Original paper

Searching COVID-19 clinical research using graph queries

Francesco Invernici, Anna Bernasconi, and Stefano Ceri
Department of Electronics, Information, and Bioengineering, Politecnico di Milano, Milan, Italy

Corresponding Author: Anna Bernasconi, *Department of Electronics, Information, and Bioengineering, Politecnico di Milano* – Via Ponzio 34/5, Milan, Italy. Phone: +39-02-23993494. Fax: +39-02-23993411, E-mail: anna.bernasconi@polimi.it

Abstract

Background: Since the beginning of the COVID-19 pandemic, more than one million studies have been collected within the COVID-19 Open Research Dataset Challenge (CORD-19), a corpus of manuscripts created to accelerate the research against the disease. Their related abstracts represent a wealth of information, which is – in many cases – yet to be explored, as well as unstructured and thus hardly searchable. Keyword-based search is the standard approach, which allows users to retrieve the documents of a corpus that contain (all or some of) the words in a target list. This kind of search, however, does not provide visual support to the task and is not suited to expressing complex queries, nor to compensating for missing specifications.
Objectives: As graphs are increasingly used to represent and query scientific knowledge, this paper proposes to consider small graphs of concepts and exploit them for expressing graph searches over existing COVID-19-related literature, providing a user-friendly search and exploration experience.
Methods: We considered the CORD-19 corpus and summarized its content by annotating publications' abstracts using terms selected from the Unified Medical Language System (UMLS) and the Ontology of Coronavirus Infectious Disease (CIDO). Then, we built a co-occurrence network that includes all relevant concepts mentioned in the corpus, establishing connections when their mutual information is relevant. A sophisticated graph query engine was built to allow the identification of the best matches of graph queries on the network, allowing as well partial matches and proposing candidate query completions (through shortest paths).
Results: We built a large co-occurrence network, consisting of 128,249 entities and 47,198,965 relationships; the GRAPH-SEARCH interface allows users to explore the network by formulating or adapting graph queries; it produces a bibliography of publications, globally ranked; each publication is further associated with the specific parts of the query that it explains, thereby allowing the user to understand each aspect of the matching.
Conclusions: Our approach supports the process of query formulation and evidence search upon a large text corpus; it can be reapplied to any scientific domain where documents corpora and curated ontologies are made available.

Keywords: Big data corpus; Clinical research; Co-occurrence network; CORD-19; Graph search; Named entity recognition; Neo4j; Text mining.

Introduction

Since the COVID-19 outbreak in early 2020, important clinical research efforts have been targeted at understanding the COVID-19 disease. More than one million studies have been collected within the COVID-19 Open Research Dataset Challenge (CORD-19), a corpus of manuscripts created to accelerate the research against the disease. Their related abstracts represent a wealth of information, which is -however- unstructured and thus hardly accessible or searchable.

Searching over literature is a non-trivial task, as it strongly relies on the quality of the data corpus, the characteristics of the search portal, and the language used to express the search. Keyword-based search is the standard search approach, which allows users to retrieve the documents of a corpus that contain some of the words in a specified target list [1], [2]. This kind of search, however, does not provide visual support to the task and is not suited to expressing complex research queries, nor to compensating for missing specifications.

The development of frontend tools and visualizations for COVID-19 knowledge graphs has been motivated by several [3], [4]. We then explored the use of small graph-based queries that can be built visually [5] to empower a literature-exploration tool: the GRAPH-SEARCH system stems from this motivation, providing both a visual language to express search queries and a friendly tool to explore relevant publications, which highlights the relationships between the original graph queries and an underlying corpus of scientific evidence, in the spirit of literature-based discovery [6].

In order to support this idea, the underlying textual corpus must be first analyzed and enriched; in our approach, the CORD-19 dataset was expressed in the form of a co-occurrence network. First, we annotated all the abstracts with terms from the Unified Medical Language System (UMLS, [7]) and the Ontology of Coronavirus Infectious Disease (CIDO, [8]). This step was much in line with classical work on ontology-based annotation (see Semantic MEDLINE [9] and our own work on genomic metadata annotation [10], [11]). Second, we built a comprehensive co-occurrence network that includes all relevant clinical and biological concepts mentioned in the corpus, linking them based on their co-occurrence in given abstracts.

The visual language employed to express a query over the network describes concepts as nodes and their co-presence within research abstracts as undirected edges; some concepts are associated with medical conditions, others with treatments, or biological entities. We also allow modifiers. Queries run on the network may correspond to the expressed graph pattern, or to a selected subpart.

The query semantics corresponds to extracting scientific evidence (i.e., publications) from the corpus, in support of the existence of the relationships linking the expressed concepts; each search process extracts the references that best explain the relationships occurring within the query. When a specified path is not present in the co-occurrence network, alternative scored and ranked shortest paths connecting the nodes expressed in the query are proposed to the user (see ‘Methods’ section). The search output provides a ranking of references because of their weight, summing up the support that they give to several relationships in the query.

Our GRAPH-SEARCH implementation is supported by a graphical interface (see ‘Data and Code Availability’ section) that allows the user to express the queries and to interpret the results in terms of concepts explained by each discovered reference, thus enabling the users to better qualify the query during the interaction; in addition, users can read the textual abstracts of the retrieved references. Such interactive exploration of the search space allows for exploring assumptions and for progressively adapting them as a result of existing evidence.

The manuscript is organized as follows: we first describe the CORD-19 dataset; the characteristics of the co-occurrence network representing CORD-19 abstracts; the technological process of building the network; the graph search operation; and the Web user interface that allows us to express graph queries and explore the retrieved results. We then present a series of example use case queries relevant to COVID-19 research and review the current state of the art. We evaluate the benefits of using our GRAPH-SEARCH as opposed to full-text indexed databases and keyword-search; finally, we draw our conclusions.

Methods

The CORD-19 dataset

The COVID-19 Open Research Dataset (CORD-19, [12]) is a corpus of academic publications about COVID-19 and related coronavirus research; it was released and maintained by the Allen Institute for AI, in collaboration with The White House Office of Science and Technology Policy and other partners. Published articles and preprints were collected from several archives, including PubMed, PubMedCentral, bioRxiv, and arXiv; since its release, it has served as the basis of many COVID-19 text mining and discovery systems [12]. The final release of June 2, 2022, indexes more than 1 million publications. As summarized in Figure 1, nearly 79% of the documents in CORD-19 have an abstract. Out of them, around 41% have a full-text JSON file available, while less than 11% of available full-text publications have no abstract in the metadata table. Thus, we decided to focus on dataset records with an abstract.

The file containing the metadata of the dataset's publications is a comma-separated table (`CORD-19metadata.csv`), including:

- a unique identifier `cord_uid` for a cluster of different records of the same publication – upon it, we performed deduplication and subsequent reconciliation of the other metadata of the cluster into a single record;
- `title` of the publication – we detected the language and filtered out those not in English;
- `abstract` of the publication – only records with an actual abstract were retained;
- `publish_time` – the distribution of publication times, shown in Figure 2, shows that COVID-19 publications increased in the first half of 2020. Spikes at the beginning of each year correspond to publications whose publish time is incomplete (only the year field was filled). Publications prior to 2020 concern MERS, SARS, and coronavirus; we removed these publications.
- `journal`'s abbreviated name – fuzzy matching of the abbreviated names was performed with a list of full names obtained by Scopus [13];
- `authors` and `doi` of the publication;
- number of citations received (`numCitedBy`), obtained by SemanticScholar APIs [14].

Records from CORD-19 are already harmonized (see Wang et al. [12]), resulting in distinct `cord_uid` keys. However, several records of the same publication are included, with different metadata. We deduplicated them and retained just one record (the one published in a peer-reviewed journal, if available, else the richest one in metadata).

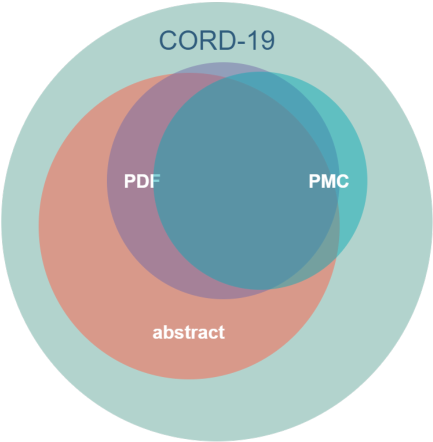


Fig. 1. Euler-Venn Diagram of the overlap of publications with abstract with publications with full-text JSON from PDF or from PubMedCentral in CORD-19.

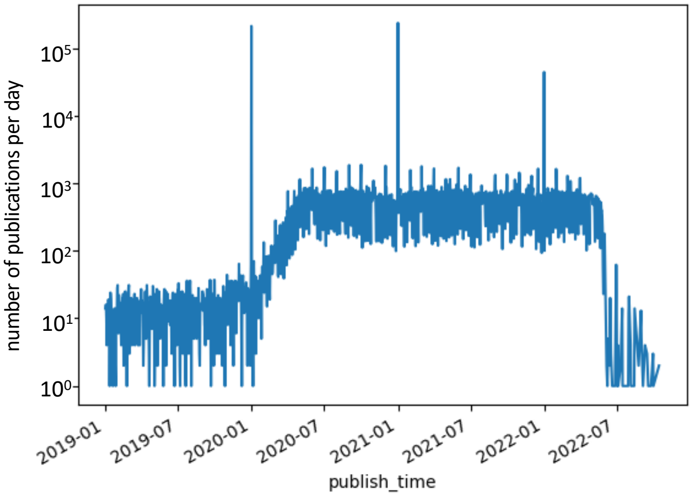


Fig. 2. Line plot showing the 10-base logarithm of the number of publications (y-axis) per publish time date (x-axis).

Co-occurrence network

The co-occurrence network was built to support graph search; it consists of entities and relationships mined from the `title` and `abstract` fields of the metadata table. For building it, we considered two sources: UMLS and CIDO. UMLS (the Unified Medical Language System [7]) is a generic source that includes many vocabularies and covers the entire spectrum of medicine; CIDO (the Ontology of Coronavirus Infectious Disease [8]) is a community-driven open-source biomedical ontology in the area of coronavirus infectious disease.

While CIDO has a simple concept structure, UMLS concepts have a taxonomy that includes macro-categories at a coarse level; each macro-category is further characterized by a type. Currently, we consider the following UMLS macro-categories: `ACTIVITIES_AND_BEHAVIORS`, `ANATOMY`, `CHEMICALS_AND_DRUGS`, CONCEPTS_AND_IDEAS, DEVICES, DISORDERS, ENTITY, GENES_AND_MOLECOLAR_SEQUENCES, GEOGRAPHIC_AREAS, LIVING_BEINGS, OBJECTS, OCCUPATIONS, ORGANIZATIONS, PHENOMENA, PHYSIOLOGY, and PROCEDURES.

