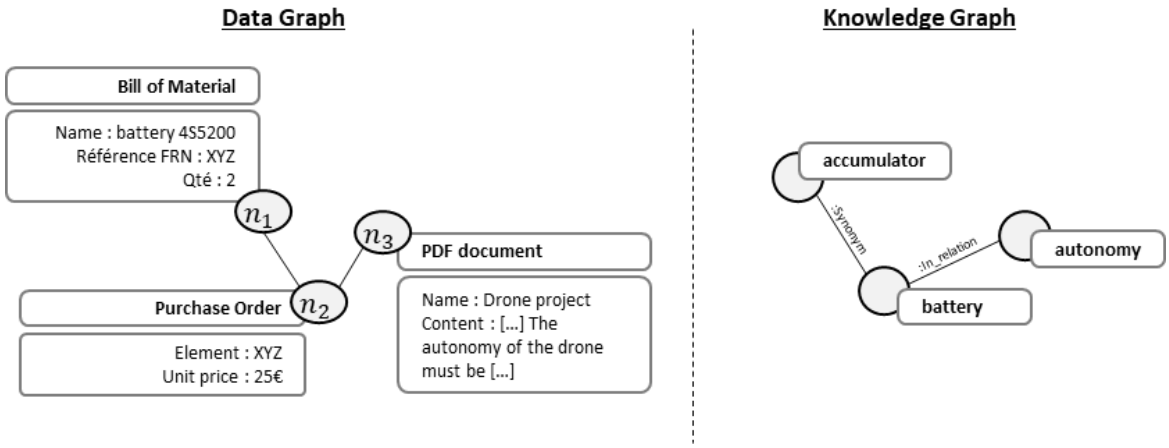


Fig. 2 - Illustration of the data graph and the knowledge graph, example of the battery of a drone

The main notations used in this paper are listed in Table 1.

Table 1 - List of main notations

Notations	Descriptions
$\boldsymbol{n}$	node of the graph
$\boldsymbol{p_n}$	properties of the node $\boldsymbol{n}$
$\boldsymbol{v_{p_n}}$	value of the property p of the node $\boldsymbol{n}$
$\boldsymbol{label_n}$	label of the node n
$\boldsymbol{r}$	relationship of the graph
$\boldsymbol{label_r}$	label of the relationship r

3.1 Data graph generation

This step consists in building the data graph G_d available to the query. This graph contains all the information of the company and allows to retrieve the answers to the queries. For this, the syntactic heterogeneity considered includes relational databases, the textual content of the text and image files and tables of the text, image and spreadsheet files. All of these elements are then transformed into a graph-oriented data model. The graph-oriented data model is described by :

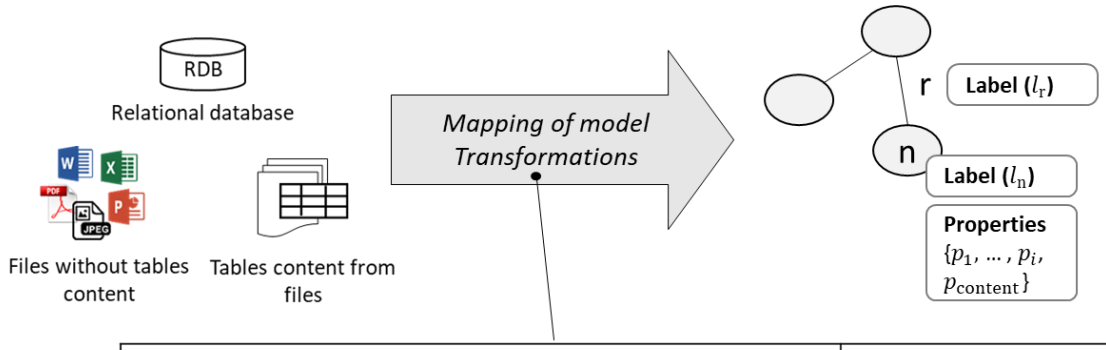
$$G_d = \{n, r\} \leftrightarrow \{: label_n\{p_n: \ v_{p_n}\}, (n) - [: label_r] \rightarrow (m)\}$$

The first transformation is from relational database records to nodes and from relationships (expressed as foreign keys) to relationships between nodes. These transformations are common to the Extraction, Transformation and Loading (ETL) tools (Patel & Patel, 2020) needed in the context. The second transformation is from all the files browsed on a server to nodes, whose properties are filled with file metadata and whose text content is embedded in a node property named “content”.

This transformation is completed by considering the content type ‘table’ contained in unstructured data. Each table row is considered as a sub-node of the document node whose properties are the

different columns of the table. Thus and as illustrated in Fig. 3, we distinguish five types of transformation:

- the creation of nodes from relational database records,
- the creation of links from relations between tables in relational databases,
- the creation of nodes from unstructured documents of the textual, spreadsheet or image type,
- creating sub-nodes from table rows found in documents,
- creating links between the previous sub-nodes and the nodes representing the original document.



	node					relationship	
datasource	n	l_n	Name (p_n)	Value (p_n)	Value($p_{content}$)	r	l_r
relational database	tuple	table name	attribute name	attribute value	-	foreign key	:in_relation_with
file	file	type	metadata name	metadata value	file text content without table content	-	-
table content in file	row	spreasheet name	column name	column value	-	with n_{file}	:contains

Fig. 3 - Set of rules for transforming structured and unstructured data to the graph-oriented data model

3.2 Queries preprocessing

To meet the expected uses of a manufacturing industry query system, the proposal includes four types of queries generating three types of answers and expressed by the user through two main variables. The two variables described below make it possible to tag key terms in the expression of the information need influencing research, similar to the part-of-speech tagging method in Natural Language Processing (NLP) (Chowdhary, 2020).

The first query type noted (i) concerns the search for any data item mentioning specific information such as "search for all items related to the battery (Q1)". The second query type noted (ii) concerns the search for data of a certain type mentioning specific information such as "search for filter cleaning standard (Q2)". The third query type noted (iii) concerns the search for a specific item of data mentioning specific information such as "search for battery price (Q3)". Finally, the fourth query type noted (iv) concerns the search for a specific phrase such as "search for battery life requirements (Q4)".

These four generic queries generate three possible types of answers: (I) the node list, (II) the list of property values and (III) the sentence list.

