KG-COVID-19 Graph Analysis

Omkar Patil, CSE, UCSC

Abstract—A one-stop representation of COVID-19 data is an important step in helping to manage the current pandemic. Some issues to create this one-stop representation is combining and integrating all the knowledge from different datasets and documents into a single unified representation and a lack of a readily available visual tool which can help understand this knowledge. I therefore propose on such tool as a part of this project. The tool will rely on a rich knowledge graph at it's back-end and an interactive web based front-end to visualize the relations between different technical terms as learned from gaining knowledge from all the documents. The tool will build an interactive force-directed network graph to leverage the rich knowledge of KG-COVID-19 with an option to dynamically filter our particular individual as well as groups of relations between the nodes.

Index Terms—document clustering, knowledge graph, graph visualization, COVID-19, network graphs, force-directed graphs

1 Introduction

Severe acute respiratory syndrome coronovirus (SARS-CoV-2) is a novel coronavirus is the agent to the current COVID-19 global pandemic which started spreading in late 2019. However this is just one of the recent strains of beta-coronavirus, the others being SARS-CoV-1 and MERS-CoV (Middle Eastern Respiratory Syndrome). Since these are widely studied diseases, lots of clinical and epidemilogical data already exists for them. COVID-19 is a complex disease involving multiple genes which affect different biological processes. Typically some patients encounter a severe illness around a week after the initial symptoms which can lead to rapid degradation of respiratory functions. Along with several secondary health issues like organ damage, blood clots or cardiac failure. Given that the disease is relatively recent the research community is still long way from learning everything about COVID-19 and it's effects

Due to rapid mobilization and collaboration of the research community a large amount of biomolecular and medical data is now available in public domain and is constantly increasing. This includes the data about viral genome, gene function, epidemiological data as well as clinical case studies of similar coronaviruses like SARS-CoV-2 (e.g. SARS-Cov, MERS-CoV, etc). However one issue which is apparent is finding a way to make the data redily available to anyone of interest in a way that they may be able to use it right away. One key challenge in this regard is having a way of aggregating this data from different formats, locations and databases. Creating lots of specialized silos of related data but no readily available mechanism of identifying the relations between to different databases. Merely aggregating it might not be enough as most of these are complex technical and medical documents which might be difficult to understand to anyone outside their feild of specialization. Furthermore navigating between related documents might be a challenge as it's difficult to read through all the documents available. KG-COVID-19 proposes a novel way of aggregating this data into a single source by using Knowledge Graphs (KGs). KGs are a widely used method in NLP community to represent heterogeneous data points and their relationships using nodes and edges of a graph structure.

This project aims to aid in that regard by leveraging the advances in natural language processing and combining them with advanced visualization techniques to create an interactive web based application which can be used to aid in understanding this huge KG. It aims to use and enhance existing work of building COVID-19 knowledge graphs frameworks like KG-COVID-19 [1] to develop an interactive graph view which can be used to visually identify related nodes. Such visual cues might be an important precursor for the scientific community to come up with important hypothesis to further their research on COVID-19.

2 Motivation

Knowledge graphs are a way of representing data and it's relations and have been widely used in the NLP community in various text related applications. The nodes of these graphs are entities like drug names, protien identifiers etc connected to other entities using relations like 'causes to'. Visualizing these nodes and their edges can help one understand relations in a much faster and better way than merely reading through the text itself.

Additionally I personally believe making this viz interactive will help develop interest of other groups of people who wish to understand these entities but don't have the technical knowledge or deep understanding of the terms used in these papers. Making this data readily accessible might help further democratize this data which can help gather new ideas as well as spread awareness among the general public.

3 Related Work

As a part of democratizing the data many organization have released huge data-sets. One of the biggest public dataset is COVID-19 Open Research Dataset (CORD-19) [4]. It is prepared by the White House and a coalition of leading research groups. Allen Institute for AI has also hosted a Kaggle challenge for citizen data scientists to work on this data. The aim of this dataset is to identify answers to multiple questions which the community is currently working

on. I plan to use a subset of this dataset as a starting point to my document clustering and knowledge graph construction as it contains the most extensive machine-readabke coronavirus litrature collection available for data mining to date.
CoronaVis [5] is an attempt to use NLP to extract the information from tweets and then visualize it. They aim to study the psychological and behavioral aspects of the text to assist in managing the ongoing crisis. This project differs in many aspects from CoronaVis but will try to take motivations for using vis with NLP.
The KG-COVID-19 [1] Framework provides a rich and well structured resource for generating knowledge graphs related to the dataset. It also provides the code for generating the KG-COVID-19 as well as an already complied KG which is readily available for use. This work is already being used extensively by the research community. Authors of KG-COVID-19 use it to generate embeddings from graph vectors. These embeddings are then used to train machine learning models for link prediction like drug to disease, drug to gene and drug to protein. An interesting observation which they found was t-SNE plot of the embeddings showed that nodes tend to cluster around biological types without any external indicator. This is an important observation as it provides evidence of the quality of relations captured by the KG. Had the KG had defects in quality these embedding would vary greatly thus making it impossible for any clustering technique to cluster them together.
This project will extensively use and refer KG-COVID-19 material and aims to build upon their work. Specifically it looks towards better utilizing the information available in the KG. COVID-19 Visual Aid will be an improvement of the original visualization of KG-COVID-19 knowledge graph node embeddings which was done using t-SNE. Furthermore COVID-19 Visual Aid will also provide an option to drill down deep into any individual topic and see all the related documents along with multiple NLP features.

