Abstract

Introduction

Many diseases require careful examination of histopathological material as part of the diagnostic process, so that patients can receive proper treatment. This is especially valid for almost all forms of human cancer, where discovering the disease in a very early stage of its development is crucial for the success of its treatment. This leads to the need to take biopsy materials for examination early in the disease progression, when the signals are still weak and difficult to detect. However, if the cancer is diagnosed in an early stage, its treatment is most likely to be successful, less invasive and more efficient overall.

There are new technology achievements allowing automation in preprocessing, slicing and digitally scanning of tissue examples, moving the action from the world of microscopes to the world of digital medicine. However, this doesn't significantly reduce the volume of work for histopathologists, as they still need to manually examine all slices and determine a potential problem and what its severity is.

On the other hand, while the volume of work is steadily increasing, the number of well-trained experts in histopathology remains the same.

This is where projects like ExaMode[1] can help. The purpose of ExaMode is to evaluate and classify the input information about a specific clinical case (histopathological slide images, the patient's health record, a synopsis from similar cases from the current clinical practice and from literature) in order to automatically decide the presence of specific malignancy and its severity level. The project will achieve that by using collected information from big volumes of scientific publications and clinical cases and applying the latest artificial intelligence (AI) state of the art achievements as well as developing new ones.

There are three data flows that need to be examined:

- histopathological images;
- clinical case synopses;
- other patients related data like age, gender, blood pressure, blood marker results and whatever else information is collected from the hospital medical system and is relevant to the possible interpretation of data and estimation about the user profile.

One of the project objectives is to develop prototypes for image annotations and semantic normalization of textual content from medical reports and scientific publications. The exploitation strategy of the project envisions that, based on the algorithms and prototypes developed in the project, a holistic solution integrating multimodal histopathology data can be built. This solution would support users (histopathologists) when making more informed decisions based on a larger amount of data (judging from similarity to other cases in their clinical practice or the identified likelihood in scientific literature). A prioritization of cases is considered, so that the cases with higher severity and likelihood of specific

[1] EXtreme-scale Analytics via Multimodal Ontology Discovery & Enhancement https://www.examode.eu/

diseases will be presented for confirmation first, and the cases with smaller likelihood – later, when there is enough time.

A consortium between 7 parties was established – two hospitals, two universities and three technological companies/organizations.

The hospitals are the two key stakeholders defining the specific requirements for the digital histopathology workflows and also providing valuable data samples, annotations, etc.:

- Cannizzaro Hospital in Catania;
- Radboud University Medical Center.

The universities are:

- HAUTE ECOLE SPECIALISEE DE SUISSE OCCIDENTALE, Switzerland (HESSO) – with research and development on visual knowledge extraction algorithms and models;
- University of Padova – with research on medical ontologies and semantic mapping between different life-science ontologies and multilingual NLP pipeline for extraction of semantic categories from texts – synopses, diagnoses, etc.

The technological companies are:

- Microscope IT - with expertise in image processing and visualization, which job is the development of prototypes for image annotations and evaluation of images from literature and clinical cases;
- Sirma AI, trading as Ontotext - with expertise in semantic data modelling and processing, which job is creating a Semantic Data Knowledge Management System as a prototype in two versions as well as providing text analysis, NLP, semantic transformations, storage and semantic searches upon stored data;
- SURF Sara - providing super-computer environment and support for training ML models, storing data and real-time environment for the prototypes.

Architecture of the solution

The Semantic Data Knowledge Management System (SDKMS) is the software part, responsible for data transformation in semantic knowledge graphs so that the semantics of different models and data from the different sources can be unified and processed in a holistic way.

A high level architecture diagram of the whole system and the interactions of its parts could be found in Figure 1.

Virtum prototypes store the metadata coming from the analysis of histopathological images to that drive and the files are read by the SDKMS and ingested to the semantic storage - more specifically, to the knowledge graph about the clinical case. After that, the data relevant to a specific clinical case, could be queried by the search services as similarity or faceted search.

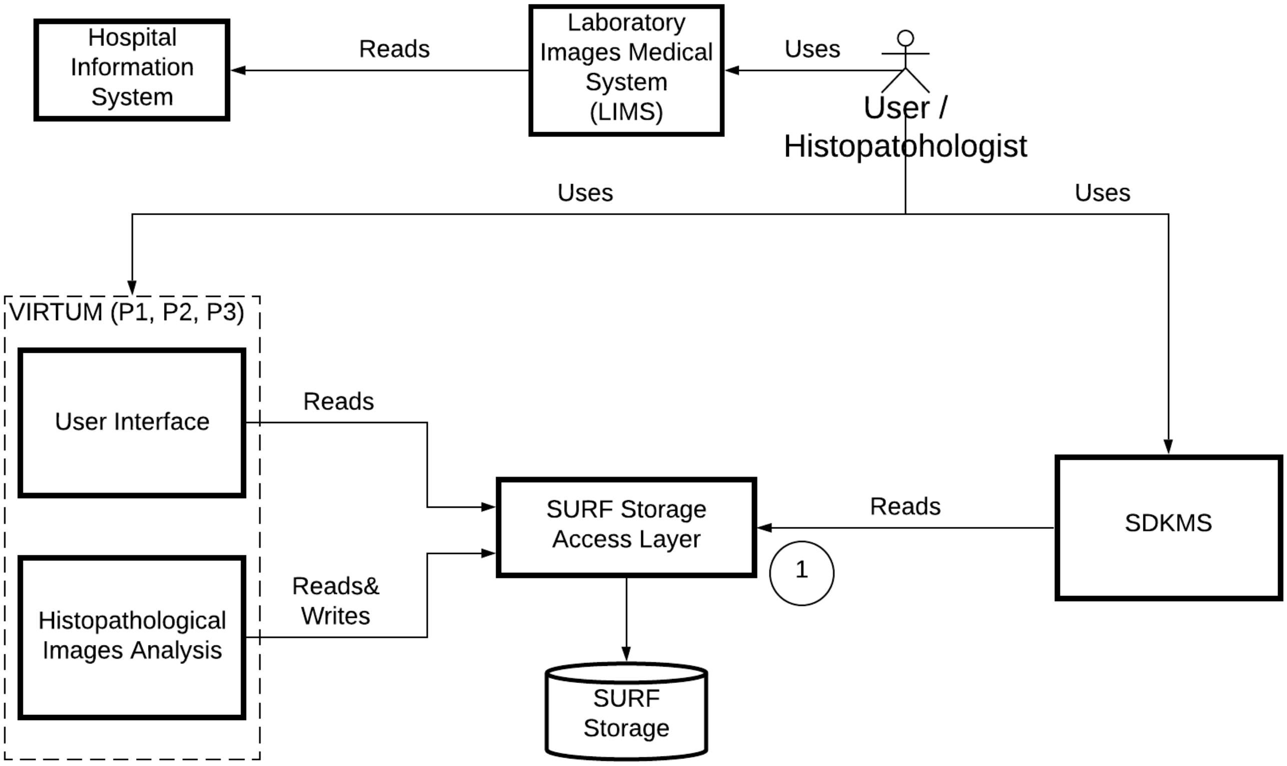

Hospital Information System
Reads
Laboratory Images Medical System (LIMS)
Uses
User / Histopatohologist
Uses
Uses
VIRTUM (P1, P2, P3)
User Interface
Reads
Histopathological Images Analysis
Reads& Writes
SURF Storage Access Layer
1
Reads
SDKMS
SURF Storage