Queries (i) and (ii) generate only answer type (I). For example, the Q1 expects in answer the list of nodes mentioning 'battery'. The Q2 expects in answer the list of nodes of the standard type mentioning 'filter cleaning'. Next, queries (iii) generate answer types (II) and (III). For example, the Q3 expects in answer the values of the properties named 'price' of nodes mentioning 'battery', the values of the properties named 'price' of neighboring nodes mentioning 'battery' and finally the sentences mentioning 'price' and 'battery' like "the price of the battery is...". Finally, queries (iv) generate answers of type sentence. For example, the Q4 expects as an answer the sentence expressing the requirement and mentioning 'autonomy'.

In order to formalise these types of queries, we use the following two variables:

(Q_w) The What variable: In order to retrieve the user's intention regarding the focus sought, the proposal uses a first variable of free text type named 'What' and noted Q_w. If this variable is empty, the user then makes a search of the type (i) and if it is filled in, the user makes a search of the type (ii), (iii) or (iv). Indeed, the user will express under the variable Q_w his will to find a standard, a price or a requirement.

(Q_a) The About What variable: The second variable used is of free text type and named About What, noted Q_a. It expresses the characteristics of the expected answer.

One variant is *($Q_{a=requirement\ or\ choice}$) Search for phrase expressing*; it integrates a specific query transformation behaviour for the values of Q_a = "requirement" and Q_a = "choice". In these cases, the system uses the NLP methods to detect all sentences containing action verbs or modals expressing Q_a following the example of the proposal (Pinquié, et al., 2016) dedicated to the search for requirements in various documents. For example, the search of requirements will look for all sentences containing verbs or modals such as "requires", "demand", "must" …

Additional conditions for the results display: In order to limit the less relevant results, the proposal removes the results obtained when Q_a and Q_w are in a same text but more than 8 words apart.

(S) The parameter sentence: In order to optimise the answer time, the proposal allows excluding the search for sentences thanks to the option sentence (Y/N) noted S.

Finally, taking the example of a drone manufacturing company, the example of transformations of query types transposed into graph search is detailed below:

Q1

Need: Find all objects related to 'batterie' (battery translated into French)

Valuation of variables: Q_a ='battery' ; S =No

Type of query: (i)

Element search in graph : all nodes mentioning 'battery'

Q2

Need: Find the process of recruitment

Valuation of variables: Q_w ='process' ; Q_a='recruitment' ; S =No

Type of query: (ii)

Element search in graph: all nodes mentioning 'recruitment' in a property and another 'process' in another property

Q3

Need: Find the battery price

Valuation of variables: Q_w ='price' ; Q_a='battery' ; S =Yes

Type of query: (iii)

Element search in graph : value of the property named 'price' associated to a node mentioning 'battery' AND value of the properties named 'price' in the nodes close to the nodes mentioning 'battery' AND sentence in a property mentioning 'price' and 'battery'

Q 4

Need: Find the battery requirement

Valuation of variables: Q_w ='requirement' ; Q_a='battery' ; S =Yes

Type of query: (iv)

Element search in graph: Sentence using verbs or modals expressing requirements and mentioning 'battery'

A transformation is then performed from these three-parameter queries Q_a, Q_w et S to a graph query. In a simplified way, expression (1) transcribes the search for a list of nodes, expression (2) the search for a property's value and (3) the search for a sentence.

(1) $MATCH(n)\ WHERE\ n.p_{1\to m} =* Q_a *\ \ RETURN\ n$
(2) $MATCH(n)\ WHERE\ n.Q_w\ IS\ NOT\ NULL\ and\ n.p_{1\to m} =* Q_a *$
$RETURN\ n.Q_w\ UNION\ ALL\ \ MATCH(n) - [* 1] -$
$(m)\ WHERE\ m.Q_w\ IS\ NOT\ NULL\ and\ m.p_{1\to m} =* Q_a * RETURN\ m.Q_w$
(3) $MATCH(n)\ WHERE\ n.p_{1\to m} =* Q_w *\ \ and\ n.p_{1\to m} =* Q_a *\ \ RETURN\ n.sentence$

3.3 Extended query

The multilingual knowledge graph: The semantic expansion of keywords is translated by the expansion of the variables Q_a and Q_w to the set of semantically close terms. To fulfill this function, a multilingual knowledge graph noted G_k is generated from external resources. In order to introduce only useful terms in G_k, a pre-filtering is performed on the terms contained in the enterprise data graph G_d. This list of terms is then refined thanks to a Tf-idf[12] vectorization (Qaise & Ramsha, 2018). The nodes of multilingual knowledge graph contains three properties: the property $label$ for the name of the term, the property $language$ for the language of the term included in the $lang$ list, the property $weight$ for the weighting of the term according to its actual use in the proposal. The multilingual knowledge graph also contains the relationship property $type$ describes the relationship type between terms. This $type$ can take the values [Synonym, Translation, In_relation, Abbreviated_by]. The multilingual knowledge graph can then be described as below.

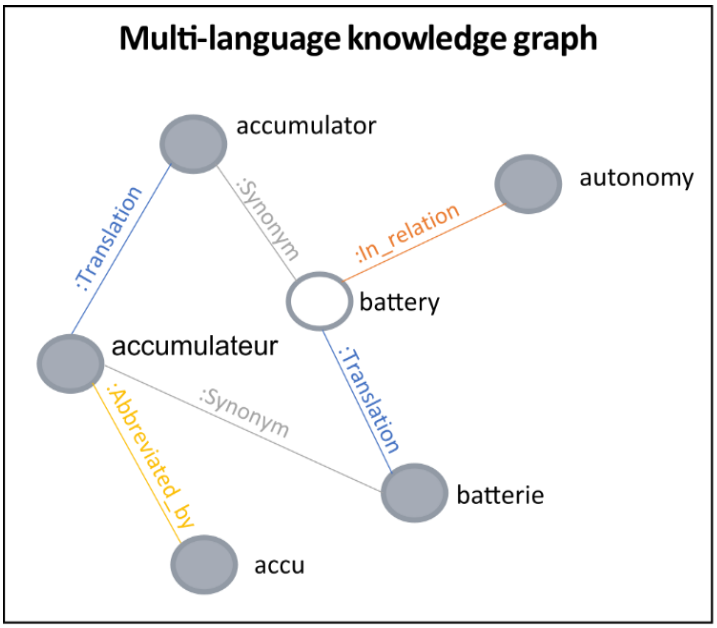
$G_k = \{n, r\} \leftrightarrow$
$\{:\text{KnowledgeGraph}\{\text{'label'}:\ word,\text{' language'}:\ lang,\text{' weight'}: w\ \},(n) - [:type] \to (m)\}$

Weighting of terms according to their relevance: The weighting of terms in the multilingual knowledge graph allows taking into account the real use of terms. Thus, the proposal adapts to the vocabulary and language of the users from the next queries. The following rule is applied: at initialization, only synonyms of another language of Q_a and Q_w are used. The other related terms are proposed to the user. If terms are used and selected, then weight = weight + 1. If terms are deselected, then weight = weight - 1.