Entities of the co-occurrence network include as attributes the Name, optionally an Umls_id when the entity is extracted from UMLS, and the Frequency associated with the entity (i.e., number of documents in CORD-19 capturing that concept). Relationships in the co-occurrence network express the co-occurrence of two entities in one or more documents of CORD-19. Each relationship has the following attributes: a Name (built as concatenation in alphabetic order of the names of the entities that co-occur), a Frequency (the number of abstracts that mention such co-occurring entities), and then several statistical indicators of the relationship's strength within the corpus: the PMI value (Pointwise Mutual Information estimator, comparing the relative frequency of two concepts occurring together in the text to the probability of either concept occurring independently [15]); the NPMI value (Normalized Pointwise Mutual Information, normalized by Shannon's self-information, ranging from -1 to 1 [16]); and the Cramer’s V value (measuring the statistical significance of the co-occurrence between two entities [17]).

Figure 3 illustrates the process of ontology creation at a conceptual level. The process applies to textual abstracts – in Figure 3 we consider an excerpt of the textual abstract of [18] – and consists of an entity recognition task aiming to extract the known ontological terms (either from UMLS or from CIDO) followed by an entity linking task; eventually, we produce a co-occurrence network, whose entities are extracted terms and whose relationships connect entities that co-occur, weighted by the strength of the co-occurrence. We next detail the data extraction and transformation process.

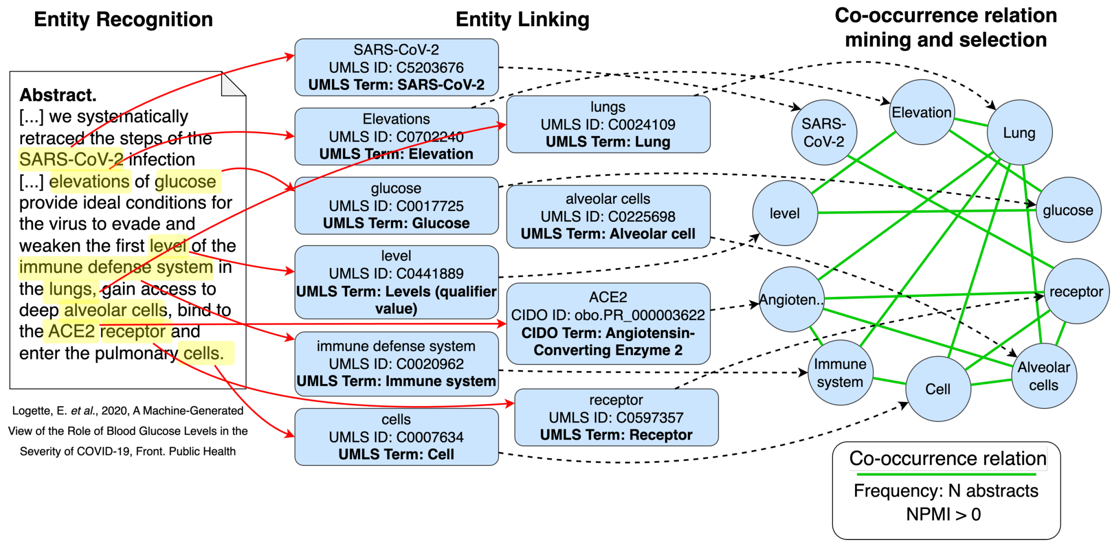

Entity Recognition
Entity Linking
Co-occurrence relation mining and selection
Abstract.
[...] we systematically retraced the steps of the SARS-CoV-2 infection [...] elevations of glucose provide ideal conditions for the virus to evade and weaken the first level of the immune defense system in the lungs, gain access to deep alveolar cells, bind to the ACE2 receptor and enter the pulmonary cells.
Logette, E. et al., 2020, A Machine-Generated View of the Role of Blood Glucose Levels in the Severity of COVID-19, Front. Public Health
SARS-CoV-2
UMLS ID: C5203676
UMLS Term: SARS-CoV-2
Elevations
UMLS ID: C0702240
UMLS Term: Elevation
lungs
UMLS ID: C0024109
UMLS Term: Lung
glucose
UMLS ID: C0017725
UMLS Term: Glucose
alveolar cells
UMLS ID: C0225698
UMLS Term: Alveolar cell
level
UMLS ID: C0441889
UMLS Term: Levels (qualifier value)
ACE2
CIDO ID: obo.PR_000003622
CIDO Term: Angiotensin-Converting Enzyme 2
immune defense system
UMLS ID: C0020962
UMLS Term: Immune system
receptor
UMLS ID: C0597357
UMLS Term: Receptor
cells
UMLS ID: C0007634
UMLS Term: Cell
SARS-CoV-2
Elevation
Lung
level
glucose
Angioten..
receptor
Immune system
Cell
Alveolar cells
Co-occurrence relation
Frequency: N abstracts
NPMI &gt; 0

Fig. 3. Rationale of co-occurrence network construction. Ontological terms are recognized in textual abstracts using entity recognition; then, this process is reiterated with ~660K publications' abstracts. Terms are connected to each other using entity linking; each relationship between entities is associated with several properties representing the co-occurrence weight, using different statistical methods. The generated connected co-occurrence network has ~128 thousand concepts and ~47 million relationships.

Data provisioning and co-occurrence network construction

The data provision workflow is represented in Figure 4; it follows the extract-load-transform paradigm. Data was extracted from CORD-19 and loaded into the data storage system. The pipeline produces three data objects: the co-occurrence network, the metadata table, and the *inverted index*, i.e., a simple postings-list whose keys are the relationships of the co-occurrence network and whose elements are links to the relevant publications where such relationships co-occur. Other data tables contain intermediate results of the extraction and curation of the entities – i.e., the nodes – of the co-occurrence network and of the computation of the co-occurrence measures employed for the relationships. For storing data tables, we selected the MariaDB relational engine [19]; for storing the co-occurrence network we selected the Neo4j graph data engine [20].

Data loading

Three tasks apply to raw CORD-19 data and produce a metadata table. Metadata was obtained by using the "GET metadata" from the S3 bucket of AllenAI; then, we performed a "Wrangling and Cleaning" step and the "Augment and Load" step on the cleaned metadata table with information from the external APIs.

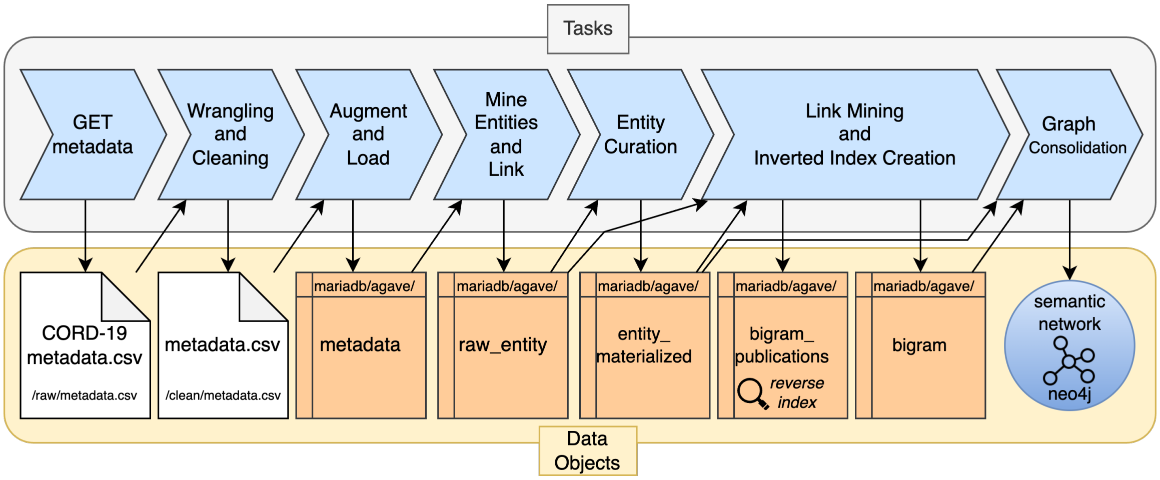


Fig. 4. Workflow diagram of the GRAPH-SEARCH data provision pipeline. Tasks are performed sequentially, each task uses data objects and produces data objects, starting from the raw `CORD-19metadata.csv` file present in CORD-19, which is translated into `metadata.csv` once cleaned. The final outcome of the pipeline is a Neo4j database containing the network.

Entities mining and linking

The "Mine Entities and Link" task takes as input the curated and augmented `metadata` table and produces the `raw_entity` table. With a single pass over the title and abstract, we performed typical Information Retrieval steps such as lexical analysis, removal of stopwords, stemming, and lemmatization. Then we performed Named Entity Recognition (NER) – consisting of the identification and extraction of entities from unstructured text and linking to UMLS and CIDO; specifically, we used the `en_core_sci_lg` model of the `scispaCy` Python package. The selected model is particularly suited for processing English-based scientific literature, providing a ~785k word vocabulary with 600k word vectors, with a declared F1-scores for mentions of 68.67 (see [21] for details on the achieved performances).

Entities are linked to UMLS and CIDO by associating each concept with the UMLS identifier (with its related type and macro-class) and/or the CIDO identifier.

Entity curation

The "Entity Curation" task aggregates the occurrences in the `raw_entity` table and outputs the `entity_materialized` table, collecting all the entities to be employed as nodes of the co-occurrence network. In this pass, we excluded the occurrences of the entities that score a low similarity with UMLS/CIDO concepts; we used a normalized string similarity measure based on the Levenshtein distance and a threshold value of 0.7. We also included within entities some *utility* terms that indicate level modifiers (such as 'high' and 'increased') or causative connectors (i.e., `induces'). Eventually, we added the entity type and macro-category, using their names in UMLS.

Link mining

The "Link Mining and Inverted Index Creation" task uses the `raw_entity` table and the `entity_materialized` table to generate the `bigram` table (i.e., information on the links of the co-occurrence network) and the `bigram_publications` table that we use as an *inverted index* in the information retrieval process.

A co-occurrence is a relationship between two concepts, and it exists when those two concepts occur in the same document. Each relationship is named using the convention "X.name - Y.name", where X and Y are the two concepts expressed as nodes, which it connects, and X.name precedes Y.name alphabetically.

