Abstract

We present a novel system that automatically extracts and generates informative and descriptive sentences from the biomedical corpus and facilitates the efficient search for relational knowledge. Unlike previous search engines or exploration systems that retrieve unconnected passages, our system organizes descriptive sentences as a relational graph, enabling researchers to explore closely related biomedical entities (e.g., diseases treated by a chemical) or indirectly connected entities (e.g., potential drugs for treating a disease). Our system also uses ChatGPT and a fine-tuned relation synthesis model to generate concise and reliable descriptive sentences from retrieved information, reducing the need for extensive human reading effort. With our system, researchers can easily obtain both high-level knowledge and detailed references and interactively steer to the information of interest. We spotlight the application of our system in COVID-19 research, illustrating its utility in areas such as drug repurposing and literature curation.†

1 Introduction

Efficiently extracting knowledge from the vast and ever-growing corpus of literature is crucial for researchers to keep up with the latest discoveries and trends in the field. The COVID-19 pandemic has highlighted this need, with thousands of related studies being published in a short period when a new disease emerges. However, surveying the latest findings requires significant effort, and researchers may struggle to see the big picture, leading to duplicated work and delaying the development of treatments (Wang et al., 2021). Hence, an exploration system that can effectively retrieve comprehensive information from the latest literature corpus is important.

†Demo video: https://www.youtube.com/watch?v=VvWs9JEP8ro System website: https://zhukerui.github.io/CovidDEER/. †Corresponding author.
To interactively retrieve up-to-date, comprehensive, and easily understandable relational sentences, and explore relationships between entities in one-hop or multi-hop connections. Additionally, we use ChatGPT and a fine-tuned relation synthesis model to generate succinct relation descriptions for entity pairs based on the retrieved sentences to aid users’ reading. It is worth mentioning that our system is automatically built without any supervised training or hand-crafted rules, making it seamlessly adaptable to any biomedical corpus with ease, and it can serve as a frontrunner for collecting knowledge in any future emergency.

2 Graph Construction

DEER (Huang et al., 2022b) is a form of knowledge representation that balances the openness and informativeness of free text and the structured representation of the knowledge graph. In this graph, nodes are entities, and edges are sentences describing the relationship between the two nodes, called relation descriptions, pointing from the subject node to the object node in the sentences. The previous DEER graph (Huang et al., 2022b) was built upon Wikipedia. Due to the limitations of the corpus, it does not contain much biomedical domain knowledge. In this section, we will introduce techniques for building a descriptive knowledge graph in the biomedical domain. Based on that, users could retrieve sentences with efficient graph queries and view the result from a connected perspective to gain a more holistic understanding of the retrieved information.

2.1 Corpus

To efficiently establish the system for retrieving knowledge about a specific topic, we build the DEER on a sub-domain corpus. In this work, we use COVID-19 Open Research Dataset (CORD-19) (Wang et al., 2020a) as a representative corpus in the biomedical domain, which comprises scientific papers related to COVID-19 and other coronaviruses, and note the DEER built from it as CovidDEER. For demonstration purposes, we used the snapshot on August 8th, 2020 to simulate a corpus when a new disease outbreak and some clinical experimental results have been published. With this corpus, we demonstrate how our system can retrieve valuable information for disease research and drug repurposing.

2.2 Pipeline

To construct CovidDEER, our system employs a pipeline that processes the corpus as follows:

**Entity Extraction and Linking** Initially, we extract biomedical entities from each sentence in the corpus and link them to biomedical ontologies using the NCBI Pubtator API and the SciSpacy library (Neumann et al., 2019). Specifically, we link the extracted entities to Cellosaurus, OMIM, MeSH, Gene, Taxonomy, and UMLS metathesaurus.

**Sentence Filtering** Next, we use SciSpacy to parse the sentences and remove those which do not have a subject entity or object entity as these entities serve as the head and tail entities in the relation description. Missing head or tail entities indicate that these sentences are not appropriate for describing relationships.