[12] Tf-idf for Term Frequency-Inverse Document Frequency

Expansion of queries by querying G_k*:* When the user submits a query, Q_a and Q_w are then associated to their proximity terms 1 of type synonymous and with a different language and to the other proximity terms 1 if their weight is greater than or equal to 1.

Finally, the weighted multilingual knowledge graph is shown in Fig. 3 following the example of the 'battery' term. The figure is composed of the graph representation of the nodes and edges representing the terms and the relationships between terms respectively. The figure also represents in the table form the properties of the different nodes of which language and weight are part. Finally, the figure lists the step of graph creation but also its enrichment over time.



Properties :

name	*language*	*weight*
battery	en	1
batterie	fr	1
accumulator	en	1
accumulateur	fr	0
accu	en	0
autonomy	en	1

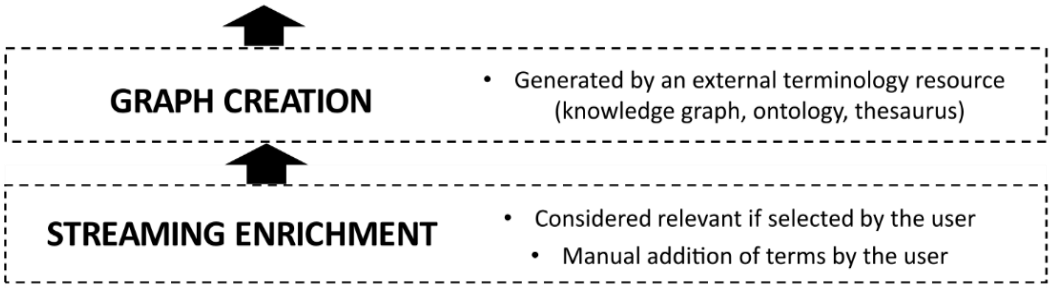


Fig. 4- Representation of the weighted multilingual knowledge graph

4. Performance evaluation

It is important to assess the relevance of the proposal presented in section **Erreur ! Source du renvoi introuvable.** in order to confront it with the research question "how to return exhaustive and relevant information in a distributed, heterogeneous and relational context". It is also important to analyse the results obtained in order to guide future research. The measures, dataset and queries used for the

proposal evaluation are presented in the section 4.1. The results and their analysis according to 'implemented functions' and 'query typologies' are presented in section 4.2.

4.1 Experiment set up

This section is split into four parts. First, the performance criteria and the measures used to evaluate the system are presented. Second, the dataset used is described according to the criteria of functional scope, syntactic and semantic diversity and volume represented. Third, a description of the queries used is given. Finally, the last part lists the tools used.

4.1.1 Performance criteria

To assess the quality of the proposal, the results obtained are judged according to three criteria. The first is its capacity to provide all the expected results represented by the recall score $r = \frac{|N_p \cap N_r|}{|N_r|}$ and the second is its capacity to provide only good results represented by the precision score $p = \frac{|N_p \cap N_r|}{|N_p|}$ where N_r is the set of results obtained for each query and N_p is the set of expected results for each query defined by the systematic scanning of all the objects contained in the dataset. Finally, the third criterion is the harmonic mean between recall and precision called F-Measure $F_\beta = (1 + \beta^2) * \frac{p*r}{\beta^2 p + r}$ where β is a weighting of one of the measures relative to the other. In our case, we choose β= 1 judging that it is just as important to obtain all relevant results as to display only relevant results. Finally, each missing or excessive result is analysed according to a root cause analysis and then classified using an Ishikawa[13] (Barsalou, 2014) method that allowed the identification of large families of anomalies (Kim, et al., 2020).

4.1.2 Dataset

Similar to the data of a manufacturing company, the PAINT'R dataset[14] was selected for its multi-activity functional coverage, its strong syntactic heterogeneity including structured and unstructured

[13] Graphical method for searching and representing the different root causes of a problem.

[14] The data set is available at : https://www.kaggle.com/dataset/a4ba6c3dbe1bc5a1cc8f05bb7ad825bcce106bff68ab582877a82107c000f9b1

data as well as its semantic heterogeneity due to multiple creators and a double language used. Each of the criteria is described below.

Functional scope of the dataset: In order to target a dataset representative of the manufacturing industry, the selected case study is a drone manufacturing company composed of several business units. The design activities include requirements management, definition of the product and its components, numerical simulation and the engineering bill of materials. Production, logistics and purchasing activities include documents relating to assembly lines, suppliers list and purchase orders. Maintenance and after-sales activities include maintenance manuals and customer returns. Methods and quality activities include process, standards and methodology documents. Project management activity include planning, roadmaps and reviews reports. Finally, the human resources activity include employee CVs.

Syntactic diversity of data: the elements of the dataset are provided from different data sources. Their syntax is heterogeneous. It includes structured data coming from the digital model of the drone managed under Dassault System's 3DExperience solution[15] (including the storage of 3D elements) and other relational databases managed in MySQL[16]. It also includes unstructured data such as text files (.doc .xml .txt .log .ppt .pdf), images (.jpg .png), videos (.mp4) and spreadsheets (.xls).

Semantic diversity of data: A multitude of actors who have the same backgrounds participated in the creation of the dataset. As a result, the vocabulary used is varied even though all the actors are French-speaking. The languages included in the dataset are therefore limited to French and English.

Quantity: The dataset is composed of 472 elements distributed as follows: 47% are unstructured data such as text files, images, videos and spreadsheets, 21% are three elements whose only function is the relationship between one object and another, 17% are relational database elements and 15% are CAD models. These 472 elements generated a data graph composed of 3260 nodes, 5411 relationships and 270 properties.