We designed a greedy algorithm – optimized for big data contexts – to extract the relationships in a single pass over the publications. This algorithm requires two *read-only* lookup tables, built before the execution: `publication_entities` (for each publication a list of mentioned entities) and `entity_publications` (for each entity, a list of mentioning publications). The complexity of the algorithm is $o(N^2)$, where N is the number of entities in the `entity_materialized` list; in practice, the number of required comparisons is low, as the number of entities in each publication is much lower than the total number of entities selected in the "Entity Curation" step.

Graph consolidation

The "Graph Consolidation" task selects data from the `entity_materialized` and `bigram` tables and migrates it to the Neo4j instance to create the co-occurrence network.

The nodes are curated in the previous "Entity Curation" step. The relationships of co-occurrence are chosen at this stage, based on their NPMI, which is the point estimate of the Mutual Information, normalized by the Shannon Self-Information between [-1,+1]; this compares the probability that the two entities occur together. We exclude the relationships with NMPI$\leq$ 0, as a non-positive NPMI indicates that the relationship is not significant.

The resulting co-occurrence network has 128,249 entities and 47,198,965 relationships, extracted from 662,105 initial publications. Using the Neo4j Graph Data Science library [22], we verified that the graph is a unique connected component—such a condition is essential to ensure that every possible formulated graph query can be matched on the co-occurrence network.

Graph query search

A graph query Q is a connected graph formed by nodes and undirected relationships, where nodes are the set of entities appearing in Q and $rels(Q)$ is a set of arbitrary relationships connecting some pairs of entities in Q. A subgraph Q' is simply a connected subset of the nodes and relationships of Q. The search strategy is composed of two steps: matching of graph query against the co-occurrence network and extraction of the relevant publications.
Graph query matching is the operation of comparing the graph query Q with the co-occurrence network N created along the procedure described in the 'Data provisioning and co-occurrence network construction' section. By construction, each entity in Q is contained in N, whereas relationships in $rels(Q)$, arbitrarily created in Q, may not be present in N. Both Q and N are connected graphs with undirected relationships; then, matching Q within N can be seen as an instance of inexact graph matching [23].
Figure 5 guides the intuition of the matching operation. A graph query A-B (in blue) is searched over a co-occurrence network (in white). No direct relationship exists between A and B on the network. However, several alternative finite paths exist (i.e., A-X-B, A-Y-Z-B, or A-V-Y-Z-B). Among these, A-X-B is found to be the 'shortest path' between A and B, as its length (or distance, in green) equals 1.

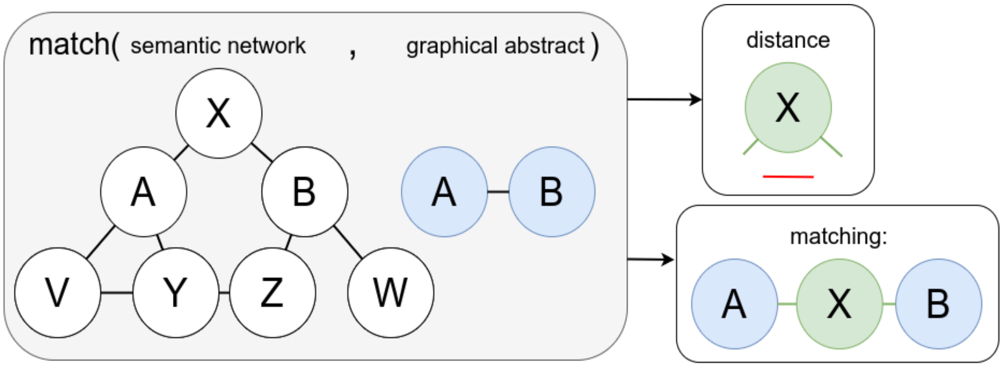


Fig. 5. Graph query matching operation.

That is, all entities in Q are matched in N; then, for each relationship r in $rels(Q)$, connecting nodes α and β, we retrieve the 'shortest paths' within N that connect α and β, i.e., a chain of relationships $r'_1, r'_2, \dots, r'_n$, where r' is in $rels(N)$, r'_1 starts from node α, and r'_n ends in node β. Shortest paths are computed by using the All Pairs Shortest Path function `allShortestPaths` available in Cypher, Neo4j v4.4 [20]. Candidate shortest paths are ranked by the average of the NPMI property associated with each relationship along the path; we retain the top ten paths in the ranking.
We refer to the set of candidate shortest paths as *expansion*; the selection of exactly one preferred path among the candidates of the expansion is performed interactively by the user of the search system, as it is strictly domain or context-specific.
Relevant publications extraction corresponds to the retrieval of the publications that mention concepts of the matched graph, using the inverted index. We access the inverted index by relationship name, using either r when it appears in the relationships $rels(N)$ of the co-occurrence network, or all the relationships $r'_1, r'_2, \dots, r'_n$ appearing in the specified $path(r)$. The score of a publication P relative to a query Q (i.e., the number of explained relationships) is computed as follows:

$$Score(P,Q) = \sum_{r \in rels(Q)} \frac{\sum_{r' \in path(r)} P_{r'}}{|path(r)|} \qquad (1)$$

The addends of the external summation represent a score assigned to each relationship r in Q. Each addend captures how well P represents r; it is equal to 1 if P directly mentions the relationship of Q (e.g., when $path(r) = r'$, with length 1) or if P mentions all the relationships of $path(r)$. Otherwise, it equals a fraction of one, counting the number of relationships $r'_1, r'_2, \dots, r'_n$ of $path(r)$ mentioned in P, divided by the length of $path(r)$.

Extracted publications are ordered by their score; they are further described by other properties, such as the sum of the NPMI of all the mentioned relationships and the date of publication.

Running example

Consider Figure 6 as an example of the four steps performed during the search:

- Panel A: *Create graph query*. Nodes are chosen among the concepts existing in the co-occurrence network; node names can be found through a dedicated browser working either by auto-completion of user-typed content (matching terminologies concepts) or by selection of Category/Type and contained concepts; search on multiple terminologies at the same time is allowed. For each concept, we provide a description and ID from the original source. Relationships can be drawn between any pair of nodes.
- Panel B: *Find paths.* For each pair of entities connected by a relationship in the graph query, the Neo4j graph is queried to find the shortest paths (at most ten) with top average NPMI scores.
- Panel C: *Select paths.* The user selects the most relevant path for each original relationship that has been expanded.
- Panel D: *Retrieve publications and return ranking to the user.* The system collects the names of all the relationships from the expanded graph query (computed in step B and selected in step C) and exploits them to retrieve the posting lists of publications (from the inverted index). It computes the relationships explained by each publication. Then, it ranks the publications by 1) the number of explained relationships of the original graph query (see Eq. 1); 2) the sum of NPMI scores of the relationships; and 3) the publication date. Finally, it shows the complete list with publications' metadata.

In Figure 6D, we observe that five expansions are produced: the first publication scores 1 in four expansions and 1/2 in the expansion at the top right-end of the graph query. Indeed, publication 1 only includes the relationship (AngII)-(1,0), which is half of the selected shortest path that connects (AngII) and (Vascular Permeability).

The second publication scores 0 in one expansion, as there is no path between (AngII) and (Vascular Permeability); 1 in three expansions; and 2/3 in the expansion at the left end of the graph query – the relationship (SARS-CoV-2)-(1,0) is not mentioned.

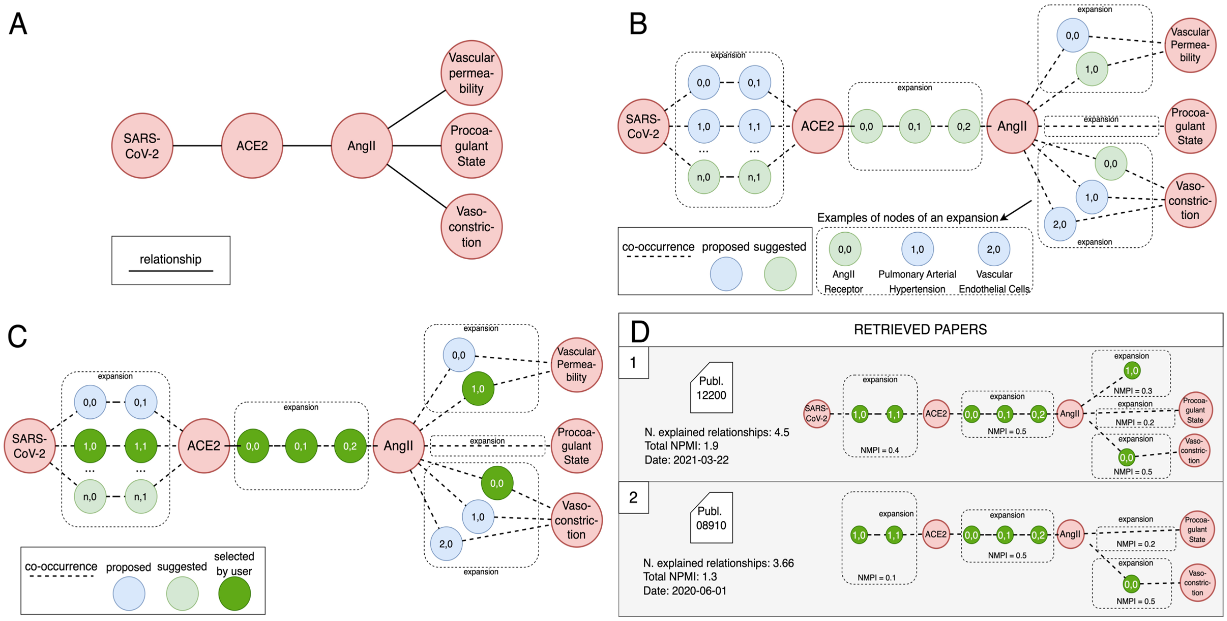


Fig. 6. (A) Example of a graph query with 6 concepts and 5 relationships. (B) Match of graph query on the co-occurrence network, with the search of shortest paths (in the dashed spaces, called *expansions*). Considering the relationship between (SARS-CoV-2) and (ACE2), its expansion includes three paths of length 3, each characterized by two intermediate nodes. Light green paths have the highest average NPMI of each expansion. (C) Regardless of the suggested paths with the highest average NPMI, users can select any path (dark green). (D) A list of publications, ranked by their score, is extracted; the score is computed using Eq. 1 and considers all the relationships in the selected paths that are mentioned in the publication.