**Sentence Scoring** Then, we gather the parameters for a scoring function and use it to score the sentences. We use the relation description score (RDS) introduced in Huang et al. (2022b) as the scoring function. This scoring function extracts the

<table>
<thead>
<tr>
<th># documents</th>
<th># nodes</th>
<th># edges</th>
</tr>
</thead>
<tbody>
<tr>
<td>72,014</td>
<td>140,574</td>
<td>863,102</td>
</tr>
</tbody>
</table>

Table 1: Corpus and graph statistics for CovidDEER, collected using the RDS threshold of 0.7.
dependency path between the head entity, tail entity and other relation-related words in a sentence and generates a score between 0 and 1 to indicate how well this sentence expresses the relationship of the entities. Higher score indicates better the sentence as a relation description. A domain-specific RDS scoring function requires data of dependency path frequency from the domain corpus. Once the scoring function is setup with adequate corpus data, it can be frozen to evaluate any in-domain sentences. In this work, we collect dependency path frequencies from the whole CORD-19 corpus.

For each pair of subject-object entities in a sentence, we apply the scoring function and store the sentence and the RDS score with the corresponding entity pair.

**Score Filtering**  For each entity pair, we filter out the low-quality sentences with the RDS score and assign the rest to the edge from the head entity to the tail entity to construct the DEER. In practice, a sentence with RDS score over 0.7 usually has a good quality.

A visualization of the data pipeline is depicted in Figure 1 and the statistics of the CORD-19 corpus and CovidDEER are listed in Table 1. Note that the graph can be easily updated with the latest knowledge by extracting relation descriptions from recent papers.

## 3 Graph Query

To retrieve sentences from the CovidDEER, our system provides a Graph Query module with some auxiliary tools to allow an interactive and flexible search. Below are the queries supported by the graph query module.

**Entity-Entity Query**  Entity-Entity query allows users to retrieve relation descriptions between two entities. This is achieved by extracting sentences that lie on the edge connecting the two entities in the constructed CovidDEER graph. Unlike from systems that retrieve all or random sentences where the entities co-occur, our system focuses on returning sentences that capture the primary relationships between the target entities. This gives users more informative and clearer sentences and saves users from being distracted by meaningless sentences.

**Entity-Type Query**  To obtain a more comprehensive overview of relationships between an entity and an entity type, our system also supports Entity-Type query, which will retrieve sentences from edges between an entity and all its neighbors belonging to an entity type. For instance, users can set the entity to COVID-19 and the entity type to Chemicals. Then the system will return relation descriptions between COVID-19 and all kinds of related chemicals, which provides some insights into the Chemical-Disease interactions related to COVID-19. Our system supports all the entity types in the ontologies mentioned in Section 2.2.

**Multi-hop Query**  Besides finding direct neighbors of an entity, users can also query multi-hop neighbors to explore more indirect connections. By specifying the entities or entity types at each hop, users can retrieve sentences for multi-hop inference. For example, a user may begin with COVID-19, set Symptom as the first-hop entity type and Chemical as the second-hop entity type to explore drugs that can treat COVID-19 related symptoms, and thus, could be used for COVID-19 treatment. With this tool, our system could beat traditional knowledge graphs by providing the contextualized knowledge, and beat the text-based search engines by allowing multi-hop retrieval with one query.

**Modifier Filtering**  When querying a popular entity, CovidDEER may return too many edges, which may distract users from catching the general relationships. To alleviate this, we define the words in a sentence that convey the relation information as the modifiers and allow users to locate interesting edges using the modifiers. We extract the noun phrases, verbs, and adjectives on the dependency path between the two entities as the modifiers. For example, Table 2 shows the frequent modifiers collected between COVID-19 and its Chemical neighbors. These modifiers provide insights into the COVID-19-Chemical relationships and users can click the modifiers to highlight the edges where they occur. This tool could also help users perform a more fine-grained query to reduce unwanted results.