Figure 1 Context diagram of the ExaMode project

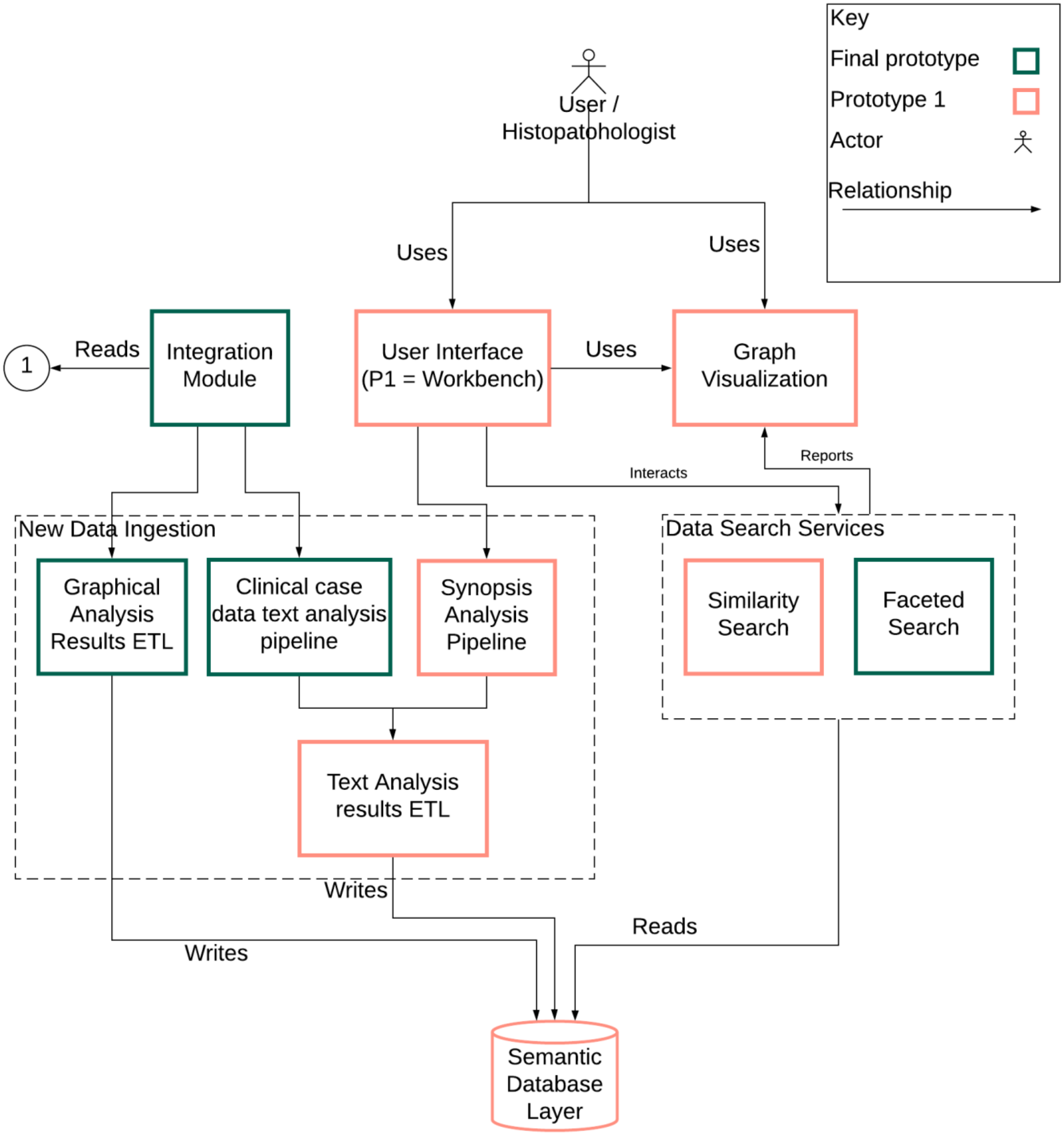


Figure 2 Context diagram of the SDKMS

There are two versions of the SDKMS planned in the scope of the ExaMode project. The first one, which is delivered at the time of writing this document, contains several modules, most of which will be included in the second version. The first version features a text analysis pipeline as part of the ETL

RDFization workflow, which is based on the ExaMode ontology schema developed by the University of Padova. The pipeline covers only one of the topics in the scope of the project (colon cancer) and will be extended to cover the remaining 3 topics.

The second version of the prototype is planned to have its own user interface module and to provide faceted search and more sophisticated algorithms for similarity search.

The high level architecture of the SDKMS is presented in Figure 2.

The boxes are colour-coded – the orange boxes show what is already implemented in the first version, while the green ones - what is planned to be included in the final version of the prototype.

Components used in the first version of the SDKMS

Semantic storage

This component is a bundle of software (GraphDB) and data in the form of stored knowledge graphs. While the first is an of-the-shelf component, providing storage and reasoning functionality as well as some convenient tools for data processing, the second element, the data, is specific for each topic. It is built using domain specific ontologies aligned with the ExaMode semantic model and aims to normalize the textual and visual information for each different case report.

Knowledge graphs in terms of GraphDB

There are plenty of definitions of knowledge graphs with small variations. We will use the definition of Aidan Hogan et al[2], which combines all of the common assumptions. According to it, the knowledge graph is *"a graph of data intended to accumulate and convey knowledge of the real world, whose nodes represent entities of interest and whose edges represent relations between these entities. The graph of data (aka data graph) conforms to a graph-based data model, which may be a directed edge-labelled graph, a property graph, etc. By knowledge, we refer to something that is known . Such knowledge may be accumulated from external sources, or extracted from the knowledge graph itself. Knowledge may be composed of simple statements, such as "Santiago is the capital of Chile", or quantified statements, such as "all capitals are cities". Simple statements can be accumulated as edges in the data graph. If the knowledge graph intends to accumulate quantified statements, a more expressive way to represent knowledge – such as ontologies or rules – is required."*

The purpose is the extraction of data semantics, allowing experts of (big) companies, scientists, journalists, etc. and also the software to manage huge volumes of not-so-structured data and to access the needed information.

GraphDB stores knowledge graphs as sets of RDF triples, where the starting node is the subject, the arc is the property and the ending object (or literal) is the value of the triple.

Usually, knowledge graphs are built on top of some existing structured data, extracted from a database (set of databases), which is interconnected with other semi-structured data (like texts) or completely unstructured data (like images or videos) and connected to standard external ontologies (like FOAF, GeoNames, etc.) for interoperability purposes.

[2] https://arxiv.org/pdf/2003.02320.pdf

If one considers relational database table rows as objects, their columns (fields) as object properties (or predicates), the table contents could be mapped to RDF-triples as <object, predicate, value>, where the values represent the column value for the specific table row. The external links to other tables are represented by placing the objects of the referred table as the values of the triples. This way the relational databases data is mapped to RDF. Moreover, there is no need for all the table fields to have their values – instead, there will be no such triple in the resulting RDF.