Results

Web Interface

With GRAPH-SEARCH, the researcher can express a query in the form of a graph query on a Web interface and retrieve a list of CORD-19 publications that best correspond to the query. During the search process, each link in the original graph query is expanded and matched with the co-occurrence network. When a relationship in the query is not available in the co-occurrence network, an expansion may suggest that several sets of concepts can explain a relationship in the original graph query; therefore, ten ranked paths are proposed to the user, who may express a preference according to her interest. After selecting one path for each expanded relationship, GRAPH-SEARCH provides a list of publications ranked by the number of explained relationships of the original graph query.

The GRAPH-SEARCH application service exposes a web user interface to query the co-occurrence network and exploit the graph-driven search methodology described in the 'Graph Query Search' section; it contains a backend (web server that exposes a RESTful API for high-level retrieval operations) and a frontend (visual interface that exploits the RESTful APIs to use the backend).

The web interface has been designed and implemented following the major steps of the algorithm described in the 'Running Example' above. The user experience has been modeled

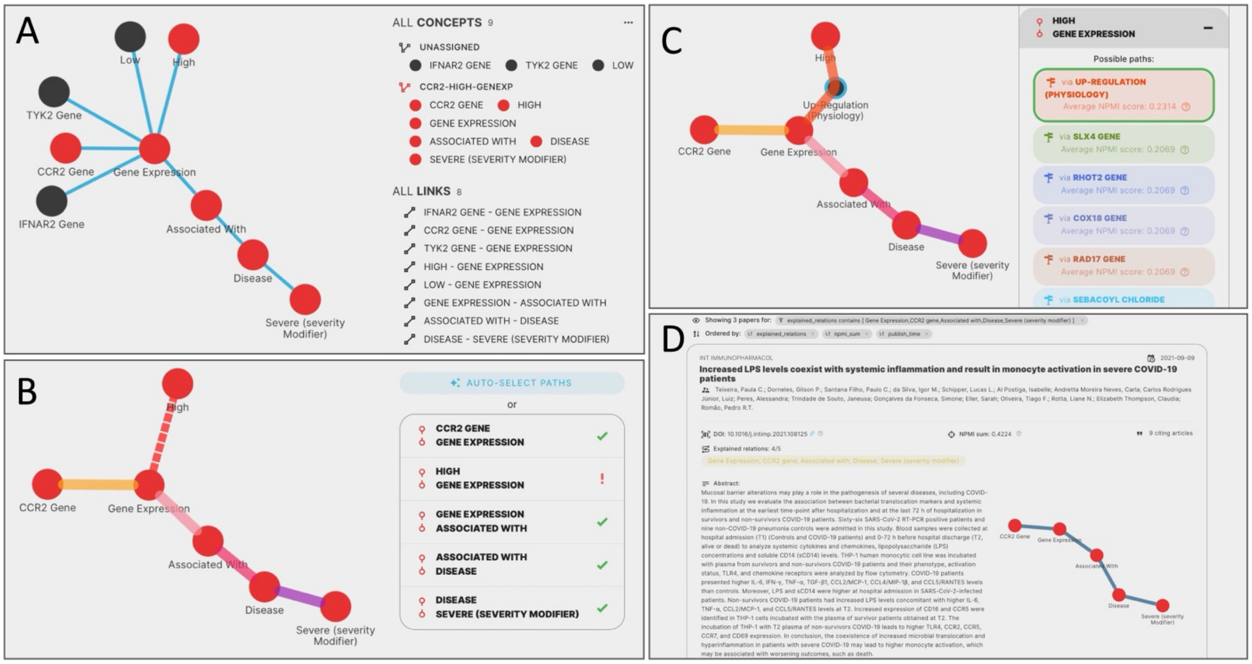
as a multi-page application; for each step of the retrieval strategy, different API services and a different page were implemented.
The frontend is built with the Vue.js framework and the D3.js library for graph illustrations; instead, the backend is written in Python and includes two components:

- `swagger_server`, which implements the web service logic, interfaces, and the models necessary to handle the persistence and asynchronicity behaviors of a multi-user system. We employed the `connexion` framework, a Flask-based web framework, and `SQLAlchemy` as the database abstraction layer;
- `core`, which implements the retrieval strategy and provides high-level programming interfaces for it. This package has been designed as an independent library that can be embedded in other applications, as it has been done with the backend service. Its implementation relies on several Python libraries, such as `neo4j`, `networkx`, and `SQLAlchemy`.

Use cases

Use case UC1 emphasizes the strength of exploratory search over graphs, by supporting users in selecting graph portions, considering/accepting proposed expansions, and browsing results in terms of NMPI and explained relationships. Use cases of increasing complexity are provided next, offering examples of searches upon graph queries with different shapes: UC2 and UC3 introduce very simple linear graph queries (one chain of nodes); UC4 shows the use of a Y-shaped graph query; and UC5 and UC6 represent more complex shapes with nodes forming triangles.

UC1. Genetic mechanisms of critical illness in COVID-19: Pairo-Castineira *et al.* [24] aim to reveal previously undescribed molecular mechanisms of critical illness in patients with COVID-19 with genome-wide studies. The results of such studies may provide therapeutic targets to modulate the host immune response to promote survival. Inspired by this publication, we create a graph query including relevant human genes that are related to higher or lower severity of COVID-19 (IFNAR2, CCR2, and TYK2 genes) and we link them to the change in the severity of the disease (see Figure 7A). Since the research idea is broad, we start the exploratory process focusing on a subgraph of the graph query (see nodes in red selected in Figure 7A). Here, we only consider the effect of the increase of expression in the CCR2 gene. Figure 7B shows how GRAPH-SEARCH expands the path between the concepts '`High`' and '`Gene Expression`' (not otherwise connected in the co-occurrence network). According to NPMI values, the most relevant concept connecting them is '`Up-Regulation (Physiology)`'. Figure 7C shows that the path going through this concept has been selected by the user among the other proposed. The Results page (Figure 7D) shows a publication (Teixeira *et al.* [25]) that covers 4/5 explained relationships of the original graph query. This means that – out of the five original relationships of the selected portion of the graph query – only four are explained by the publication (all except for the one between '`Gene Expression`' and '`High`'). At this point, the user can consider other portions of the graph query, or the entire query.



ALL CONCEPTS 9

UNASSIGNED

● IFNAR2 GENE ● TYK2 GENE ● LOW

CCR2-HIGH-GENEXP

● CCR2 GENE ● HIGH

● GENE EXPRESSION

● ASSOCIATED WITH ● DISEASE

● SEVERE (SEVERITY MODIFIER)

ALL LINKS 8

IFNAR2 GENE - GENE EXPRESSION

CCR2 GENE - GENE EXPRESSION

TYK2 GENE - GENE EXPRESSION

HIGH - GENE EXPRESSION

LOW - GENE EXPRESSION

GENE EXPRESSION - ASSOCIATED WITH

ASSOCIATED WITH - DISEASE

DISEASE - SEVERE (SEVERITY MODIFIER)

AUTO-SELECT PATHS

or

CCR2 GENE
GENE EXPRESSION ✓

HIGH
GENE EXPRESSION !

GENE EXPRESSION
ASSOCIATED WITH ✓

ASSOCIATED WITH
DISEASE ✓

DISEASE
SEVERE (SEVERITY MODIFIER) ✓

HIGH
GENE EXPRESSION

Possible paths:

via UP-REGULATION (PHYSIOLOGY)
Average NPMI score: 0.2314

via SLX4 GENE
Average NPMI score: 0.2069

via RHOT2 GENE
Average NPMI score: 0.2069

via COX18 GENE
Average NPMI score: 0.2069

via RAD17 GENE
Average NPMI score: 0.2069

via SEBACOYL CHLORIDE

Showing 3 papers for: explained_relations contains [Gene Expression,CCR2 gene,Associated with,Disease,Severe (severity modifier)]

Ordered by: explained_relations npmi_sum publish_time

INT IMMUNOPHARMACOL 2021-09-09

Increased LPS levels coexist with systemic inflammation and result in monocyte activation in severe COVID-19 patients

Teixeira, Paula C.; Dorneles, Gilson P.; Santana Filho, Paulo C.; da Silva, Igor M.; Schipper, Lucas L.; Al Postiga, Isabelle; Andretta Moreira Neves, Carla; Carlos Rodrigues Júnior, Luiz; Peres, Alessandra; Trindade de Souto, Janeusa; Gonçalves da Fonseca, Simone; Eller, Sarah; Oliveira, Tiago F.; Rotta, Liane N.; Elizabeth Thompson, Claudia; Romão, Pedro R.T.

DOI: 10.1016/j.intimp.2021.108125 NPMI sum: 0.4224 9 citing articles

Explained relations: 4/5

Gene Expression; CCR2 gene; Associated with; Disease; Severe (severity modifier)

Abstract:
Mucosal barrier alterations may play a role in the pathogenesis of several diseases, including COVID-19. In this study we evaluate the association between bacterial translocation markers and systemic inflammation at the earliest time-point after hospitalization and at the last 72 h of hospitalization in survivors and non-survivors COVID-19 patients. Sixty-six SARS-CoV-2 RT-PCR positive patients and nine non-COVID-19 pneumonia controls were admitted in this study. Blood samples were collected at hospital admission (T1) (Controls and COVID-19 patients) and 0-72 h before hospital discharge (T2, alive or dead) to analyze systemic cytokines and chemokines, lipopolysaccharide (LPS) concentrations and soluble CD14 (sCD14) levels. THP-1 human monocytic cell line was incubated with plasma from survivors and non-survivors COVID-19 patients and their phenotype, activation status, TLR4, and chemokine receptors were analyzed by flow cytometry. COVID-19 patients presented higher IL-6, IFN-γ, TNF-α, TGF-β1, CCL2/MCP-1, CCL4/MIP-1β, and CCL5/RANTES levels than controls. Moreover, LPS and sCD14 were higher at hospital admission in SARS-CoV-2-infected patients. Non-survivors COVID-19 patients had increased LPS levels concomitant with higher IL-6, TNF-α, CCL2/MCP-1, and CCL5/RANTES levels at T2. Increased expression of CD16 and CCR5 were identified in THP-1 cells incubated with the plasma of survivor patients obtained at T2. The incubation of THP-1 with T2 plasma of non-survivors COVID-19 leads to higher TLR4, CCR2, CCR5, CCR7, and CD69 expression. In conclusion, the coexistence of increased microbial translocation and hyperinflammation in patients with severe COVID-19 may lead to higher monocyte activation, which may be associated with worsening outcomes, such as death.

Fig. 7. GRAPH-SEARCH screens dedicated to UC1. (A) Graph query; (B) Find paths; (C) Select paths; (D) First publication on the results page.