Figure 2 shows an example interface of our system, where the retrieved results are displayed as a

<table>
<thead>
<tr>
<th>Type</th>
<th>Frequent Modifiers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nouns</td>
<td>treatment (14), chloroquine (6), efficacy (4), hydroxychloroquine (4), therapy (2), option (2), patient (2)</td>
</tr>
<tr>
<td>Verbs</td>
<td>show (7), treat (5), use (5), propose (2), include (2)</td>
</tr>
<tr>
<td>Adjs</td>
<td>apparent (4), antiviral (3), effective (3), safe (2), severe (2), antimalarial (2)</td>
</tr>
</tbody>
</table>

Table 2: Frequent Modifiers between Chemicals and COVID-19.
Figure 2: The web interface of CovidDEER. The interface shows a graph retrieved by a two-hop query: “COVID19” – 10 “Disease or Syndrome” entities – 5 “Pharmacologic Substance” related entity types. The metformin is selected (in blue) and a directed path, COVID19 → Respiratory Distress Syndrome, Adult → metformin, is used for relation summary by ChatGPT and relation synthesis model.

graph, and users can checkout sentences by clicking the edges.

4 Relation Synthesis Model

Although CovidDEER displays the relational sentences in a graph view to reveal the connections between entities, considerable manual effort is still required to read and digest the information on the edges. Performing multi-hop logic inference is even harder as users need to find associated sentences across different edges. To reduce users’ reading effort, we trained a relation synthesis model (Huang et al., 2022b), which is based on a Fusion-in-Decoder model (Izacard and Grave, 2020) trained to take sequences of relation descriptions from the multi-hop paths between two entities in DEER and generate one single relation description for the entities. Each training data is collected by selecting the highest RDS-scored sentences on each edge in the multi-hop paths between a target entity pair as the input and the highest RDS-scored sentence on the one-hop path between the target pair as the output. In order to allow summarizing relation descriptions on each edge, we also add the lower-scored sentences on the one-hop path into the input. Since large language models have demonstrated strong capabilities through simple prompting (OpenAI, 2023; Anil et al., 2023; Qin et al., 2023; Bubeck et al., 2023; Huang and Chang, 2023), in addition to the fine-tuned model, we also prompt ChatGPT (OpenAI, 2022) to generate a short passage to summarize the relationship from the retrieved sentences. Detailed steps for fine-tuning and the prompt for ChatGPT can be found in Appendix B & C. By reading the generated relation descriptions first, users can get a general idea of the relation between the entities and then decide whether to read the retrieved sentences or not.

5 System Demonstration & Evaluation

In this section, we first evaluate the relation synthesis model by assessing the faithfulness of the generation with respect to the input relation descriptions. Then we demonstrate our system’s capacity in discovering unknown knowledge and locating information of interest with a drug repurposing task and a literature curation task respectively.

5.1 Relation Synthesis Model Evaluation

Huang et al. (2022b) have demonstrated the capability of the relation synthesis model to generate easily understandable relation descriptions. However, in the biomedical domain, it is crucial for the model to generate truthful sentences and not mislead the reader with erroneous information. Table 4 provides an example of the model’s generation, where the extracted relation descriptions for (COVID-19, Pneumonia) and (Pneumonia, Vaccines) are the inputs to the model. The 1-hop relation summary is the summarized relation description over the sen-
5.2 Case Study 1: Drug Repurposing

Drug repurposing intends to identify new uses for drugs that were originally used to treat other diseases. CovidDEER can aid researchers in identifying candidate drugs through the following steps:

- Set the target disease as the starting node.
- Search the first-hop neighborhood for diseases and symptoms related to the target disease.
- Search the second-hop neighborhood for drugs used to treat those related diseases and symptoms.

Suppose a researcher wants to discover the candidate drugs for COVID-19. By setting COVID-19 as the starting node and the Diseases and Symptoms entity type as the first-hop neighbors, the system retrieved a set of disease or symptom entities and the frequent modifiers. We select several verb modifiers that might indicate a “correlation” relationship between the entities and COVID-19. Then, we update the first-hop neighbors to 10 of these “correlated” entities and set 5 Pharmacologic substance related entity types as the second-hop neighbors. The retrieved two-hop graph can be seen in Figure 2. Similarly, we select several modifiers that might indicate a “treatment” relation to find candidate drugs. The selected modifiers are shown in Figures 3 and 4 and the collected candidate drugs are listed in Table 3.