However, in order for the above to be a knowledge graph, it must also be easily connectible to common linked open data (LOD) repositories so that a compatibility with the other data sources can be achieved. One of the possible ways to transform relational data is by using commonly known ontologies and their property types as properties for the triples. Another way is by defining mappings and transitional steps from the terms of the specified domain to the LOD.

The common approach for including text in knowledge graphs is by using NLP for detecting semantic categories and their interactions, forming sets of triples and then ingesting them to the semantic storage, together with the original text. There are usually lateral text or JSON storages in addition to the semantic storage.

Images are included in the knowledge graph by extracting metadata from them (by reading their header fields), but mainly by applying AI algorithms for image processing, segmentation, object detection, etc. Then, the extracted image metadata is generalized and transformed to a set of RDF triples, which are further ingested in the semantic storage.

Current ExaMode knowledge graph

Knowledge graphs contain two main components:

- Core ontologies, describing the domain. In the context of the ExaMode project, they are life science and medical ontologies. They are combined in a so-called Annotation Schema, which provides a consistent format for extracted information during the project components execution and references to LOD. Additionally, as per the project design, a set of extracted scientific articles will be processed using the developed NLP and image processing algorithms and the results will be stored in the knowledge graph. This will allow searching for them and accessing them in the context of a specific clinical case, based on the calculated similarity and faceted search algorithms.
- Collected and transformed data about clinical cases, containing information from the electronic health record (EHR) of a specific patient, histopathological images extracted metadata and the physician’s conclusion – synopsis. All these data will be represented as a subgraph by means of the annotation scheme from the previous item.

Currently, the knowledge graph of the project contains RDF-triples about the following ontologies:

- Unified Medical Language System (UMLS) semantic network[3;]
- SNOMED CT[4] – ontology for description of diseases, drugs and EHR;
- PathLex[5] - anatomic pathology lexicon;

[3] https://www.nlm.nih.gov/research/umls/index.html

[4] http://www.snomed.org/

[5] https://bioportal.bioontology.org/ontologies/PATHLEX

- Thesaurus of Prostate Cancer related categories;
- ICD-0-3-T[6] - international classification of diseases;
- BCGO[7] - beta cells genomics ontology;
- Mondo Diseases Ontology[8];
- University of Padova specific ontology for annotating texts from publications;
- Association for Occupational and Environmental Clinics (AOEC) Colon Cancer ontology.

A GraphDB entry point with the installed knowledge graph in its current state can be found on https://examode.ontotext.com/.

One is able to browse data, execute SPARQL queries and use all the installed GraphDB Workbench functionality. The **examode** repository must be selected from the top right selection list.

There are more than 132 million RDF triples currently loaded, of which 80 million are explicit and the other 52 million are inferred.

If one follows the above link and clicks Explore -> Class Hierarchy from the tab in the left pane, it can be seen that the knowledge graph currently contains more than 77 K classes with millions of links between them (see Figure 3).

Further activities are planned in the following directions:

- Enrichment of the knowledge graph by adding specialized ontologies about the rest of the diseases in the project's focus: breast cancer, lungs cancer, etc.
- Increase of interconnections between different subgraphs by increasing the number of mappings between the included ontologies, using sameAs or subClassOf predicates.
- Adding multilingual labels.

The Workbench

The user interface provided on http://examode.ontotext.com is included in the GraphDB distribution and is known as GraphDB Workbench. It is used in the first version of the SDKMS for data entry, queries and visualization purposes.

Although the Workbench is quite powerful and offers general purpose functionality, its use as a UI for the project is limited by the need for users to be familiar with the semantic knowledge domain and to be trained to think in terms of RDF, write/modify SPARQL queries and to interpret the extracted results. Even using predefined queries requires their adoption for specific data to be included as VALUE clauses. Keeping in mind that the users of the SDKMS are doctors, it is not expected of them to be experts in the field of semantic knowledge and ontologies. So there is a specialized UI planned for the development in the second prototype.

Besides the above, the Workbench is going to remain the tool for ingesting data from external sources, e.g., publications and ontologies.

[6] https://bioportal.bioontology.org/ontologies/ICD-O-3-T

[7] https://github.com/obi-bcgo/bcgo

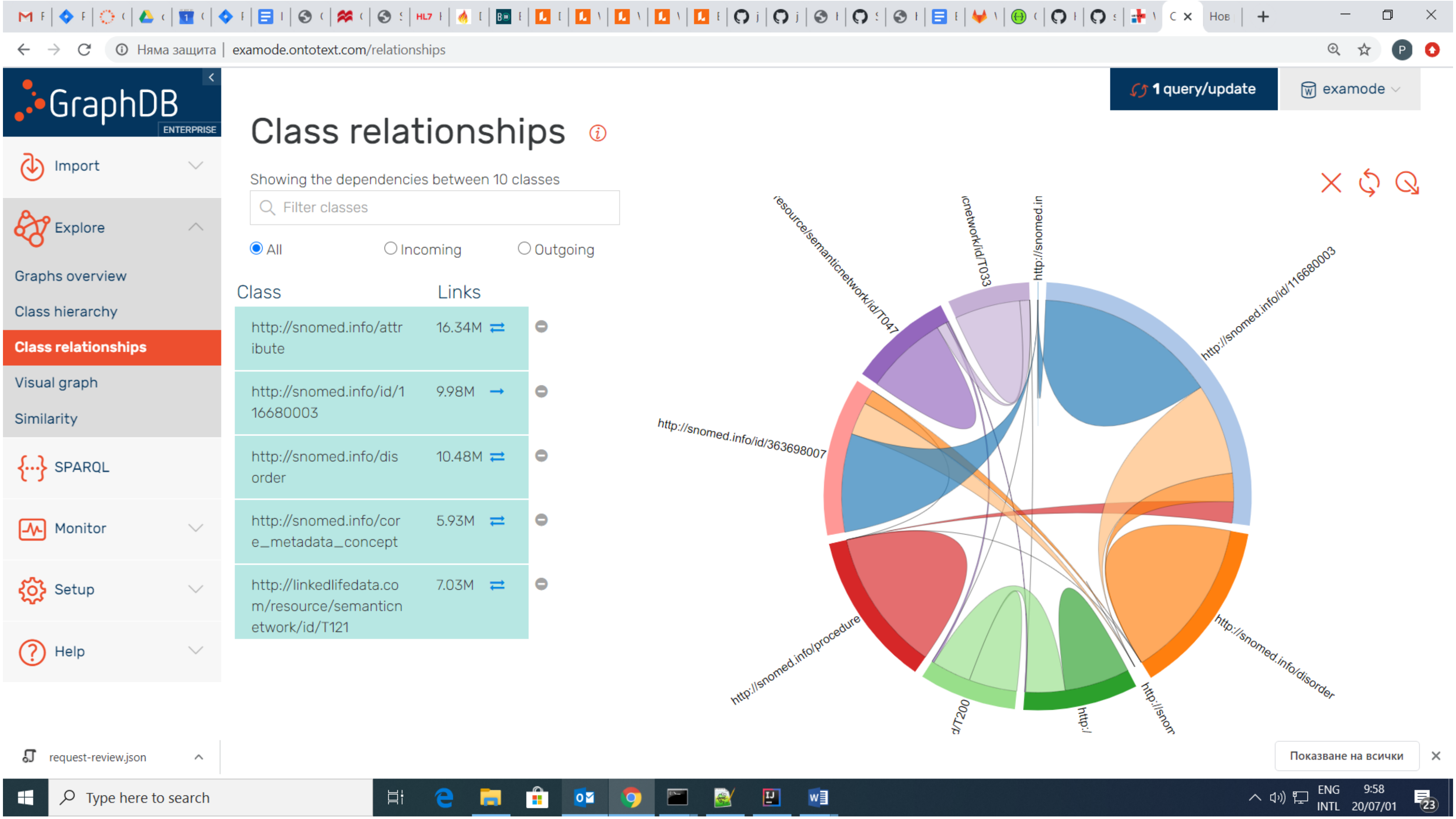
[8] https://mondo.monarchinitiative.org/