UC2. COVID-19 and cystic fibrosis: Cystic fibrosis is a disorder that affects mostly the lungs, the digestive system, and other organs in the body. It is widely known that also COVID-19 affects the respiratory system. *How has their connection been investigated in CORD-19?* The simplest possible graph query in GRAPH-SEARCH holds two nodes (cystic fibrosis and COVID-19) connected by one relationship of co-occurrence. '`Cystic fibrosis`' is represented by UMLS concept ID C0010674; and '`COVID-19`' by the UMLS concept ID C5203670. The two concepts are not directly connected within the network; among the proposed paths in the expansion, we choose the one through the concept '`Respiratory secretion viscosity alteration`' (UMLS ID 3537094). Only one publication in CORD-19 explains this path, covering it completely, with an NPMI sum of 0.5668. Kratochvil *et al.* [26] characterized the composition of respiratory secretions of intubated COVID-19 patients finding that they closely resemble those of cystic fibrosis, a minor observation unrelated to clinical severity. In general, the lack of relevant clinical references confirmed our expectation that cystic fibrosis did not impact COVID-19 severity.

UC3. COVID-19 and NSAIDs: During the second year of the pandemic interest arose in the possibility of intervening at the onset of mild-to-moderate COVID-19 symptoms in outpatients (instead of hospitalized patients); it was suggested that this could prevent the progression to a more severe illness and long-term complications. More specifically, Perico *et al.* [27] investigated the use of anti-inflammatory drugs, especially non-steroidal anti-inflammatory drugs (NSAIDs) as a therapeutic strategy. In our graph query, we include as main concepts '`COVID-19`' (C5203670) - '`Outpatients`' (C0029921) - '`Anti-Inflammatory Agents, Non Steroidal`' (C0003211) - '`Cyclooxygenase 2 Inhibitors`' (C1257954), the last being a specific class of NSAIDs. In this case, no expansion of the original graph query is performed, as all the relationships are present in the co-occurrence network. The Results page contains a list of 440 publications, whose abstracts discuss the concepts in the graph query from different perspectives and approaches. The top three results include work from Consolaro *et al.* [28] – a home-treatment algorithm based on anti-inflammatory drugs; Popovych *et al.* [29] – discussing the therapeutic efficacy of the BNO 1030 extract, which is a phytotherapeutic anti-inflammatory agent; and Sava *et al.* [30] – exposing the results of a ninety-day treatment of patients with severe COVID with a specific NSAID drug, tocilizumab.

UC4. Elevated blood glucose levels and COVID-19 severity: Elevated blood glucose levels are considered a risk factor for the severity of the disease. With GRAPH-SEARCH, we compose a Y-shaped graph query (see Figure 8), expressing that high levels of blood glucose or increasing blood glucose can induce a severe illness. This example makes sophisticated use of *utility* terms; these are provided in a specific list of the concepts' browser of GRAPH-SEARCH.

As a consequence, we obtain a list of 395 results, where the top-ranked publication explains 5/5 relationships: Logette *et al.* [1] reports on the relationship between blood glucose levels and the severity of COVID-19. All following publications, ranked in descending order by the number of explained relationships of the original graph query, explain at most 3/5 relations.

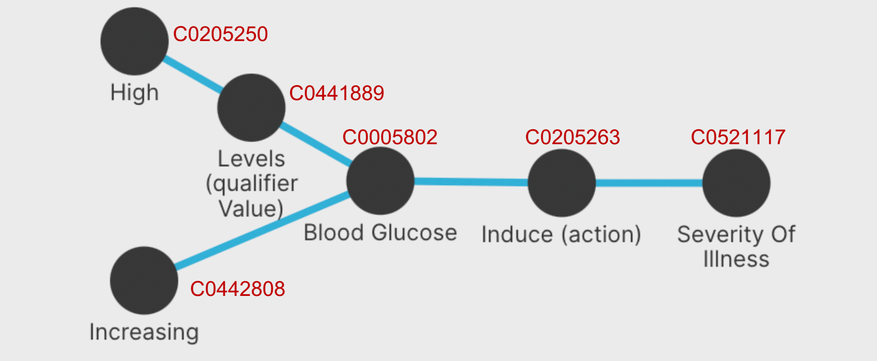


Fig. 8. Graph query of UC4, with UMLS concepts IDs in red.

UC5. COVID-19, ACE2, and cardiovascular diseases: Patel *et al.* in [31] hypothesized that SARS-CoV-2 infection could be associated with the shedding of ACE2. In their study, it is suggested that in patients with cardiovascular diseases, there is increased shedding of ACE2; consequently, higher levels of ACE2 in blood circulation are associated with the downregulation of membrane-bound ACE2. The graph query in Figure 9A expresses this query, by connecting COVID-19 patients with cardiovascular diseases; as they have more circulating ACE2, consequently there is a downregulation of membrane-bound ACE2. When running this query, two relationships are not found in the co-occurrence network; the first paths suggested by the system as possible explanations are not meaningful w.r.t. the context, thus we select alternative concepts, i.e., '`Subacute Endocarditis`' and '`Intensive Care Unit`' (see Figure 9B). Results can be ranked by number of citations; we found two publications particularly interesting, by Yamaguchi *et al.* [32] and Gupta *et al.* [33], as they propose solutions for the prevention and treatment of the side effects of COVID-19 for patients with cardiovascular diseases.

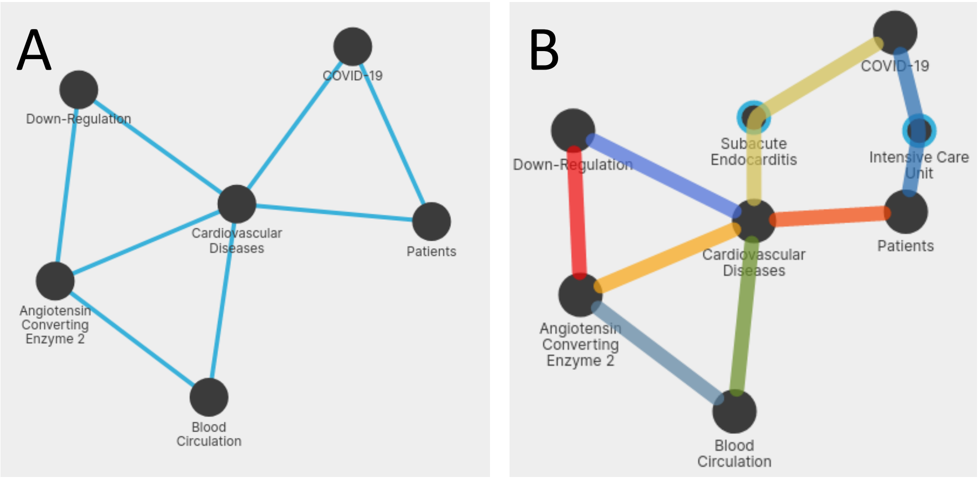


Fig. 9. Graph query of UC5 (A) and found paths (B).

UC6. COVID-19 Vaccines and Myocarditis: The side effects of vaccines are a topic of relevance. Here, we investigate the connection between events of heart inflammation (e.g., myocarditis) among adolescents and the COVID-19 Moderna vaccine. We compose a graph query in GRAPH-SEARCH with four nodes (see Figure 10A); a triangle is formed by '`Adolescent (age group)`' (C0205653), '`Myocarditis`' (C0027059), and the '`Moderna COVID-19 Vaccine`' (CIDO ID obo.VO _0005157); the vaccine entity is connected to the '`COVID-19`' (C5203670) node. COVID-19 and Moderna COVID-19 Vaccine are not directly connected; among the possible paths suggested by GRAPH-SEARCH, the two scoring the highest sum of mutual information are through '`Vaccination`' and '`Myopericarditis`'. The latter refers to both myocarditis and pericarditis (i.e., the inflammation of the pericardium which is the sac that surrounds the heart). The latter concept allows us to expand the initial query to complete the match with the co-occurrence network (see Figure 10B). On the Results page, 190 bibliographic resources are provided. The top-ranked one, which explains all four relationships of the graph query, is a report by Gargano *et al.* [34] that suggests the implication of the use of mRNA vaccines with a higher risk for myocarditis in males aged 12-29 years. The following results do not explain the relationship between the COVID-19 Moderna Vaccine and COVID-19 through Myopericarditis, as they explain only three relations. These results, for instance, report adverse events of Myocarditis after vaccination in the US [35] and Korea [36].

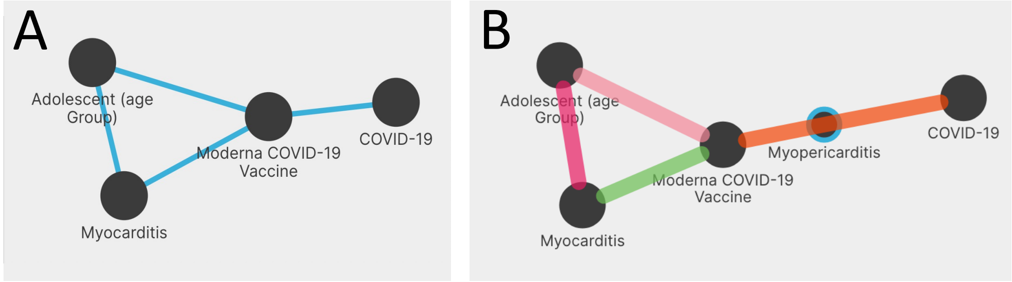


Fig. 10. Graph query of UC6 (A) and found paths (B).

Query performances

GRAPH-SEARCH queries are composed of two computationally intensive steps: 1) the graph query matching over the co-occurrence network and 2) the retrieval and ranking of publications related to the query. For each such step, we run a performance analysis.

Specifically, we simulated random queries with 2, 4, 6, 8, or 10 nodes from the existing co-occurrence network; we assume that these are the typical use case scenarios – as queries represent small queries of researchers created through the graphical interface.

We separately measure computation times of the first and second steps (respectively shown in Figure 11A and 11B); each experiment has been repeated on 10 queries, generated randomly using the 'Random walk with restarts sampling' method of Neo4j. We observe that the computational times for graph matching is in all cases below 2.2 seconds, and its growth is less-than-linear with the number of the nodes, whereas the retrieval operation typically takes up to 3 seconds, with a small number of outliers due to cache misses – the resulting user delay in these scenarios seems quite acceptable.

We also created random graph queries by removing increasing percentages of their relationships, to simulate the difference between exact and inexact graph search (thereby triggering the search for alternative shortest paths); Computational times (not shown for brevity) are not significantly affected.