<table>
<thead>
<tr>
<th>Candidate drugs</th>
</tr>
</thead>
<tbody>
<tr>
<td>nitric oxide, lamb preparation, beta-Lactams, Leukotriene B4, sphingosine 1-phosphate, amoxicillin, Macrolide Antibiotics, Macrolides, beta-Lactams, rifampin, Hydroxymethylglutaryl-CoA Reductase Inhibitors, methylprednisolone, trivalent influenza vaccine, Fibrates, lipid modifying drugs, plain, Corticosteroid ophthalmologic and otologic preparations, metformin, inhibitors, Corticosteroid ophthalmologicals, Bilirubin, Fibrates, nitazoxanide, atorvastatin, Artemisinins, antagonists</td>
</tr>
</tbody>
</table>

5.3 Case Study 2: Literature Curation

Literature curation (Wiegers et al., 2009) is the process of identifying documents relevant to a task or topic and locating and annotating the content of interest in these documents. The latter requires the curator to read through the whole document, which can be tedious and time-consuming. Our system provides an interface where users can run the pipeline in Figure 1 to build a DEER on any article indexed in PubMed and use this graph to locate the information relevant to the curation target.

Suppose a curation task is to collect Drug-Target interactions and a curator is assigned a relevant article. The curator first submits the article’s PubMed reference number (PMID) or PMC Identifier (PM-
6 Related Work

Exploration System Exploration systems aim to help users learn the content in the data sources through simple queries (Wang and Lo, 2021). Some systems are designed to retrieve sentence-level text pieces for a specific need. Wang et al. (2020b) retrieve textual evidence that semantically matches the queried statement. Lahav et al. (2022) build a set of scientific challenges and directions through entity co-occurrence. Taub-Tabib et al. (2020) develop a lightweight query language to retrieve sentences that syntactically match an example sentence. In contrast, our system collects relational sentences into a graph structure and displays the retrieved sentences in a graphic view that shows the connection between the text pieces, which is not seen in previous works.

Literature-Based Discovery Literature-based discovery (LBD) tools aim to discover unknown knowledge and generate new hypotheses by connecting current knowledge scattered in different literature together (Swanson, 2008), which is commonly used in biomedical tasks like drug repurposing and interaction prediction. Early LBD tools (Swanson, 1986; Smalheiser and Swanson, 1996) require manual effort in organizing information from the passages. Recent studies (Pu et al., 2023) approach LBD as a link prediction task over knowledge bases, where new knowledge is discovered as predicting new links between concepts. Our system is more like the early LBD tools. With the high-quality relational sentences and the advance of LLM in language understanding, our system greatly alleviates the user’s reading workload and allows a rough verification of the generated hypothesis based on the retrieved sentences.

7 Conclusion

In this work, we developed an exploration system in the biomedical domain that operates on a COVID-related corpus facilitating efficient retrieval of relational knowledge and enabling tasks such as drug repurposing and literature curation. We demonstrate the advantages of managing a raw text corpus in a descriptive knowledge graph, including streamlined management, support for multi-hop reasoning across sentences from various articles, and comprehensive visualization of entity connections in the domain. Additionally, we equipped users with a modifier filtering module and a relation synthesis model that offer an overview of the relations on the edge before reading. In future work, we aim to enhance the accuracy and reliability of the relation descriptions generated for user reference.
Limitations

Our system currently only support at most 2-hop query, since the number of entities in the graph will grow exponentially as the path gets longer, which will cause difficulty in reading the graph and reasoning along the path. This hinders our system from studies of more complex network like biochemical pathways, which involves several steps of reaction, and limits the possible knowledge that could be discovered by the system.

The relation synthesis model is trained to generate a single relation description, which is sometimes incapable to cover all the necessary information about the relationship of the target entities. ChatGPT could generate a short paragraph with more details included, but it is more costly than running a local fine-tuned model.