Visual graph

The Visual graph, a dedicated GraphDB functionality for visualization, shows query results or specified object links in a way that allows at-a-glance view and better interpretation.

When users select the Visual graph from the left pane of the Workbench, this will open a screen like in Figure 4. The options for how to work with it are the following:

- Placing a link of an object, e.g., http://linkedlifedata.com/resource/umls/id/C0001432 in the Easy-graph field brings results as the ones shown on the Figure 5.



Class relationships

Showing the dependencies between 10 classes

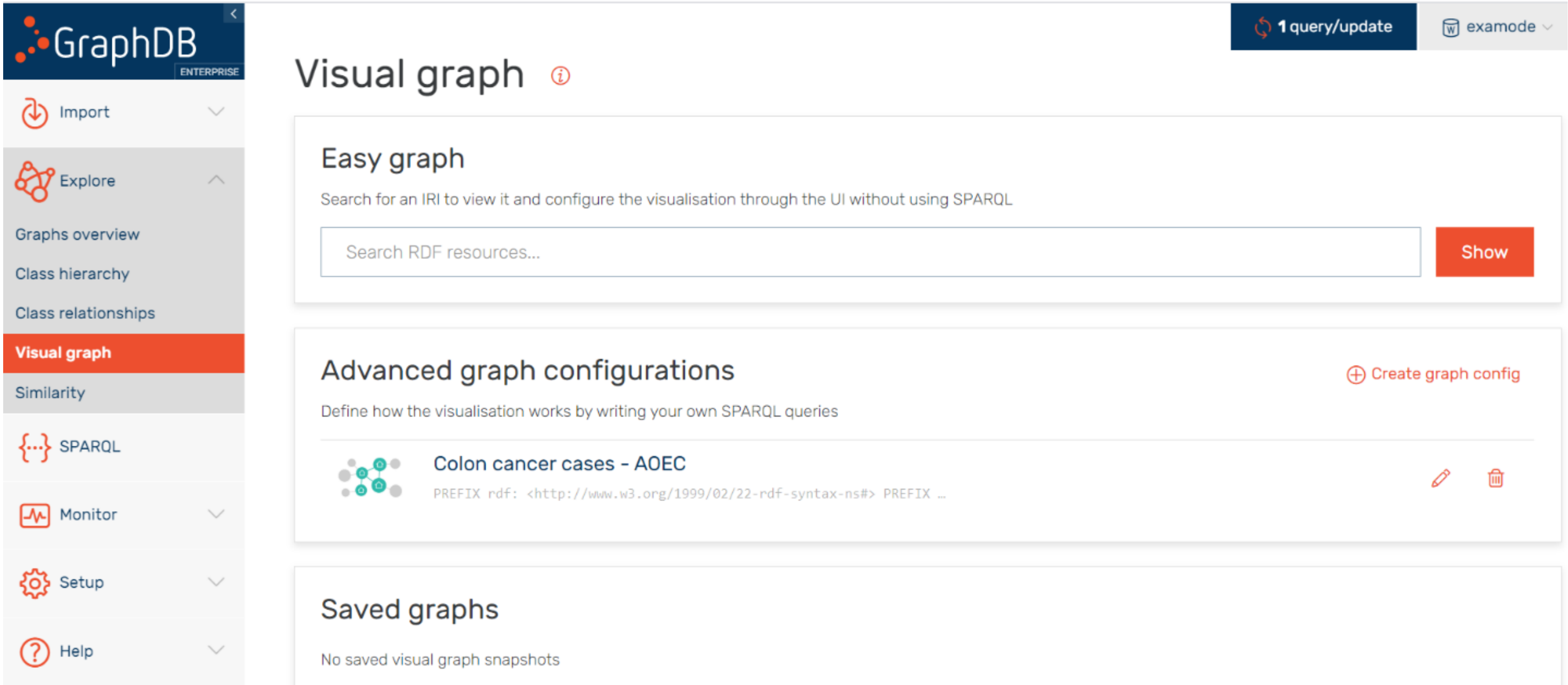
Filter classes

● All ○ Incoming ○ Outgoing

Class	Links
http://snomed.info/attribute	16.34M ⇄
http://snomed.info/id/116680003	9.98M →
http://snomed.info/disorder	10.48M ⇄
http://snomed.info/core_metadata_concept	5.93M ⇄
http://linkedlifedata.com/resource/semanticnetwork/id/T121	7.03M ⇄

resource/semanticnetwork/id/T047
cnetwork/id/T033
http://snomed.in
http://snomed.info/id/116680003
http://snomed.info/id/363698007
http://snomed.info/procedure
d/T200
http:/
http://snom
http://snomed.info/disorder

Figure 3 Class relationships



Visual graph

Easy graph

Search for an IRI to view it and configure the visualisation through the UI without using SPARQL