4.1.3 Queries

Functional scope of queries: The queries used to have been selected to meet a list of expected uses. The uses of this list have been characterised as necessary and innovative for the manufacturing

[15] https://www.3ds.com/3dexperience/
[16] https://www.mysql.com/

industry by the digital consulting firm Capgemini[17]. This use list is for different user profiles who can act and search for data throughout the product's life cycle. These queries include the following four types of queries: (i) the search for all elements (Q_w= ' '), (ii) the search for elements answering a certain type of element ($Q_w \neq$ ' ' and $Q_w \notin P$ where P is the set of properties of the graph), (iii) the specific value search ($Q_w \neq$ ' ' and $Q_w \in P$) and (iv) the specific phrase search (Q_w <> ' ' and $Q_w \in S$), mentioning one or more terms ($Q_a \neq$ ' ').

Semantic diversity scope of the queries: the *queries* used were chosen according to two languages, French and English, in accordance with the languages used in the dataset.

Samples: The four queries Q1, Q2, Q3 and Q4 cited in section 3.2 are part of the 25 used for all 25 queries.

4.1.4 Implementation

The system under consideration uses several open source tools described in Table 2. Neo4J is chosen as a graph data storage system because it is judged with ArangoDB[18] as one offering the best features with a great ease of use and a good query power (Diogo & Jorge, 2018). Moreover, Neo4J has a large community with a rich help documentation. It is notably used in several works indicated in the section 2.2 such as (Schabus & Scholz, 2017) and (Noel, et al., 2016). Apache Tika[19] is chosen because it is a usual tool for extracting metadata and text from documents. In the case of text in images, Tesseract[20] also under the Apache foundation is chosen. Concerning the use of natural language processing algorithms, the use of the toolkit provided by Standford NLP[21] was selected in particular the use of the existing Neo4J plugin[22]. Python[23] is the chosen programming language because it is intuitive, massively used and well documented with many open source libraries at disposal of which Py2neo[24] is part and allows communicating with Neo4J.

Function	**Tools**
Graph storage	Neo4J
Tect extractor	Apache Tika
Text extractor from images	Tesseract
Natural Language Processing	Stanford CoreNLP

[17] https://www.capgemini.com/
[18] ArangoDB, the multi-model database for graph and beyond
[19] https://tika.apache.org/
[20] https://opensource.google/projects/tesseract
[21] https://standfordnlp.github.io/CoreNLP/
[22] https://github.com/graphaware/neo4j-nlp-stanfordnlp
[23] https://www.python.org/
[24] https://py2neo.org/2020.0/

Python-neo4J library	Py2neo
Development language	Python

Table 2 - List of tools used and their functions

4.2 Experiment evaluation

The evaluation was conducted in parallel with the proposal construction. Thus, it is easier to evaluate the real contribution of each sub-function. The order of the steps was defined to deal with as many remaining anomalies as possible.

Step 0 "initial system": The data graph contains the metadata and textual content without table treatment and the keywords of the query are not expanded.

Step 1 "with table data integration": integration of the table transformation in the graph.

Step 2 " with the semantic expansion ": integration of the semantic expansion of the query.

Step 3 "with the filtering of particularly relevant results": integration of the 'additional conditions for the results display'.

Three queries were excluded from the various results presented in section 4.2.1 and section 4.2.2. The justification for this exclusion as well as the results obtained with these three applications are detailed in section 4.2.3.

4.2.1 Impact of the different functions

Each step has been evaluated according to the recall and precision performance criteria. These two measurements are presented in an orthogonal normalized Cartesian graph shown in Fig. 5. On this graph, the ideal situation is then the 1:1 coordinates and the vectors allow visualising the improvement achieved on both criteria after the implementation of each function. Each step allowed for the evolution of one or both of the measurements. Only step 3 of the filtering of less relevant results reduced the recall score by masking relevant desired results. Overall, it seems that, compared to a rudimentary system, the addition of the different functions increased recall from 0.34 to 0.85, so 51 percentage points. The proposal allowed to increase the precision from 0.42 to 0.81, so 39 percentage points. The F-Measure evaluation also shows that despite the reduction in recall in step 3, the overall performance of the system improved with each new feature. Indeed, the F-measure increased from 0.38 to 0.64 to 0.73 to 0.78.

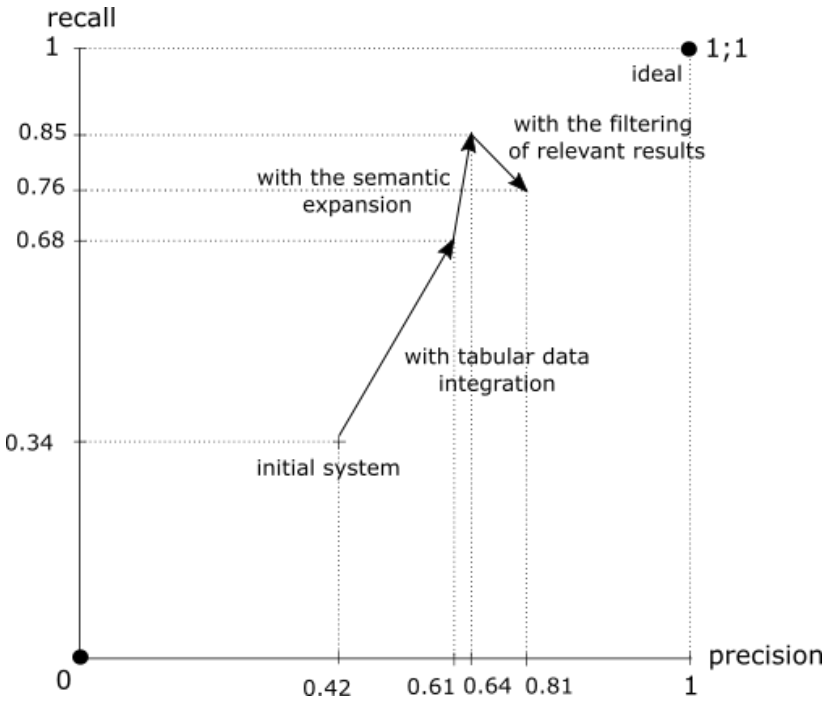


Fig. 5 - Vector representation of the recall and precision evolution according to the steps