Acknowledgements

This material is based upon work supported by the National Science Foundation IIS 16-19302 and IIS 16-33755, Zhejiang University ZJU Research 083650, IBM-Illinois Center for Cognitive Computing Systems Research (C3SR) and IBM-Illinois Discovery Accelerator Institute (IIDAI), grants from eBay and Microsoft Azure, UIUC OVCR CCIL Planning Grant 434S34, UIUC CSBS Small Grant 434C8U, and UIUC New Frontiers Initiative. Any opinions, findings, conclusions, or recommendations expressed in this publication are those of the author(s) and do not necessarily reflect the views of the funding agencies.

References


A  Content of PMID 34767876

Molecular mechanisms of hepatotoxic cholestasis by clavulanic acid: Role of NRF2 and FXR pathways.

Treatment of beta-lactamase positive bacterial infections with a combination of amoxicillin (AMOX) and clavulanic acid (CLA V) causes idiosyncratic drug-induced liver injury (iDILI) in a relevant number of patients, often with features of intrahepatic cholestasis. This study aims to determine serum bile acid (BA) levels in amoxicillin/clavulanate (A+C)-iDILI patients and to investigate the mechanism of cholestasis by A+C in human in vitro hepatic models. In six A+C-iDILI patients, significant elevations of serum primary conjugated BA definitely demonstrated A+C-induced cholestasis. In cultured human Upcyte hepatocytes and HepG2 cells, CLAV was more cytotoxic than AMOX, and, at subcytotoxic concentrations, it altered the expression of more than 1,300 genes. CLAV, but not AMOX, downregulated the expression of key genes for BA transport (BSEP, NTCP, OSTalpha and MDR2) and synthesis (CYP7A1 and CYP8B1). CLAV also caused early oxidative stress, with reduced GSH/GSSG ratio, along with induction of antioxidant nuclear factor erythroid 2-related factor 2 (NRF2) target genes. Activation of NRF2 by sulforaphane also resulted in downregulation of NTCP, OSTalpha, ABCG5, CYP7A1 and CYP8B1. CLAV also inhibited the BA-sensor farnesoid X receptor (FXR), in agreement with the downregulation of FXR targets BSEP, OSTalpha and ABCG5. We conclude that CLAV, the culprit molecule in A+C, downregulates several key biliary transporters by modulating NRF2 and FXR signaling, thus likely promoting intrahepatic cholestasis. On top of that, increased ROS production and GSH depletion may aggravate the cholestatic injury by A+C.

B  Relation Synthesis Model Fine-tuning

As the CORD-19 dataset we use for demonstration is not large enough to train a relation synthesis model, we collected a training, validation, and test dataset from a subset of articles randomly selected from PubMed. All the target sentences have an RDS score greater than 0.75 and all the input sentences have an RDS score greater than 0.7. This resulted in a total of 615,561, 12,824, and 12,825 data in the training, validation, and test dataset respectively, which is comparable in size to the one used in DEER. We trained the model for 20 epochs. Other settings are the same as Huang et al. (2022b). In Section 5, we discuss our manual evaluation of the quality of generation.

C  Relation Synthesis with ChatGPT

The input to ChatGPT consists of a prompt and an overall relation context.

- The overall relation context is formed by concatenating the relation context on each edge with a new line character. The relation context on each edge starts with a description “Relation between [head_entity] and [tail_entity]: n” with [head_entity] and [tail_entity] replaced by the head and tail entity of the edge and is followed by the top 5 sentences on the edge sorted by the RDS score and concatenated by the new line character.

Using the sentences in Table 4 as an example, the input for ChatGPT is

Given the context below, describe the relation between COVID-19 and Vaccines in one sentence.

Relation between COVID-19 and Pneumonia: Coronavirus disease 2019 (COVID-19) is a novel type of highly contagious pneumonia caused by the severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2). Conversely, SARS-CoV, MERS-CoV, and COVID-19 may initially present asymptomatically, but can progress to pneumonia, shortness of breath, renal insufficiency and, in some cases, death.

Relation between Pneumonia and Vaccines: Despite the availability of safe and effective antibiotics and vaccines for treatment and prevention, pneumonia is a leading cause of death worldwide and the leading infectious disease killer. Despite advances in managerial practices, vaccines, and clinical therapies, pneumonia remains a widespread problem and methods to enhance host resistance to pathogen colonization and pneumonia are needed.