Advanced graph configurations

Define how the visualisation works by writing your own SPARQL queries

Colon cancer cases - AOEC

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> PREFIX …
```

Saved graphs

No saved visual graph snapshots

Figure 4 Visual graph functionalities

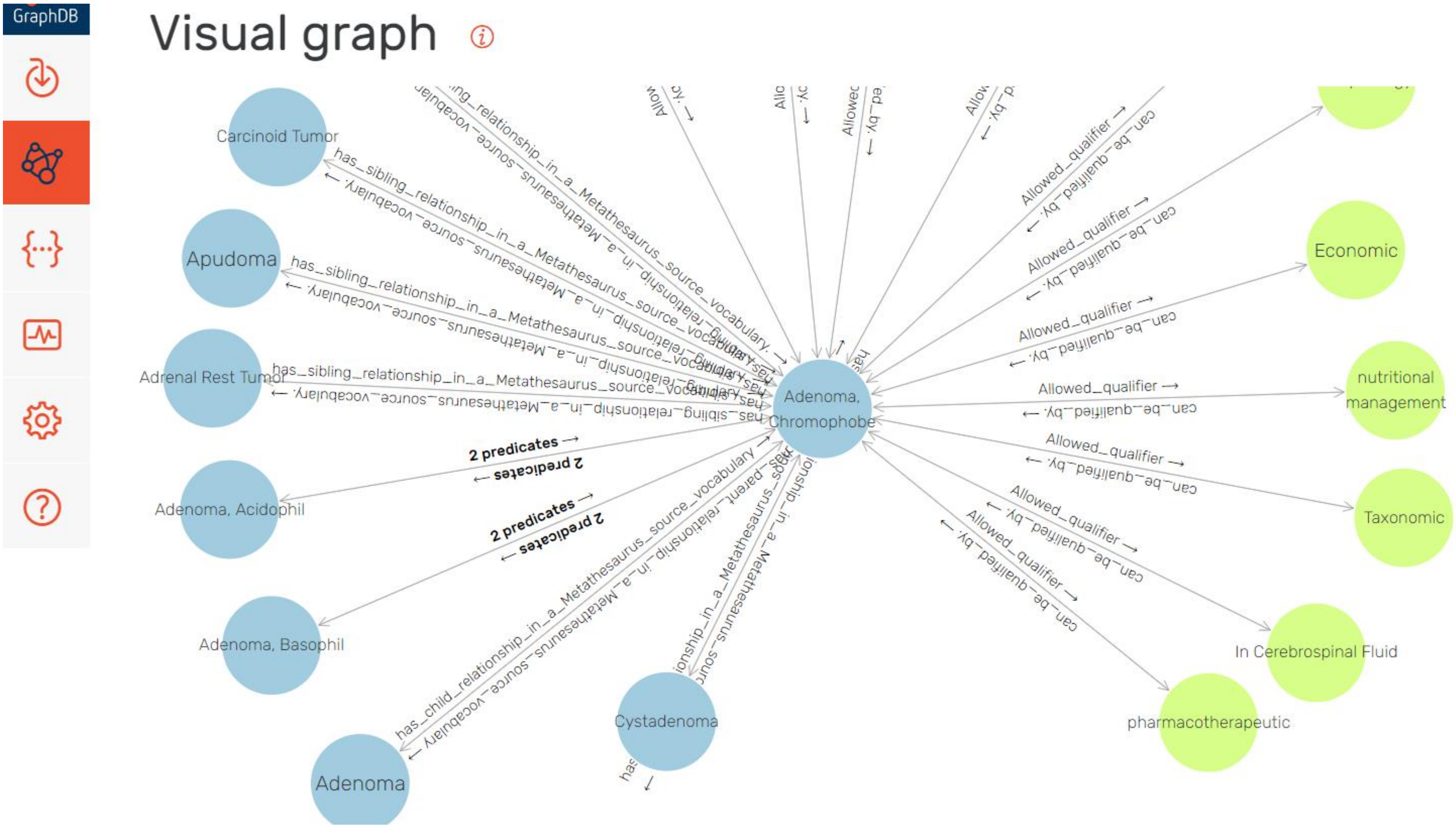


Figure 5 Links of a node

- Using the Advanced graph configurations where users can modify and run some pre-defined queries or write their own ones. See the following query as an example:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX exa: <https://w3id.org/examode/>
CONSTRUCT
{
  ?case exa:hasDiagnosis ?diagnosis ;
      exa:hasLocation ?location ;
      exa:hasDysplasia ?displasia ;
      exa:hasProcedure ?procedure ;
      exa:hasTopography ?topography ;
      exa:hasGender ?gender ;
      exa:hasAge ?age
}
WHERE {
        ?case rdf:type exa:ColonClinicalCaseReport ;
    exa:hasOutcome ?outcome .
  OPTIONAL{graph ?g{?outcome rdf:type ?diagnosis}}
  OPTIONAL{?outcome exa:hasLocation ?location}
  OPTIONAL{?outcome exa:hasDysplasia ?displasia}
  OPTIONAL{graph ?g {?outcome exa:hasIntervention ?intervention .
  ?intervention rdf:type ?procedure ;
          exa:hasTopography ?topography }}
  OPTIONAL{?case exa:hasGender ?gender}
  OPTIONAL{?case exa:hasAge ?age}
} LIMIT 20
```

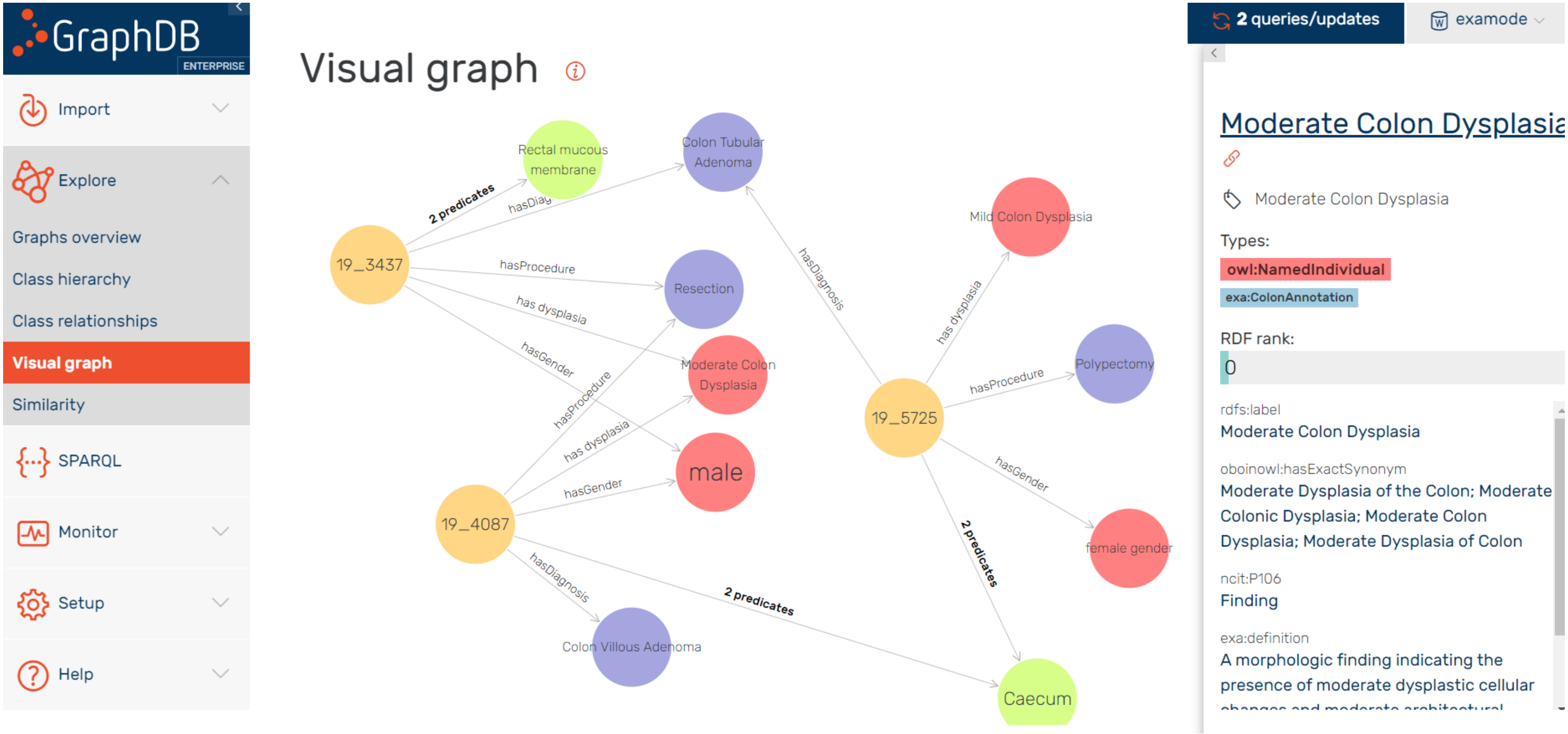
The example query above results in a visualization such as in Figure 6.