Fig. 6. shows the distribution of anomalies at each step. The anomalies called ‘syntax issue’ represent missing results because of the searched content is not exploited in the graph. These anomalies have all been resolved with the integration of table transformations. The anomalies called ‘semantic issue’ represent missing results because the terms used are not strictly the same as in the query. These anomalies have almost all been solved by the integration of the query expansion. Nevertheless, there are still anomalies related to the use of non-obvious abbreviations and compound words missing from knowledge graph. The anomalies called ‘relevance issue’ represent the results answering the query but considered less relevant than those targeted by the user. These anomalies remain the largest number although the integration of the additional rules in step 3 allowed a reduction of 51%. The anomalies called ‘link issue’ represent missing results due to absence of relationships in the graph between distributed data but referring to the same entity in reality. For example, this is the case when searching for ‘price’ information managed by a specific reference known only by the purchasing department and therefore unrelated to the potential terms used in the query. Finally, anomalies called ‘OCR issue’ are related to the non-recognition of text in the image despite the OCR technologies used. It is noted that the anomalies ‘link issue’ and ‘OCR issue’ have not been resolved by the proposal. In addition, it can be noted that step 2 increases the number of anomalies ‘relevance issue’ due to the increase in the number of displayed results it allows for.

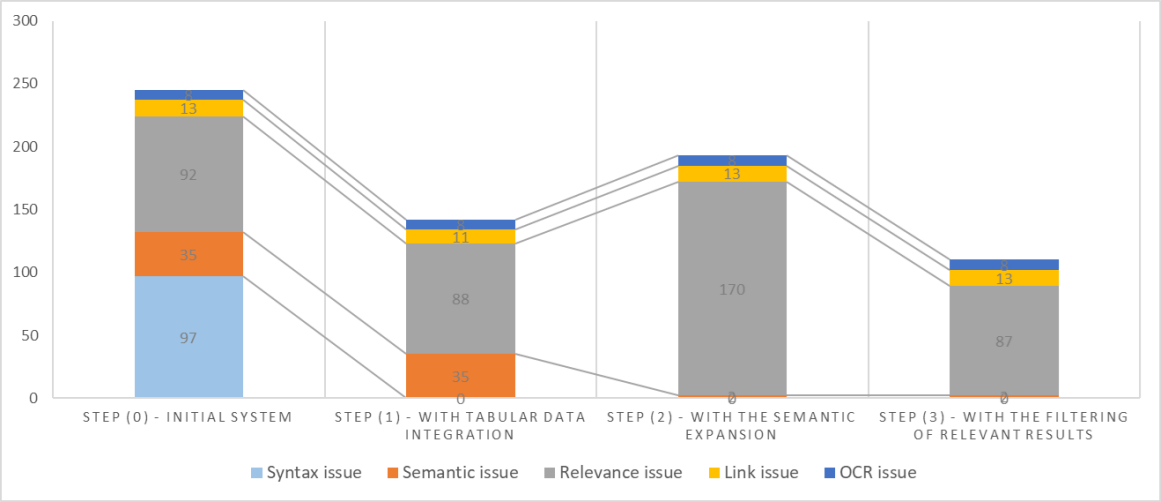


Fig. 6 - Cumulative number distribution of anomalies according to the steps

4.2.2 Impact of the query typology on the category of anomaly

The distribution of anomalies according to the type of query concerned is shown in Fig. 7 and Fig. 8 for step 0 and step 3 respectively. We have determined the score of anomaly distribution according to the type of query concerned with the variation in the number of queries per type. We then used a weighted average whose score expression S is defined as follows :

$$S = 1 * \sum_{i=1}^{4}\left(\frac{N_i}{N} * \frac{Q_{tot}-Q}{Q_{tot}-Q_i}\right) \text{ or } S_i = \frac{N_i}{1-\frac{Q_i}{Q}} * \sum_{i=1}^{4}\left(\frac{1-\frac{Q_i}{Q}}{N_i}\right)$$

where S, Q and N are respectively the score, the number of queries and the number of anomalies for the type of query, Q_{tot} the total number of queries and Q_i and N_i respectively the number of queries and the number of total anomalies for the query type.

The scores representing the remaining anomalies are presented in Table 3 for step 0 and in Table 4 for step 3.

Anomalies related to the performance of OCR tools have been excluded from the following analysis. During step 0, all the query typologies were subject to anomalies and all of them were more or less impacted by the lack of semantic search and the display of less relevant results. Anomalies related to syntax processing problems were mostly distributed over typologies (iii) and (iv) and anomalies related to missing implicit relationships were distributed exclusively over typologies (iii). At step 3 with the whole proposal, the majority of the remaining anomalies related mainly to the typology (iii) and more prominently to the typology (iv), especially on anomalies related to the 'relevant issue' . This fact suggests that future proposals should consider this typology as a priority and then typology (iv) to a lesser extent.