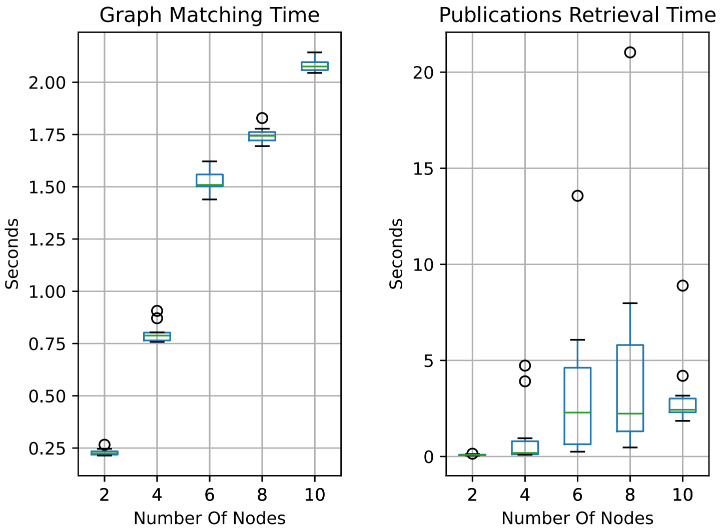


Fig. 11. Boxplots measuring the time for the graph matching operation (A) and the publication retrieval operation (B) performed using complete graph queries of – respectively – 2, 4, 6, 8, and 10 nodes (each repeated 10 times).

Related work

In this section, we review classic approaches to search over co-occurrence networks, then we focus on the specific use of bio-ontologies in information extraction systems, finally, we propose a close comparison with COVID-19-specific search systems.

Semantic-network search

The task of searching and extracting literature documents over co-occurrence networks with graph-based queries can be considered through the subproblems that compose it. To query a co-occurrence network with a graph-like query, a similarity measure between graphs must be defined. Existing methods in the context of graph databases include definitions of graph edit distances and maximum common subgraphs [23], but a later approach introduced a similarity measure based on a graph kernel between pairs of documents, which exploits the shortest paths between nodes as units to compare graphs [37]. Considering the construction of the co-occurrence networks from datasets of literature documents, different approaches are available to extract concepts to represent nodes in the network and connections between them. The survey from Han *et al.* [38] and the work by Shi *et al.* [39] present all the main methodologies and text-mining pipeline architectures, here applied to engineering and design (i.e., subsets of scientific literature). G-Bean [40] is also relevant related work, i.e., a graph-based tool that exploits ontologies for graph-based query expansion to support the user search intention discovery.

Literature annotation and bio-ontologies

The incorporation of bio-ontologies in information extraction and information retrieval has demonstrated its efficacy through diverse applications, such as patent information retrieval [41] and identification of concept domains [42]. Bio-ontologies are also applied in natural language processing tasks, like NER [43]. Moreover, [44] illustrates the application of bio-ontologies in retrieving biomedical datasets, while [45] emphasizes their role in literature search facilitation and metadata organization. The potential for refining search queries through ontology-guided expansion is also a recurring theme in the biomedical literature for information retrieval; [46] and [47] show query expansion methodologies using different medical vocabularies.

A fundamental aspect of research in this domain pertains to the availability and utilization of suitable corpora and datasets; works such as [48] and [49] have provided foundational annotated and curated resources that underpin the experimental frameworks addressing these tasks. Lately, the integration of bio-ontologies with Language Models has also gained traction within the context of bio-information extraction [50], [51].

COVID-19-specific literature discovery

With the outbreak of the COVID-19 pandemic, several open-access datasets have been collected, including the National Institute of Health's COVID-19 [52], the Human Coronaviruses Data Initiative [53], and COVIDScholar [54].

The CORD-19 dataset received the widest attention. Several knowledge graphs that exploit this dataset were proposed at the beginning of the pandemic for representing biomedical entities (e.g., CORD-NER [55] and COVID-19 KG [56]) or publications metadata (e.g., Covid-19-Literature [57]). More recently, CovidPubGraph [58] has provided a comprehensive and updated knowledge graph, which integrates information from multiple sources, making results available through a SPARQL endpoint. Lastly, CovidGraph [59] exposed a knowledge graph in the Neo4j browser; several external ontologies are used to tag entities. The focus of these resources is more on organization and semantic enrichment than on exploration.

The purpose of the TREC-COVID initiative [60] was that of setting up specific retrieval tasks in response to the pandemic, to be shared and addressed collaboratively by the community.

Instead, GRAPH-SEARCH aims to make literature about COVID-19 searchable and explorable. This objective is common to other two systems, LitCovid and Outbreak.info; these support enhanced keyword-based search, but they do not offer any graph-based search support.

LitCovid [1] was developed within the US National Institutes of Health (NIH) as a comprehensive resource of literature on COVID-19 (372,221 publications at the time of writing), updated regularly starting from PubMed. Publications are manually screened to determine if they are relevant to COVID-19, they are assigned to categories (such as overview, disease mechanism, transmission dynamics, treatment, case report, and epidemic forecasting), associated geographical location, and annotated with drug or chemical-related information found in their title/abstract – if applicable. The updated version [61] introduced the long-covid category, added annotations on variants and vaccines, and supported with machine learning algorithms the topic categorization (with a more updated model) and entity recognition (with NER). The interface allows us to apply filters on Country, Journal, Drug, Variant, and Vaccine and compose search strings combining AND, OR, NOT operators (not documented); results are ranked by Relevance, based on the widely used BM25 ranking function of Lucene. LitCovid positively compares its performances to the classical keyword search of PubMed (where annotations/tags are not used).

Outbreak.info Research Library [2] is a project of the Hughes, Su, Wu, and Andersen labs at Scripps Research. It offers a searchable interface of COVID-19 publications (complementing the content of LitCovid integrating preprint servers), together with clinical trials, datasets, protocols, and other resources. The data structure upon which the search is performed is supported by a schema; entities are connected by links with various semantics. The visual interface allows the use of some filters and keyword search; results are ranked by relevance based on Lucene's Practical Scoring Function on Elasticsearch (prioritizing the query normalization factor, coordination factor, term frequency, inverse document frequency).

Discussion

In this section we discuss how the proposed graph query search could be compared to other information extraction setups. For this purpose, we focus on two use case queries, i.e., the linear query presented in UC3 (four nodes in a linear pattern), and the red subgraph shown in UC1 (a non-linear six nodes query, expanded with an additional node in GRAPH-SEARCH).

Comparing with COVID-19-literature search systems

First, we considered running the use cases on the COVID-19 literature-dedicated search systems LitCovid and Outbreak.info. Both systems were queried by using concept names corresponding to UMLS terms in the nodes; unfortunately, they both suffer from the limitations of Boolean search. Specifically, if we search with conjunctive clauses and exact search (e.g., using `"Outpatients" AND "Anti-Inflammatory Agents, Non Steroidal" AND "Cyclooxygenase 2 Inhibitors" AND "COVID-19"` for UC3), no system returns any result. Dealing with exact search is hard (e.g., with LitCovid, the query `"Cyclooxygenase Inhibitors"` produces 3 results, whereas the query `"Cyclooxygenase 2 Inhibitors"` produces 5 results, although apparently more restrictive; instead, the query `Cyclooxygenase Inhibitors` (no quotes), without exact search, produces 12,287 results (including all references referring to generic inhibitors). In Table 1, we report the results of LitCovid with conjunctive queries but no exact matching, while a similar search is not supported by Outbreak.info. In comparison, GRAPH-SEARCH reports respectively 327 results for UC1 and 440 results for UC3. These outputs are hardly comparable, mainly because with LitCovid it is not possible to build a unique graph-shaped query; therefore, results of single conjunctive queries need to be evaluated one after the other, whereas GRAPH-SEARCH aggregates together the results of several conjunctive chains; it also expands given concepts with their acronyms (e.g., '`Anti-Inflammatory Agents, Non Steroidal`' is also searched as '`NSAIDs`'). Additionally, GRAPH-SEARCH allows for the expansion of specific links by adding new concepts (e.g., '`Up-Regulation (Physiology)`' in UC1). No domain-specific system for COVID supports graph-based search, allowing for a more insightful comparison.

Comparing with search on full-text indexed corpora

We also attempted a comparison with search operations performed on a baseline created by full-text indexing the CORD-19 titles and abstracts. Specifically, we employed the full-text indexing option of MariaDB, an open-source fork of MySQL [19]. Typically, full-text indexes work well for regular text; they build an index over specific words rather than the whole text—consequently, they show good performances for searches of specific words. The same queries used on LitCovid/Outbreak.info were used on this setup: on MariaDB, we employed the "Natural language mode" documented on [62] and thus removed the 'AND' Boolean operators and parentheses. To be part of the index, words must appear in less than 50% of the documents to be considered potentially relevant and to be used in searches (consequently, '`COVID-19`' and '`SARS-CoV-2`' are not considered relevant). Results are returned in descending order of relevance; limitations include the exclusion of partial (or very short/long) words.

Notwithstanding our attempts, we note that the comparison of the GRAPH-SEARCH approach with the full-text indexing setup is very difficult for many reasons:

a. the databases upon which search is performed are built on different assumptions (e.g., to be part of the index, words must appear in less than 50% of the documents; the co-occurrence network only includes entities that score high similarity with ontology concepts and exclude relationships with a negative NMPI);
b. in one case we perform separate keyword-search sessions with separate results (with associated precision/recall measures); in the other, we retrieve aggregated results (with summarized measures);
c. on one side, the ranking produced is only on single query result sets; on the other side, it is a global ranking.

Results are reported in Table 1; they must be read considering all these aspects. Note that results achieved with keyword-search are restricted to manipulating Boolean expressions, adding, and dropping keywords. Differently, the results on GRAPH-SEARCH (respectively 327 and 440) are inspectable, with ranking, ordering, filtering, and visualization options dedicated to the explained chains of entities; using our search paradigm, users can compose graph queries; more complex topologies also allow a stronger explainability of results.