The UI allows users to rearrange the view by dragging nodes as well as to click on them to obtain additional information, which opens in a side pane. Users can also expand and collapse nodes that are objects (not literals) to see their links and explore them further.

- The third option (Figure 4) is to explore already saved results from the first two options.

We strongly recommend visiting the link https://examode.ontotext.com and playing with this functionality.

In the first version of the prototype the Visual graph is used “as is” and some queries have been defined so they could be used as a template for search before visualization. The functionality for generating visual graph images is planned to be integrated with the user interface of the second/final prototype so that it is accessible to users in an easy and straightforward way.



Moderate Colon Dysplasia

Moderate Colon Dysplasia

Types:

owl:NamedIndividual

exa:ColonAnnotation

RDF rank:

0

rdfs:label

Moderate Colon Dysplasia

oboinowl:hasExactSynonym

Moderate Dysplasia of the Colon; Moderate Colonic Dysplasia; Moderate Colon Dysplasia; Moderate Dysplasia of Colon

ncit:P106

Finding

exa:definition

A morphologic finding indicating the presence of moderate dysplastic cellular changes and moderate architectural

Figure 6 Query results visualization by visual graphs

Similarity search

The business requirement addressed by the similarity search is that users may need more information about a specific clinical case and the probability of one outcome or another. They can check this either by exploring all cases and finding which are the most similar to theirs or by searching in the stored publications metadata.

There are several challenges related to similarity:

- How to use information that is as complete as possible?
- How to define proper similarity metrics?
- How to perform similarity search in the most effective and productive way?

We address the first problem by combining all three data flows – images analysis and classification results metadata, the provided parts of EHR and the provided synopses. Each clinical case is represented as an object (node) in the knowledge graph and all the known data about it - as connections to this node. In this way, it becomes a part of the knowledge graph and, as it has more properties, it makes the information in the KG more interconnected and more easily accessed by search algorithms.

When a node is connected to other nodes (semantic concepts) of an ontology and one follows its hierarchy and interconnection to other ontologies, its corresponding clinical case becomes connected to the other cases through their links to the same semantic categories. The most intuitive way to define when two nodes are most similar while inspecting their links is by counting the sum of common tokens and normalizing it by the number of tokens.

There are a number of algorithms for text similarity and knowledge graph nodes similarity. In the first version of the SDKMS, we use the Predicate-based Semantic Indexing (PSI) as part of GraphDB. The PSI is an adoption of Random Indexing for application on knowledge graphs and, as a specific case of the Vector Symbolic Architecture (VSA) approach, it is a type of a vector space model.

All vector space models assign vectors to the different elements of a glossary, e.g., to a set of words in a specific set of documents in a way that allows using vector operations to perform complex tasks such as calculating the similarity between documents. In the context of knowledge graphs, all the nodes and relations are terms and obtain their vector representations – indices. Different graphs, subgraphs, etc. are documents

In particular, VSA algorithms assign vectors (indices) of high dimensions (10 000 and above) and define binding operators together with their inverse operators, also called release. Both the binding and the inverse map to the same vector space as the inverse could be not exact, but only approximate. However, in that case, even if the release doesn't map to the exact origin vector, it must be the nearest of all possible results. If we denote the similarity operator with ·, the binding with ⊗ and the inverse with ∅, the following statements are true:

- The binding of two vectors a□b maps to a completely dissimilar vector: a (a□b) H 0 and (a□b) bH0
- (a (a□b)) b H 1

The space dimension and the implementation of binding and release operators vary among the algorithms as well as calculating the similarity that is usually a vector distance function.

The PSI as an adoption of Random Indexing algorithm has the following characteristics:

1. For each node **e** and relation **r** from the knowledge graph, random vectors **I(e)** and **I(r)** are generated. They are both from the same vector space with a high dimension.
2. For each node e_0 a context vector is assigned:

   $$S(e_0) = \sum_{e \in E, r \in R} \quad w(e, r, e_0) I(e) I(r),$$

   where $w(e, r, e_0)$ is 1, if there is such triple(e, r, e_0) in the knowledge graph, and otherwise - 0.

3. A similar approach is taken for the inverse relations r_inv (if any):

   $$S(e) = \sum_{e \in E, r\_inv \in R} \quad w(e_0, r\_inv, e) I(r\_inv) I(e),$$

   where $w(e_0, r\_inv, e)$ is 1, if there is such triple (e_0, r_inv, e) in the knowledge graph, and otherwise - 0.

Although all index vectors **I(e)** and **I(r)** are initialized randomly, the context vectors **S(e)** of objects embed information about their existence as an object in the knowledge graph triples.

The above algorithm works only if the dimension of the vector space is large enough.

The similarity between the two vectors a and b, ***a ·b***, is 1 minus the Humming distance between **S(a)** and **S(b)**.

The main benefits of using PSI for calculating similarities are the following:

- As in all vector space models, the vector operations are quite fast on contemporary CPUs - as almost all of them have built-in vectorization functions, which is much faster than sequences of rule-based if-then-else logic.
- The PSI doesn’t require any optimization or ML steps. The calculation of indices is straightforward and its cost is predictable.