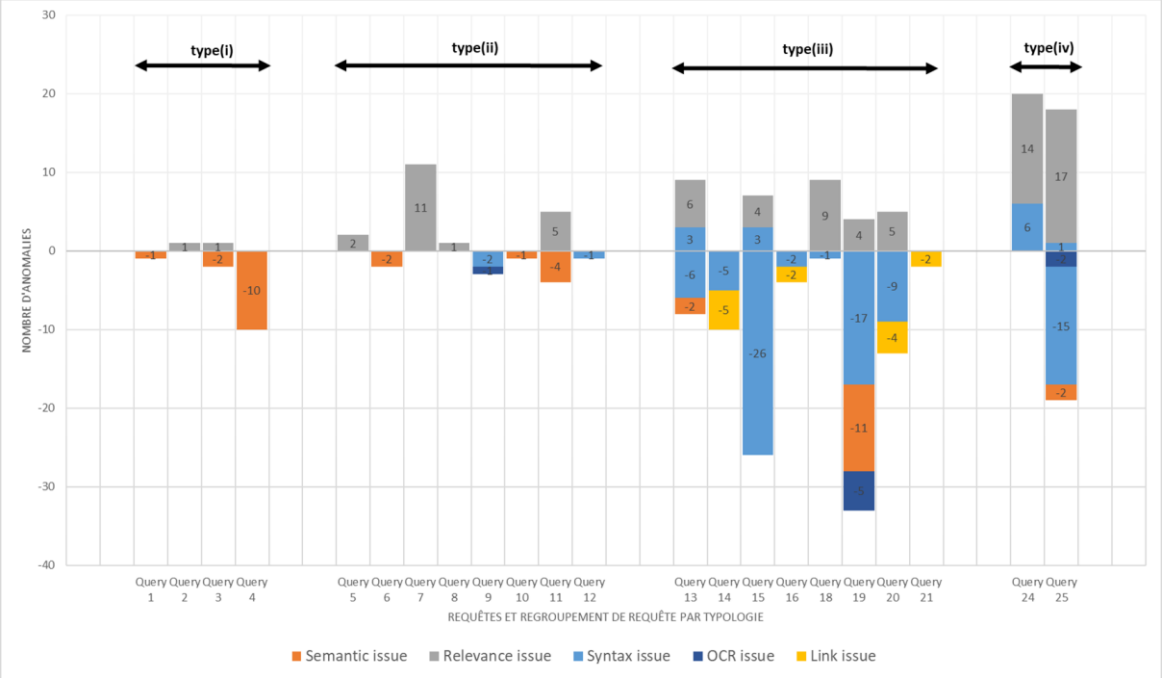


Fig. 7 - Distribution of anomalies by query and their groupings by type in step 0

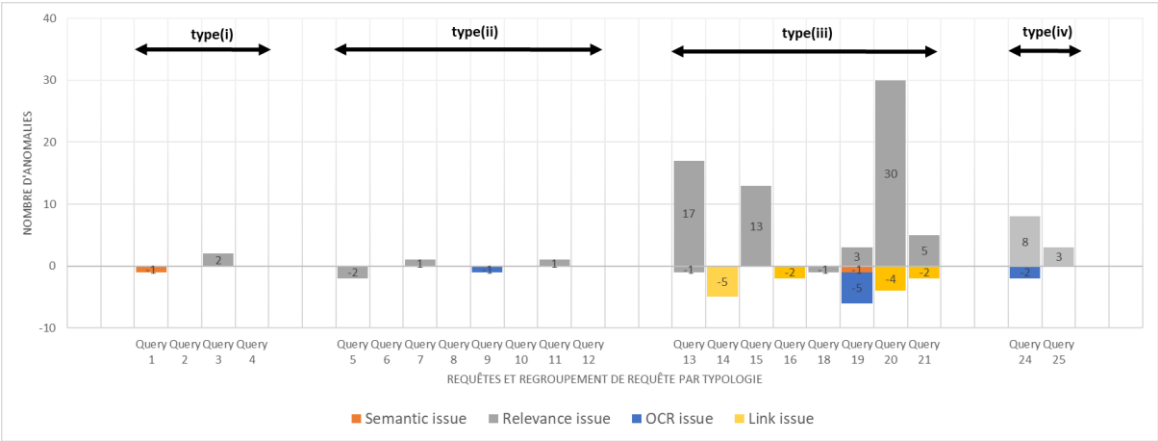


Fig. 8 - Distribution of anomalies by query and their groupings by type in step 3

	Syntax	Semantic	Relevance	Implicit link
Query type (i)	-	0.42	0.03	-
Query type (ii)	0.03	0.18	0.20	-
Query type (iii)	0.68	0.33	0.30	1.00
Query type (iv)	0.30	0.07	0.47	-

Table 3 - Score for the representation of anomalies according to the query type in step 0

	Syntax	Semantic	Relevance	Implicit link
Query type (i)	-	0.56	0.03	-
Query type (ii)	-	-	0.04	-
Query type (iii)	-	0.44	0.76	1.00
Query type (iv)	-	-	0.17	-

Table 4 - Score for the representation of anomalies according to the query type in step 3

4.2.3 Case of queries 17 and 23

Query 17 "Q_w = 'property' and Q_a= 'engine'" has a large number of relevant answers and, from step 0, represents more than 150 anomalies related to the same semantic search problem. The search for the term 'property' did not launch the search for the attributes named "prop".

Query 28 "Q_w = 'speed' and Q_a= 'Serge Bernard'" also represents a large number of relevant answers (sum of speeds recorded by a sensor). From step 0, this query represents more than 700 anomalies related to the input data format. These anomalies are then processed by the proposal. In order to allow a coherent analysis of the distribution of the anomalies, these two queries have therefore been excluded. Nevertheless, their anomalies have been resolved already at step 1 for query 23 and at step 2 for query 17.

To avoid favouring a type of query already represented, query 26 "Q_w = 'choice' and Q_a = 'engine'" was excluded as it has the same results as query 25 "Q_w = 'choice' and Q_a = 'engine'".

5. Discussion

The proposal was evaluated with a data set and queries representative of an industrial case. Each step integrates a new main functionality, which confirmed their relevance either by the gain in completeness or by the gain in precision of the results obtained. Thus, we obtained a gain in recall of 0.51 for a total score of 0.85 and a gain in precision of 0.39 for a total score of 0.81 between a simple information search system and the proposal system. Nevertheless, a significant number of the remaining anomalies come from less relevant results and a minority of others from missing links. Remember that anomalies related to OCR problems are not considered here. An additional analysis according to the typology of queries was also performed. This analysis made it possible to highlight the "search for a specific value" type query. This type has the highest number of anomalies, including the display of less relevant results. It would therefore be wise to isolate this type of query in order to provide appropriate answers and increase the recall score. On the other hand, multiple techniques from web search engine technologies allow optimizing the relevance of the results displayed to the user, either by personalization according to the user, his history and profile or judged by the popularity of the results. In addition, other anomalies have not been addressed such as those related to the absence of implicit relationships between the different data. The detection of these implicit links can be integrated into the data matching domain known under several terms as record linkage, entity matching or entity resolution too (Talburt, 2011) and whose process can be decomposed by analysing the data model and then matching the records to finally lead to an automatic classification exercise (Christen, 2012). To a lesser extent, remaining anomalies in the semantic approach are highlighted

such as the use of abbreviations that are difficult to detect or the use of compound terms. It may be envisaged here to enrich incoming lexical resources or to integrate new domain-specific lexical resources, including for example abbreviations known only within a company.