Query		N. of retrieved publications		
		LitCovid	MariaDB	GRAPH-SEARCH
UC1	(Severe (severity modifier)) AND (Disease) AND (Associated With) AND (Gene Expression) AND (High) AND (CCR2 gene)	316	11	327
	(Severe (severity modifier)) AND (Disease) AND (Associated With) AND (Gene Expression) AND (High) AND (CCR2 gene) AND Up-Regulation (Physiology)	52	12	
UC3	(Outpatients) AND (Anti-Inflammatory Agents, Non Steroidal) AND (Cyclooxygenase 2 Inhibitors)	972	4	440
	(Outpatients) AND (Anti-Inflammatory Agents, Non Steroidal) AND (COVID-19)	1714	3	
	(Outpatients) AND (Cyclooxygenase 2 Inhibitors) AND (COVID-19)	3018	1	
	(Anti-Inflammatory Agents, Non Steroidal) AND (Cyclooxygenase 2 Inhibitors) AND (COVID-19)	37322	5	
	(Outpatients) AND (Anti-Inflammatory Agents, Non Steroidal) AND (Cyclooxygenase 2 Inhibitors) AND (COVID-19)	902	5	
	“Outpatients” AND “Anti-Inflammatory Agents, Non Steroidal” AND “Cyclooxygenase 2 Inhibitors” AND “COVID-19”	0	5	

Table 1. Results of evaluation of UC1 (Fig. 7) and UC3 queries when performed on the LitCovid search interface, on the full-text indexed MariaDB database, and on GRAPH-SEARCH.

Conclusion

GRAPH-SEARCH is the first search engine to propose the exploration of COVID-19 scientific literature using visual graph queries. GRAPH-SEARCH provides several unique features such as the possibility to describe concepts using well-known ontologies, to establish co-occurrence relationships between any two concepts of choice, to support search queries with concepts proposed and ranked by the system, and to browse resulting publications exploiting several visual and analytical measures.

The completeness and accuracy of the information captured in the co-occurrence network strictly depend on the advances of the NER methods employed during the steps of entity mining and linking. Other systems have employed expert curation (e.g., LitCovid) or community-driven curation (e.g., Outbreak.info). Although expert curation can improve the search experience, it does not properly scale; we opted for the exploitation of well-known biomedical ontologies such as UMLS/CIDO and to trust state-of-the-art NLP models used for Entity Recognition in our data provision pipeline.

The ability of our system to extract results was evaluated, attempting a comparison with existing published systems (LitCovid and Outbreak.info) and with full-text indexing search. We recognize that comparisons between the results retrieved from these systems are not ideal, as it is very critical to compare single search runs with a system where the result is built progressively on the graph – considering a set of aspects altogether (how the network was built and pruned, shortest path computation, completion with additional nodes, global ranking of results).

Co-occurrence networks are conventionally used for analyzing extensive text and big data. Common applications have involved sentiment analysis [63] and detection of prevailing topics [64]. Here, each node is a word occurring in a set of user-generated social media posts. Moreover, word-co-occurrence networks are present in clinical applications, e.g., [65] proposed to encode recordings of speech data used for recognizing Alzheimer's patients and controls. In all such cases, GRAPH-SEARCH may be employed to find specific subgraphs and propose completions of missing links.

In this work, we have demonstrated the capability of domain-specific (even inexact) graph query matching when semantics is considered only for nodes; we are aware of the limitations of this approach, which -at this stage- is considered a modeling choice. In future work, we plan to extend our search system to semantically rich knowledge graphs with both entities and relationships, thereby enriching the expressivity of graph queries (also including the possibility to capture relationships' semantics, with state-of-the-art methods [66] or as we already experimented in [67]). Then, we aim to formalize the use of graph queries in the context of graph databases, by studying the complexity of graph search and connecting it to classical theories of subgraph matching, shortest path search, and conjunctive query processing.

We also aim to conduct extensive empirical studies, to measure user satisfaction with systems such as GRAPH-SEARCH, analyzed along the three dimensions of usability, usefulness in deepening their knowledge of certain connected topics, and support of user's intentions in knowledge exploration.

Data and code availability

The data processing pipeline is available as a Docker image on https://hub.docker.com/r/frinve/graph-search. The GRAPH-SEARCH application is available on http://gmql.eu/graph-search and documented in the WIKI https://github.com/FrInve/graph-search/wiki/.

Acknowledgements

A.B. and S.C. conceived the work; F.I., A.B., and S.C. jointly conceptualized and designed the framework; F.I. implemented the pipelines of data ingestion, network construction, graph-query engine; F.I. and A.B. curated the user experience; A.B. drafted the manuscript; F.I. and S.C. improved it; all authors revised the final version of the manuscript; S.C. supervised the project.

The authors would like to thank Luca Minotti for implementing the front-end of the GRAPH-SEARCH Web application. This research is supported by the PNRR-PE-AI FAIR project funded by the NextGenerationEU program.

Conflicts of interest

None declared.

Abbreviations

CIDO	Coronavirus Infectious Disease Ontology
CORD-19	COVID-19 Open Research Dataset
NER	Named Entity Recognition
NPMI	Normalized Pointwise Mutual Information
NSAID	Non-Steroidal Anti-Inflammatory Drugs
UMLS	Unified Medical Language System

References

1. Chen Q, Allot A, Lu Z. LitCovid: an open database of COVID-19 literature. Nucleic Acids Research. 2021;49(D1):D1534-40. doi:10.1093/nar/gkaa952

2. Tsueng G, Mullen JL, Alkuzweny M, Cano M, Rush B, Haag E, et al. Outbreak.info Research Library: A standardized, searchable platform to discover and explore COVID-19 resources. Nature Methods. 2023;20(4):536-40. doi:10.1038/s41592-023-01770-w

3. Kejriwal M. Knowledge graphs and COVID-19: opportunities, challenges, and implementation. Harvard Data Science Review. 2020;(Special Issue 1). doi:10.1162/99608f92.e45650b8

4. Peng J, Xu D, Lee R, Xu S, Zhou Y, Wang K. Expediting knowledge acquisition by a web framework for Knowledge Graph Exploration and Visualization (KGEV): case studies on COVID-19 and Human Phenotype Ontology. BMC Medical Informatics and Decision Making. 2022;22(Suppl 2):147. doi:10.1186/s12911-022-01848-z

5. Ware C. Visual queries: The foundation of visual thinking. In: Knowledge and information visualization: Searching for synergies. Springer; 2005. p. 27-35. doi:10.1007/115101542

6. Cheerkoot-Jalim S, Khedo KK. Literature-based discovery approaches for evidence-based healthcare: a systematic review. Health and Technology. 2021;11(6):1205-1217. doi:10.1007%2Fs12553-021-00605-y

7. Bodenreider O. The Unified Medical Language System (UMLS): integrating biomedical terminology. Nucleic Acids Research. 2004;32(Suppl 1):D267-70. doi:10.1093/nar/gkh061

8. He Y, Yu H, Ong E, Wang Y, Liu Y, Huffman A, et al. CIDO, a community-based ontology for coronavirus disease knowledge and data integration, sharing, and analysis. Scientific Data. 2020;7:181. doi:10.1038/s41597-020-0523-6

9. Rindflesch, T.C., Kilicoglu, H., Fiszman, M., Rosemblat, G. and Shin, D., 2011. Semantic MEDLINE: An advanced information management application for biomedicine. *Information services & use*, *31*(1-2), pp.15-21. doi:10.3233/ISU-2011-0627

10. Bernasconi, A., Canakoglu, A., Masseroli, M. and Ceri, S., 2020. META-BASE: a novel architecture for large-scale genomic metadata integration. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 19(1), pp.543-557. doi:10.1109/tcbb.2020.2998954

11. Fernández, J.D., Lenzerini, M., Masseroli, M., Venco, F. and Ceri, S., 2015. Ontology-based search of genomic metadata. IEEE/ACM transactions on computational biology and bioinformatics, 13(2), pp.233-247. doi:10.1109/TCBB.2015.2495179

12. Wang LL, Lo K, Chandrasekhar Y, Reas R, Yang J, Eide D, et al. CORD-19: The COVID-19 Open Research Dataset. arXiv preprint arXiv:200410706. 2020. doi:10.48550/arXiv.2004.10706

13. Rose ME, Kitchin JR. pybliometrics: Scriptable bibliometrics using a Python interface to Scopus. SoftwareX. 2019;10:100263. doi:10.1016/j.softx.2019.100263

14. Silva D, Rohatgi S. semanticscholar; 2023. Last accessed online: Nov 15th, 2023. https://github.com/danielnsilva/semanticscholar

15. Church KW, Hanks P. Word Association Norms, Mutual Information, and Lexicography. Computational Linguistics. 1990;16(1):22-9. https://aclanthology.org/J90-1003

16. Bouma G. Normalized (pointwise) mutual information in collocation extraction. Proceedings of Conferences of the German Society for Computational Linguistics and Language Technology (GSCL). 2009;30:31-40.

17. Cramer H. Mathematical methods of statistics. vol. 43. Princeton university press; 1999.
18. Logette E, Lorin C, Favreau C, Oshurko E, Coggan JS, Casalegno F, et al. A machine-generated view of the role of blood glucose levels in the severity of COVID-19. Frontiers in Public Health. 2021:1068. doi:10.3389/fpubh.2021.695139
19. MariaDB Foundation. MariaDB; 2023. Last accessed online: Nov 15th, 2023. https://mariadb.org/
20. Neo4j. Neo4j Graph Database; 2023. Last accessed online: Nov 15th, 2023. https://neo4j.com/
21. Neumann M, King D, Beltagy I, Ammar W. ScispaCy: Fast and Robust Models for Biomedical Natural Language Processing. In: Proceedings of the 18th BioNLP Workshop and Shared Task. Florence, Italy: Association for Computational Linguistics; 2019. p. 319-27. doi:10.18653/v1/W19-5034
22. Neo4j. Neo4j Graph Data Science; 2023. Last accessed online: Nov 15th, 2023. https://github.com/neo4j/graph-data-science
23. Zhu Y, Qin L, Yu JX, Cheng H. Finding top-k similar graphs in graph databases. In: Proceedings of the 15th International Conference on Extending Database Technology; 2012. p. 456-67. doi:10.1145/2247596.2247650
24. Pairo-Castineira E, Clohisey S, Klaric L, Bretherick AD, Rawlik K, Pasko D, et al. Genetic mechanisms of critical illness in COVID-19. Nature. 2021;591(7848):92-8. doi:10.1038/s41586-020-03065-y
25. Teixeira PC, Dorneles GP, Santana Filho PC, da Silva IM, Schipper LL, Postiga IA, et al. Increased LPS levels coexist with systemic inflammation and result in monocyte activation in severe COVID-19 patients. International Immunopharmacology. 2021;100:108125. doi:10.1016/j.intimp. 2021.108125
26. Kratochvil MJ, Kaber G, Demirdjian S, Cai PC, Burgener EB, Nagy N, et al. Biochemical, biophysical, and immunological characterization of respiratory secretions in severe SARS-CoV-2 infections. JCI insight. 2022;7(12). doi:10.1172/jci.insight.152629
27. Perico N, Cortinovis M, Suter F, Remuzzi G. Home as the new frontier for the treatment of COVID-19: the case for anti-inflammatory agents. The Lancet Infectious Diseases. 2022;23(1):E22-33. doi:10.1016/S1473-3099(22)00433-9
28. Consolaro E, Suter F, Rubis N, Pedroni S, Moroni C, Pasto E, et al.` A home-treatment algorithm based on anti-inflammatory drugs to prevent hospitalization of patients with early COVID-19: A matched-cohort study (Cover 2). Frontiers in Medicine. 2022;9. doi:10.3389/fmed.2022.785785
29. Popovych V, Koshel I, Malofiichuk A, Pyletska L, Semeniuk A, Filippova O, et al. A randomized, open-label, multicenter, comparative study of therapeutic efficacy, safety and tolerability of BNO 1030 extract, containing marshmallow root, chamomile flowers, horsetail herb, walnut leaves, yarrow herb, oak bark, dandelion herb in the treatment of acute non-bacterial tonsillitis in children aged 6 to 18 years. American Journal of Otolaryngology. 2019;40(2):265-73. doi:10.1016/j.amjoto.2018.10.012
30. Sava M, Sommer G, Daikeler T, Woischnig AK, Martinez AE, Leuzinger K, et al. Ninety-day outcome of patients with severe COVID-19 treated with tocilizumab-a single centre cohort study. Swiss Medical Weekly. 2021;151(w20550). doi:10.4414/smw.2021.20550
31. Patel SK, Juno JA, Lee WS, Wragg KM, Hogarth PM, Kent SJ, et al. Plasma ACE2 activity is persistently elevated following SARS-CoV-2 infection: implications for COVID-19