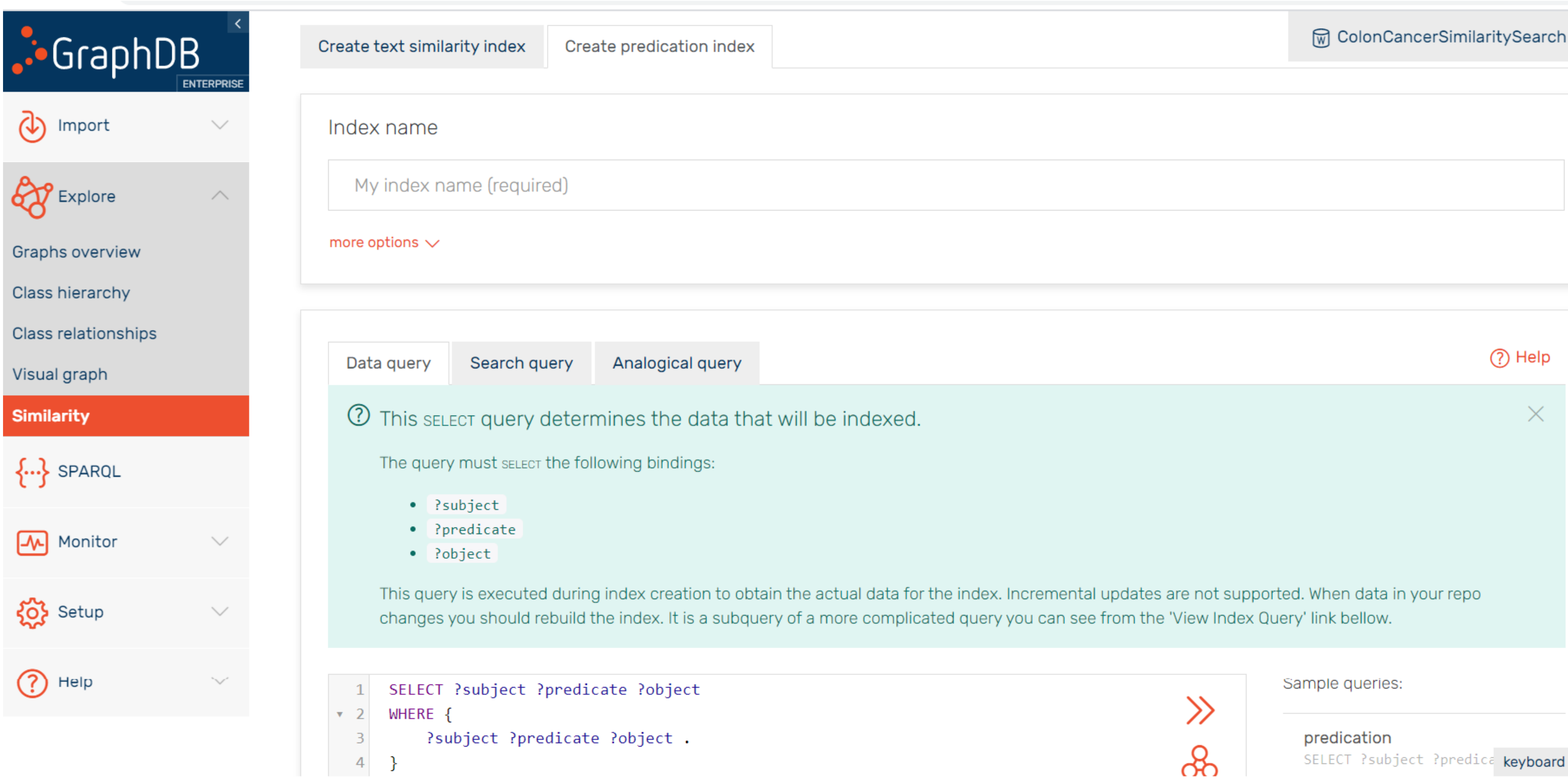
The main drawback of the PSI, which is common for all vector-space models, is that it is a closed system. One cannot estimate the similarity of a non-indexed entity to other indexed entities. However, the fast rebuilding of indices (for the current volume of the ExaMode project knowledge graph, it takes less than a minute on a regular desktop configuration) makes the frequent rebuild possible.

It is also worth noting that the information can be added in batches and after that the index can be rebuilt before exploring new clinical cases.

Similarity search example

Let us visit the link https://examode.ontotext.com where the instance of GraphDB with the loaded knowledge graph of the ExaMode project is. We need to select the repository (ColonCancerSimilaritySearch) and Explore ->Similarity on the left pane.

There are two algorithms provided. The first one is for the textual index, which is the common Random Indexing, but works well only on rich text fields. The other one – the predicate index - is the PSI explained above. A list of already built indices appear and users are able to create their own.



My index name (required)

This SELECT query determines the data that will be indexed.

The query must SELECT the following bindings:

- `?subject`
- `?predicate`
- `?object`

This query is executed during index creation to obtain the actual data for the index. Incremental updates are not supported. When data in your repo changes you should rebuild the index. It is a subquery of a more complicated query you can see from the 'View Index Query' link bellow.

```
SELECT ?subject ?predicate ?object
WHERE {
    ?subject ?predicate ?object .
}
```

```
SELECT ?subject ?predica
```

Figure 7 Creating a predication index

Let us select **Create New Index** and then the tab **Create predicate index**. A screen as the one in Figure 7 appears.

Each index has two queries – a data query for extracting the triples, which will be indexed, and a search query, which defines the use of the index. In the above example, there is no filtering and all the triples in the knowledge graph will be included in the index.

The search query could be as follows:

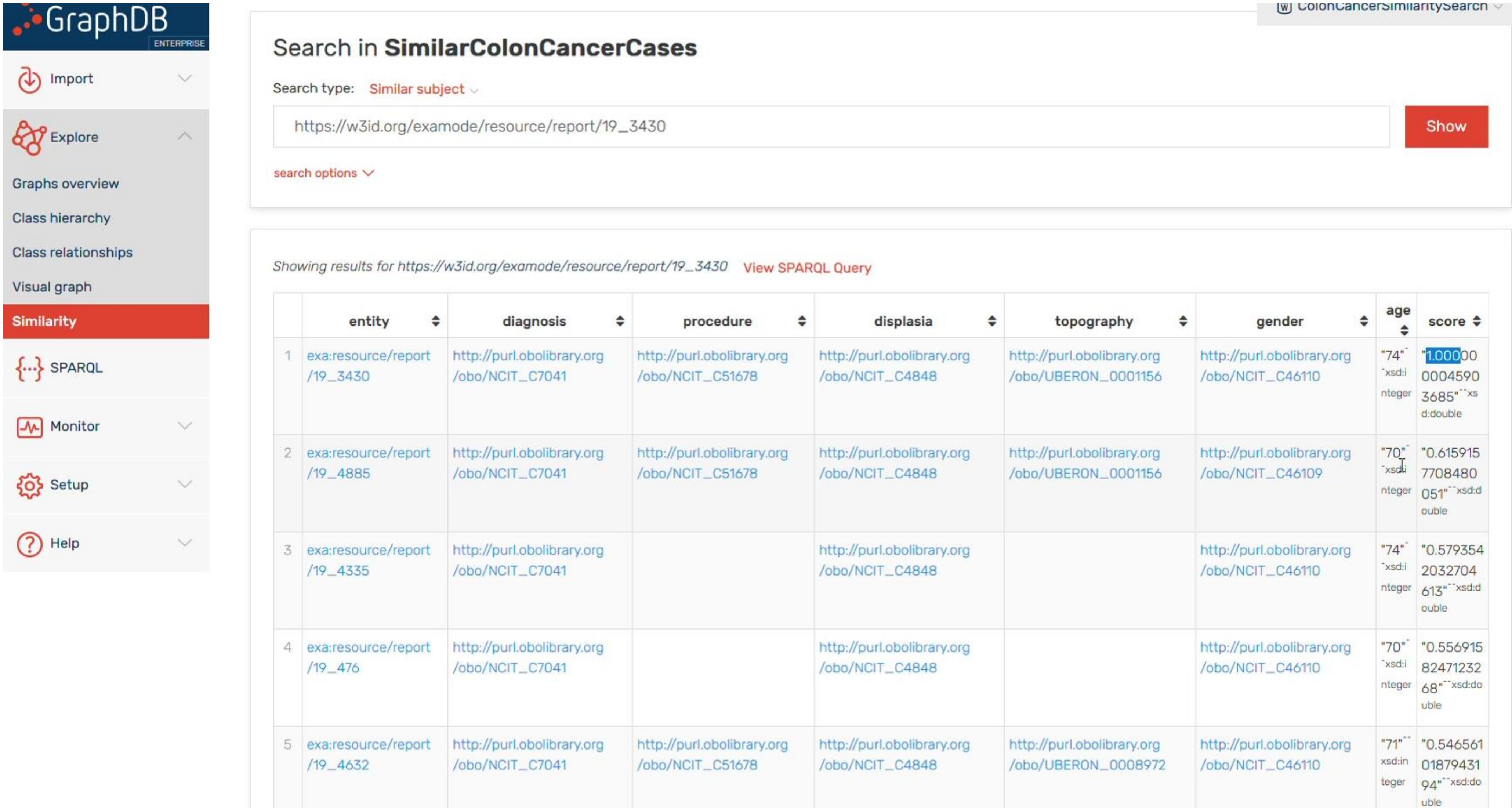
```
PREFIX :<http://www.ontotext.com/graphdb/similarity/>
PREFIX inst:<http://www.ontotext.com/graphdb/similarity/instance/>
PREFIX psi:<http://www.ontotext.com/graphdb/similarity/psi/>

PREFIX exa: <https://w3id.org/examode/>
SELECT ?entity ?diagnosis ?procedure ?displasia ?procedure ?topography ?gender ?age ?score {
        ?search a ?index ;
                ?searchType ?query;
                psi:searchPredicate ?psiPredicate;
                :searchParameters ?parameters;
                ?resultType ?result .
        ?result :value ?entity ;
                :score ?score .
        OPTIONAL{?entity exa:hasDiagnosis ?diagnosis }
        OPTIONAL{?entity exa:hasLocation ?location }
        OPTIONAL{?entity exa:hasDysplasia ?displasia }
        OPTIONAL{?entity exa:hasProcedure ?procedure }
        OPTIONAL{?entity exa:hasTopography ?topography }
        OPTIONAL{?entity exa:hasGender ?gender }
        OPTIONAL{?entity exa:hasAge ?age }
}
```