Apart from these possible improvements for the treatment of the various remaining anomalies and the treatment of functionalities necessary for the deployment of such a solution in a real size industrial case (automation of all accesses and transformation, reconciliation of data over time with maturity and obsolescence management, rights management and improvement of the response time at scale), we distinguish three main families of possible perspectives of improvement of the proposal. Firstly, it is possible to enrich the data graph with new data and their syntactic specificities. We can mention for example semi-structured documents such as XML whose labelling could then be exploited, 3D geometries, images or videos which would allow to search by shape or even the textual style to distinguish titles and paragraphs. The enrichment of the graph can also be envisaged by the automatic or not labelling of nodes or properties. This classification could improve the search and exploration capabilities of the data. Secondly, it is possible to propose a personalisation of the search according to the user and his profile, integrating his own experience in terms of search and navigation history as well as that of his colleagues. Finally, it is also possible to integrate new interrogation methods in order to tend, for example, towards the natural expression of the need for information, to allow more complex interrogation requests of the elements of the graph as well as the analysis of the graph, to allow the exploration of the tree structure directly by the user or finally to automate the requests of an external information system.

6. Conclusion

In this paper, the authors proposed an information retrieval system named i-DATAQUEST to answer the research question "how to return exhaustive and relevant information in a distributed, heterogeneous and relational context". The results showed that the proposal integrating both a data graph to model the data and their syntactic specificities and a graph for the semantic extension of queries allows answering a varied information retrieval. The use of two variables to define queries allows searching a list of elements but also specific values or sentences contained in structured or unstructured data. The results also show the interest of adding an automatic link detection function and of improving the evaluation of the results' relevance.

Acknowledgements

This work was supported by CapGemini company registered within the framework of the “PLM of the future” chair, carried in Arts et Métiers Institute of Technology by P. Véron. and F. Segonds.

References

Abadi, D. J., 2008. *Query execution in column-oriented database systems,* Massachusetts Institute of Technology, USA: s.n.

Alassad, M., Spann, B. & Agarwal, N., 2021. Combining advanced computational social science and graph theoretic techniques to reveal adversarial information operations. *Information Processing & Management,* 58(1), p. 102385.

Angles, R. & Gutierrez, C., 2008. Survey of graph database models. *ACM Computing Surveys (CSUR),* 40(1), pp. 1-39.

Azad, H. K. & Deepak, A., 2019. Query expansion techniques for information retrieval: a survey. *Information Processing & Management,* 56(5), pp. 1698-1735.

Bader, S. R. et al., 2020. A knowledge graph for industry 4.0. *European Semantic Web Conference,* pp. 465-480.

Barsalou, M. A., 2014. *Root Cause Analysis: A Step-By-Step Guide to Using the Right Tool at the Right Time.* Boca Raton: CRC Press.

Batra, S. & Tyagi, C., 2012. Comparative analysis of relational and graph databases. *International Journal of Soft Computing and Engineering (IJSCE),* 2(2), pp. 509--512.

Beisheim, N., Kiesel, M. & Rudolph, S., 2018. Digital manufacturing and virtual commissioning of intelligent factories and Industry 4.0 systems using graph-based design languages. *Transdisciplinary Engineering Methods for Social Innovation of Industry,* Volume 4, pp. 93-102.

Berven, A. et al., 2020. A knowledge-graph platform for newsrooms. *Computers in Industry,* Volume 123, p. 103321.

Buchgeher, G., Gabauer, D., Martinez-Gil, J. & Ehrlinger, L., 2021. Knowledge Graphs in Manufacturing and Production: A Systematic Literature Review. *IEEE Access,* Volume 9, pp. 55537-5554.

Chowdhary, K. R., 2020. Natural language processing. In: *Fundamentals of Artificial Intelligence.* s.l.:Springer, New Delhi, pp. 603-649.

Christen, P., 2012. The Data Matching Process. In: *Data Matching.* s.l.:Springer, Berlin, Heidelberg, pp. 22-35.

Dakiche, N., Tayeb, F. B. S., Slimani, Y. & Benatchba, K., 2019. Tracking community evolution in social networks: A survey. *Information Processing & Management,* 56(3), pp. 1084-1102.

Dawood, H. A., 2014. *Graph theory and cyber security.* s.l.:IEEE.

Diogo, F. & Jorge, B., 2018. Graph Databases Comparison: AllegroGraph, ArangoDB, InfiniteGraph, Neo4J, and OrientDB. In: *Proceedings of the 7th International Conference on Data Science, Technology and Applications, DATA 2018.* Porto, Portugal: s.n., pp. 373-380.

Dou, D., Wang, H. & Liu, H., 2015. *Semantic data mining: A survey of ontology-based approaches.* s.l.:IEEE.

Ehrlinger, L. & Wöß, W., 2016. Towards a Definition of Knowledge Graphs. *SEMANTiCS (Posters, Demos, SuCCESS),* Volume 48, pp. 1-4.

Emmott, S., Alaybeyi, S. & Mullen, A., 2019. *Magic Quadrant for Insight Engines.* [Online]
Available at: https://www.gartner.com/en/documents/3961025/magic-quadrant-for-insight-engines

Gröger, C., Schwarz, H. & Bernhard, M., 2014. *The deep data warehouse: link-based integration and enrichment of warehouse data and unstructured content.* s.l.:IEEE.

Guo, W., Tian, Q., Jiang, Z. & Wang, H., 2018. A graph-based cost model for supply chain reconfiguration. *Journal of manufacturing systems,* Volume 48, pp. 55-63.

hedberg Jr, T. S., Bajaj, M. & Camelio, J. A., 2020. Using graphs to link data across the product lifecycle for enabling smart manufacturing digital threads. *Journal of computing and information science in engineering,* 20(1), p. 011011.

Henkel, R., Wolkenhauer, O. & Waltermath, D., 2015. Combining computational models, semantic annotations and simulation experiments in a graph database. *Database,* Volume 2015, pp. 1-16.

Huet, A. et al., 2021. CACDA: A knowledge graph for a context-aware cognitive design assistant. *Computers in Industry,* Volume 125, p. 103377.

Kammler, F., Hagen, S., Brinker, J. & Thomas, O., 2019. Leveraging the value of data-driven service systems in manufacturing: a graph-based approach. In: *Proceedings of the 27th European Conference on Information Systems (ECIS) , Stockholm & Uppsala, Sweden, June 8-14.* s.l.:s.n.