pathogenesis and consequences. European Respiratory Journal. 2021;57(5). doi:10.1183%2F13993003.03730-2020
32. Yamaguchi T, Hoshizaki M, Minato T, Nirasawa S, Asaka MN, Niiyama M, et al. ACE2-like carboxypeptidase B38-CAP protects from SARS-CoV-2induced lung injury. Nature communications. 2021;12:6791. doi:10.1038/s41467-021-27097-8
33. Gupta S, Mitra A. Challenge of post-COVID era: management of cardiovascular complications in asymptomatic carriers of SARS-CoV-2. Heart Failure Reviews. 2021;27:239-49. doi:10.1007%2Fs10741-021-10076-y
34. Gargano JW, Wallace M, Hadler SC, Langley G, Su JR, Oster ME, et al. Use of mRNA COVID-19 vaccine after reports of myocarditis among vaccine recipients: update from the Advisory Committee on Immunization PracticesUnited States, June 2021. Morbidity and Mortality Weekly Report. 2021;70(27):977. doi:10.15585/mmwr.mm7027e2
35. Oster ME, Shay DK, Su JR, Gee J, Creech CB, Broder KR, et al. Myocarditis cases reported after mRNA-based COVID-19 vaccination in the US from December 2020 to August 2021. Jama. 2022;327(4):331-40. doi:10.1001/jama.2021.24110
36. Lee E, Kim K, Kim M, Yang HJ, YUM HY, Lee MH, et al. Adverse reactions to coronavirus disease 2019 vaccines in children and adolescents. Allergy, Asthma & Respiratory Disease. 2022:9-14.
37. Nikolentzos G, Meladianos P, Rousseau F, Stavrakas Y, Vazirgiannis M. Shortest-Path Graph Kernels for Document Similarity. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. Copenhagen, Denmark: Association for Computational Linguistics; 2017. p. 1890-900. doi:10.18653/v1/D17-1202
38. Han J, Sarica S, Shi F, Luo J. Semantic networks for engineering design: state of the art and future directions. Journal of Mechanical Design. 2022;144(2):020802. doi:10.1115/1.4052148
39. Shi F, Chen L, Han J, Childs P. A data-driven text mining and semantic network analysis for design information retrieval. Journal of Mechanical Design. 2017;139(11):111402. doi:10.1115/1.4037649
40. Wang JZ, Zhang Y, Dong L, Li L, Srimani PK, Yu PS. G-Bean: an ontology-graph based web tool for biomedical literature retrieval. BMC Bioinformatics. 2014;15(Suppl 12):S1. doi:10.1186/1471-2105-15-S12-S1
41. Taduri S, Law KH, Kesan JP, Sriram RD. Utilization of bio-ontologies for enhancing patent information retrieval. In: 2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC). vol. 2. IEEE; 2019. p. 91-6. doi:10.1109/COMPSAC.2019.10189
42. Dinh D, Tamine L. Identification of concept domains and its application in biomedical information retrieval. Information Systems and e-Business Management. 2015;13:647-72. doi:10.1007/s10257-014-0259-y
43. Gao S, Kotevska O, Sorokine A, Christian JB. A pre-training and self-training approach for biomedical named entity recognition. PloS One. 2021;16(2):e0246310. doi:10.1371/journal.pone.0246310
44. Wang X, Huang Z, van Harmelen F. Ontology-based semantic similarity approach for biomedical dataset retrieval. In: Health Information Science: 9th International Conference, HIS 2020, Amsterdam, The Netherlands, October 20–23, 2020, Proceedings 9. Springer; 2020. p. 49-60. doi:10. 1007/978-3-030-61951-05

45. Maraver P, Armananzas R, Gillette TA, Ascoli GA. PaperBot: open-source web-based search and metadata organization of scientific literature. BMC Bioinformatics. 2019;20:50. doi:10.1186/s12859-019-2613-z

46. Diaz-Galiano MC, Martin-Valdivia MT, Urena-Lopez L. Query expansion with a medical ontology to improve a multimodal information retrieval system. Computers in Biology and Medicine. 2009;39(4):396-403. doi:10.1016/j.compbiomed.2009.01.012

47. Dong L, Srimani PK, Wang JZ. Ontology graph based query expansion for biomedical information retrieval. In: 2011 IEEE International Conference on Bioinformatics and Biomedicine. IEEE; 2011. p. 488-93. doi:10.1109/BIBM.2011.15

48. Mohan S, Li D. MedMentions: A Large Biomedical Corpus Annotated with UMLS Concepts. In: Automated Knowledge Base Construction (AKBC); 2018. doi:10.24432/C5G59C

49. Basaldella M, Liu F, Shareghi E, Collier N. COMETA: A Corpus for Medical Entity Linking in the Social Media. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). Association for Computational Linguistics; 2020. http://dx.doi.org/10.18653/v1/2020.emnlp-main.253

50. Fei H, Ren Y, Zhang Y, Ji D, Liang X. Enriching Contextualized Language Model from Knowledge Graph for Biomedical Information Extraction. Briefings in Bioinformatics. 2021 May;22(3):bbaa110. doi:10.1093/bib/bbaa110

51. Kalyan KS, Rajasekharan A, Sangeetha S. AMMU: a survey of transformer-based biomedical pretrained language models. Journal of Biomedical Informatics. 2022;126:103982. doi:10.1016/j.jbi.2021.103982

52. NIH OPA iSearch COVID-19 Portfolio; 2023. Last accessed: Nov 15th, 2023. https://icite.od.nih.gov/covid19/search/

53. Human Coronaviruses Data Initiative; 2021. Last accessed: Nov 15th, 2023. https://about.lens.org/covid-19/

54. Dagdelen J, Trewartha A, Huo H, Fei Y, He T, Cruse K, et al. COVIDScholar: An automated COVID-19 research aggregation and analysis platform. Plos one. 2023;18(2):e0281147. doi:10.1371/journal.pone.0281147

55. Wang X, Song X, Li B, et al. Comprehensive Named Entity Recognition on CORD-19 with Distant or Weak Supervision. arXiv preprint arXiv:200312218. 2020. doi:10.48550/arXiv.2003.12218

56. Chen C, Ross KE, Gavali S, et al. COVID-19 Knowledge Graph from semantic integration of biomedical literature and databases. Bioinformatics. 2021;37(23):4597-8. doi:10.1093/bioinformatics/btab694

57. Steenwinckel B, Vandewiele G, Rausch I, Heyvaert P, Taelman R, Colpaert P, et al. Facilitating the analysis of COVID-19 literature through a knowledge graph. In: International Semantic Web Conference. Springer; 2020. p. 344-57. doi:10.1007/978-3-030-62466-8_22

58. Pestryakova S, et al. CovidPubGraph: A FAIR Knowledge Graph of COVID-19 Publications. Scientific Data. 2022;9:389. doi:10.1038/s41597-022-01298-2

59. Gutebier L, et al. CovidGraph: a graph to fight COVID-19. Bioinformatics. 2022;38(20):4843-5.¨ doi:10.1093/bioinformatics/btac592

60. Roberts, K., Alam, T., Bedrick, S., Demner-Fushman, D., Lo, K., Soboroff, I., Voorhees, E., Wang, L.L. and Hersh, W.R., 2021. Searching for scientific evidence in a pandemic: An overview of TREC-COVID. *Journal of Biomedical Informatics*, *121*, p.103865. doi:10.1016/j.jbi.2021.103865

61. Chen Q, Allot A, Leaman R, Wei CH, Aghaarabi E, Guerrerio JJ, et al. LitCovid in 2022: an information resource for the COVID-19 literature. Nucleic Acids Research. 2023;51(D1):D1512-8. doi:10.1093/nar/gkac1005
62. MariaDB. Full-Text Index Overview; 2023. Last accessed online: Nov 15th, 2023. https://mariadb.com/kb/en/full-text-index-overview/
63. Fudolig MI, Alshaabi T, Arnold MV, Danforth CM, Dodds PS. Sentiment and structure in word co-occurrence networks on Twitter. Applied Network Science. 2022 Dec;7(1):1-27. Doi: 10.1007/s41109-022-00446-2
64. Segev E. Textual network analysis: Detecting prevailing themes and biases in international news and social media. Sociology Compass. 2020 Apr;14(4):e12779. Doi: 10.1111/soc4.12779
65. Millington T, Luz S. Analysis and classification of word co-occurrence networks from Alzheimer's patients and controls. Frontiers in Computer Science. 2021 Apr 29;3:649508. Doi: 10.3389/fcomp.2021.649508
66. Singhal A, Simmons M, Lu Z. Text mining genotype-phenotype relationships from biomedical literature for database curation and precision medicine. PLoS computational biology. 2016 Nov 30;12(11):e1005017. doi:10.1371/journal.pcbi.1005017
67. Serna García G, Al Khalaf R, Invernici F, Ceri S, Bernasconi A. CoVEffect: interactive system for mining the effects of SARS-CoV-2 mutations and variants based on deep learning. GigaScience. 2023;12:giad036. doi:10.1093/gigascience/giad036