The result is retrieved with its url ?entity and similarity score ?score. Users can add fields to include more details in the result as shown in the OPTIONAL fields of the example above. To include similarity results with missing values for some of the properties, it is necessary to use OPTIONAL subquery clauses.

To run the query, users must select the URI of an object and click the **Show** button. In the example below, this is https://w3id.org/examode/resource/report/19_3430 (See Figure 8). There, the similarity score is on the furthest column on the right. As one can see, the most similar case is the same as the one queried with similarity score approximately 1.0, as expected.

The second result scores 0.616 and differs from the one in the query only by gender and age. The third result has a similarity score of about 0.58 and has the same property values as the first but two of them have not been provided.

One can visit the site and perform other searches. The exact columns to be displayed are specified in the search query where users can also apply more complex criteria.



Search in **SimilarColonCancerCases**

Search type: Similar subject

https://w3id.org/examode/resource/report/19_3430

Showing results for https://w3id.org/examode/resource/report/19_3430

	entity	diagnosis	procedure	displasia	topography	gender	age	score
1	exa:resource/report/19_3430	http://purl.obolibrary.org/obo/NCIT_C7041	http://purl.obolibrary.org/obo/NCIT_C51678	http://purl.obolibrary.org/obo/NCIT_C4848	http://purl.obolibrary.org/obo/UBERON_0001156	http://purl.obolibrary.org/obo/NCIT_C46110	"74"^^xsd:integer	"1.0000000045903685"^^xsd:double
2	exa:resource/report/19_4885	http://purl.obolibrary.org/obo/NCIT_C7041	http://purl.obolibrary.org/obo/NCIT_C51678	http://purl.obolibrary.org/obo/NCIT_C4848	http://purl.obolibrary.org/obo/UBERON_0001156	http://purl.obolibrary.org/obo/NCIT_C46109	"70"^^xsd:integer	"0.6159157708480051"^^xsd:double
3	exa:resource/report/19_4335	http://purl.obolibrary.org/obo/NCIT_C7041		http://purl.obolibrary.org/obo/NCIT_C4848		http://purl.obolibrary.org/obo/NCIT_C46110	"74"^^xsd:integer	"0.5793542032704613"^^xsd:double
4	exa:resource/report/19_476	http://purl.obolibrary.org/obo/NCIT_C7041		http://purl.obolibrary.org/obo/NCIT_C4848		http://purl.obolibrary.org/obo/NCIT_C46110	"70"^^xsd:integer	"0.5569158247123268"^^xsd:double
5	exa:resource/report/19_4632	http://purl.obolibrary.org/obo/NCIT_C7041	http://purl.obolibrary.org/obo/NCIT_C51678	http://purl.obolibrary.org/obo/NCIT_C4848	http://purl.obolibrary.org/obo/UBERON_0008972	http://purl.obolibrary.org/obo/NCIT_C46110	"71"^^xsd:integer	"0.5465610187943194"^^xsd:double

Figure 8 Similarity search results

Big data aspects of the SDKMS (Conclusion)

The SDKMS is a significant part of the ExaMode project as it is one of the two commercial software prototypes that will result from it. The project is in development under the EU Horizon-2020 call in Topic ICT-12 RIA in the field of Big Data technologies.

ExaMode addresses exa-scale data from various sources - scientific literature and clinical research on histopathological images of human tissues samples relevant to different diseases where such an examination approach is applicable.

There are two main complementary work streams. One is to use the huge collected volume of visual and textual data for training Machine Learning models to detect specific visual configurations (artefacts) in images as well as to extract the semantics from textual descriptions. The second is to develop prototypes that utilize these models and to provide histopathologists with the functionality to automate and prioritize their everyday activities, allowing them to examine higher volumes of samples with better accuracy and productivity. The prototypes will be implemented first in a hospital environment in Azienda Ospedaliera per l'Emergenza Cannizzaro hospital in Italy.

The SDKMS will utilize the ML models trained within the scope of the project. It will feature the workflows of extracting semantics from textual input data (e.g., synopses) and will unite them with the outcomes of histopathological images analysis. Following this approach, a multi-modal semantic representation of each specific clinical case will be generated. Each case constitutes a knowledge graph, which is mapped to other well known ontologies in the health-care field like SNOMED CT, MONDO, etc. and uses data representation standards like HL7 and FHIR. This connects the clinical case related data to LOD sources in the world as well as to all case-relevant scientific and clinical research reports.

The user of the system (a histopathologist) will be able to search for similarities and will receive information about the distribution of the specific case characteristics in the collected data as well as in the literature, which matches one of the specific requirements of ICT-12 RIA "**new methods for extreme-scale analytics, deep analysis, precise predictions and decision making support".**

In addition to the above, the SDKMS will provide a novel tool for knowledge graph visualization in a scalable and user friendly way, which is relevant to the requirement **"novel visualization techniques"**.

Appendix 1. List of abbreviations

AI	Artificial Intelligence
EHR	Electronic Health Record
HIS	Hospital Information System
KG	Knowledge Graph
LIMS	Laboratory Images Management System
LOD	Linked Open Data
NLP	Natural Language Processing -a complex of algorithms and AI models for extracting semantic categories and meanings from free style text
PSI	Predicate-based Semantic Indexing
RDF	Resource Description Framework – standard for representation of semantic knowledge
SDKMS	Semantic Data Knowledge Management System – this software system and its versions
UI	User Interface
VSA	Vector Symbolic Architecture