Kim, L. et al., 2020. Essential issues to consider for a manufacturing data query system based on graph. In: *Roucoules L., Paredes M., Eynard B., Morer Camo P., Rizzi C. (eds) Advances on Mechanics, Design Engineering and Manufacturing III. JCM 2020. Lecture Notes in Mechanical Engineering.*. s.l.:Springer, Cham., pp. 374-353.

Kim, L. et al., 2020. i-DATAQUEST: A Proposal for a Manufacturing Data Query System Based on a Graph. In: C. Springer, ed. *IFIP International Conference on Product Lifecycle Management.* s.l.:s.n., pp. 227-238.

Ko, H. et al., 2021. Machine learning and knowledge graph based design rule construction for additive manufacturing. *Additive Manufacturing,* Volume 37, p. 101620.

Kotov, A. & Zhai, C., 2012. *Tapping into knowledge base for concept feedback: leveraging conceptnet to improve search results for difficult queries.* s.l.:s.n.

Kuhn, M. & Franke, J., 2021. Data continuity and traceability in complex manufacturing systems: a graph-based modeling approach. *International Journal of Computer Integrated Manufacturing,* 34(5), pp. 549-566.

Leavitt, N., 2010. Will NoSQL databases live up to their promise?. *Computer,* 43(2), pp. 12-14.

Leng, J. & Jiang, P., 2019. Dynamic scheduling in RFID-driven discrete manufacturing system by using multi-layer network metrics as heuristic information. *Journal of Intelligent Manufacturing,* 30(3), pp. 979-994.

Lin, J.-R., 2020. *OpenBridgeGraph: Integrating Open Government Data for Bridge Management.* s.l.:s.n.

Li, X., Sun, C. & Zia, M. A., 2020. Social influence based community detection in event-based social networks. *Information Processing & Management,* 57(6), p. 102353.

Martínez-Bazan, N. et al., 2007. Dex: high-performance exploration on large graphs for information retrieval. In: *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management.* s.l.:s.n., pp. 573-582.

Meškelė, D. & Frasincar, F., 2020. ALDONAr: A hybrid solution for sentence-level aspect-based sentiment analysis using a lexicalized domain ontology and a regularized neural attention model. *Information Processing & Management,* 57(3), p. 102211.

Moniruzzaman, A. & Hossain, S. A., 2013. Nosql database: New era of databases for big data analytics-classification, characteristics and comparison. *International Journal of Database Theory and Application,* 6(4).

Mordinyi, R., Schindler, P. & Biffle, S., 2015. *Evaluation of NoSQL graph databases for querying and versioning of engineering data in multi-disciplinary engineering environments.* s.l.:IEEE.

Munir, S., Jami, S. I. & Wasi, S., 2020. Knowledge graph based semantic modeling for profiling in Industry 4.0. *International Journal on Information Technologies & Security,* 121(1).

Nayak, A., Poriya, A. & Poojary, D., 2013. Type of NOSQL Databases and its Comparison with Relational Databases. *International Journal of Applied Information Systems (UAIS),* 5(4), pp. 16-19.

Noel, S. et al., 2016. CyGraph: graph-based analytics and visualization for cybersecurity. *Handbook of Statistics,* Volume 35, pp. 117-167.

Patel, M. & Patel, D. B., 2020. Progressive Growth of ETL Tools: A Literature Review of Past to Equip Future. *Rising Threats in Expert Applications and Solutions,* pp. 389-398.

Patil, N., Kiran, P., Kavya, N. & Naresh, P., 2018. A survey on graph database management techniques for huge unstructured data. *International Journal of Electrical and Computer Engineering,* 8(2), p. 1140.

Pinquié, R., Véron, P. & Croué, N., 2016. Requirement mining for model-based product design. *International Journal of Product Lifecycle Management,* 9(4), pp. 305-332.

Qaise, S. & Ramsha, A., 2018. Text mining: use of TF-IDF to examine the relevance of words to documents. *International Journal of Computer Applications,* 181(1), pp. 25-29.

Qiao, Y. et al., 2020. Heterogeneous graph-based joint representation learning for users and POIs in location-based social network. *Information Processing & Management,* 57(2), p. 102151.

Raza, M. A. et al., 2019. A taxonomy and survey of semantic approaches for query expansion. *IEEE Access,* Volume 7, pp. 17823-17833.

Reinsel, D., Gantz, J. & Rydning, J., 2018. *The Digitization of the World From Edge to Core.* [Online] Available at: https://www.seagate.com/files/www-content/our-story/trends/files/idc-seagate-dataage-whitepaper.pdf

Sabou, M., Ekaputra, F. J. & Biffle, S., 2017. Semantic web technologies for data integration in multi-disciplinary engineering. In: *Multi-disciplinary engineering for cyber-physical production systems.* s.l.:Springer, Cham, pp. 301-329.

Schabus, S. & Scholz, J., 2017. Spatially-Linked Manufacturing Data to Support Data Analysis. *Journal for Geographic Information Science,* 15(1), pp. 126-140.

Talburt, J. R., 2011. *Entity resolution and information quality.* San Francisco, CA: Elsevier.

Wu, Z. et al., 2018. Semantic hyper-graph-based knowledge representation architecture for complex product development. *Computers in Industry,* 100(10.1016/j.compind.2018.04.008.), pp. 43-56.

Yang, l., Cormican, K. & Yu, M., 2019. Ontology-based systems engineering: A state-of-the-art review. *Computers in Industry,* Volume 111, pp. 148-171.

Yoon, B.-H., Kim, S.-K. & Kim, S.-Y., 2017. Use of Graph Database for the Integration of Heterogeneous Biological Data. *Genomics & informatics,* 15(1), pp. 19-27.

Zhou, B., Bao, J., Chen, Z. & Liu, Y., 2021. KGAssembly: Knowledge graph-driven assembly process generation and evaluation for complex components. *International Journal of Computer Integrated Manufacturing,* pp. 1-21.

Zhou, B. et al., 2021. A novel knowledge graph-based optimization approach for resource allocation in discrete manufacturing workshops. *Robotics and Computer-Integrated Manufacturing,* Volume 71, p. 102160.

Zhou, W. & Han, W., 2019. Personalized recommendation via user preference matching. *Information Processing & Management,* 56(3), pp. 955-968.