4 Dataset

KG-COVID-19 consists of 447,766 nodes and 21,611,628 edges from 14 different data sources. These data sources have a varied origin making the resulting knowledge graph a complex web of multiple interrelated entities. It contains Drug and chemical compound data from DrugCentral, PharmGKB, TTD, ChEMBL. Functional annotations and synonyms of coronavirus genes and proteins from Gene Ontology, UniProtKB, HGNC and SciBite-CORD-19. Ontology-based annotation from HPO and Mondo along with Biological systems like protein pathways from GO-CAM.

Source	Nodes	Edges
ChEMBL SARS-CoV-2 subset	4282	7357
STRING	21779	11759454
SciBite-CORD-19	139484	9257840
drug-central	3753	13900
gene-ontology	21	-
go-cams	155	3724
go-plus	78125	170375
hp	15657	19523
intact	2895	1203
mondo	41247	81906
mondo-ontology	3	-
pharmgkb	2453	5720
sars-cov-2-gene-annot	2528	46150
ttd	29089	82668
zhou-host-proteins	125	127
NBCI	-	17686

5 Preliminary Work

5.1 Choosing a proper subset of the KG

As the dataset is huge (450K nodes and 21 million edges) it is difficult to fit the entire dataset into memory. One of the initial problems was that the edges file was huge (4GB) and thus wasn't able to load it into the memory. As a result it was important to find an effective subset of the dataset to start working with. The nodes file is small (89 MB) so I started by taking a stratified split of 10 records based on their sources. These nodes were then matched with their corresponding edges to generate a trimmed-edges file containing 6433 edges. this file contains edges where either of the above 10 nodes are endpoints of the edge. Thus it is important to generate the back connections to these edges by selecting the remaining nodes which occur in edges (apart from original 10) to generate trimmed-nodes file.
The original node and edges files were eventually loaded into the memory using a High-RAM machine on Google Colab. The machine used had a RAM of 25G and a similar machine is recommended to load this dataset if using the existing code.
The current loading system divides the edges into two files with a suffix of head and tail to make it easier to load it into the memory. Head contains the first 10,000,000 edges while tail contains the remaining 11,443,190 edges. Both these files are loaded as separate dataframes and 21 out of the original 49 columns are dropped. These dropped columns usually either contain numerical data or very low number of categorical data (0.0001%). Thus dropping them doesn't affect the categorical variability of the dataset. All the columns containing external identifiers are preserved for any future use case. Similarly 12 out of the original 29 columns of the nodes dataframe are dropped in similar fashion as the edges.
Dropping these columns helps reduce the size of the dataframes and makes it easier to perform various analysis or processing without the system running out of memory.

5.2 Identifying visualizations for KG

Bubble chart is a multi-variable chart which can be thought of as a combination of proportional area charts and scatter

plots. They are normally used to compare and show the relationships between categorical attributes in the form of circles of different sizes. The size of the bubble can be used to visually emphasize the numerical value of the category (count in our case).
Force-directed graphs are one of the good visualization techniques for drawing graphs in an aesthetically-pleasing way. One of the key constraints while drawing these graphs is to position nodes in such a way that the edge overlap is minimum. This is particularly useful in our scenario as the goal of this project is to develop a tool which can be used by the community to visually analyse the KG. It is also easy to move around the nodes of the graph and thus can be an important technique to provide interactivity.
ObservableHQ [6] provides a beginner friendly way to create D3.js visualization along with multiple examples from community. One of the good things about this tool is that external code from other notebooks can be readily imported and used as is which significantly reduces development time for new visualizations.

Fig. 1. Example of a basic force-directed graph from the dataset

6 Data Exploration and Analysis

This was the most time consuming part of the project as it involved identifying possible use cases upon which the vis were developed. It included a very detailed analysis of the contents of the dataset with lot of exploratory filtering.
The final processes which were used for the vis are summarised below.

6.1 Creating categorical counts

This uses the inbuilt value counts method available in pandas to identify counts of various categorical variables. Some rows contains multiple categories separated by pipe. for such rows the it is counted in both the categories.

6.2 Creating trimmed knowledge graphs

This involves a multi-step process. Filter function identifies all nodes in the KG containing the keyword. it then matches all the edges containing the nodes either as source or target. then the node list is updated with corresponding nodes in the edges.
Due to the connected nature of the graph, once we have identified all the edges originating from a particular node, It is also important to add the destination nodes to our list otherwise the edge doesn't make any sense.
Another problem which can be encounter as we add additional nodes is that they may be connected to more edges. Thus it is important to limit the degree of separation from our original nodes. This is done use depth parameter. It controls the number of iterations the find, match and update functions may be run.

6.3 Input format for vis

The input to the vis is in the two forms.
For bubble charts it is a csv file with two columns containing the name and count of the categories. For force graph it contains two list of nodes and edges. each of the node contains an id. edges have the corresonding id in either their source or target attributes.
It was found during the exploration of the KG that around 65,000 unique node ids mentioned in the edges file did not contain a corresponding id in the node file. Thus, further filtering is done to remove edges where a matching node id is not available.

6.4 Connecting to external knowledge sources

The node id is a combination of two terms. The original source and it's id in the source. Thus this source information is used to connect the id to it source webpage. The current vis supports 8 of the original 14 sources with more to be added further.

7 Existing/Preliminary Vizualization

The interactive dashboard does a good job of summarizing the origins and relations of all the nodes and entities. Some examples can be seen below.

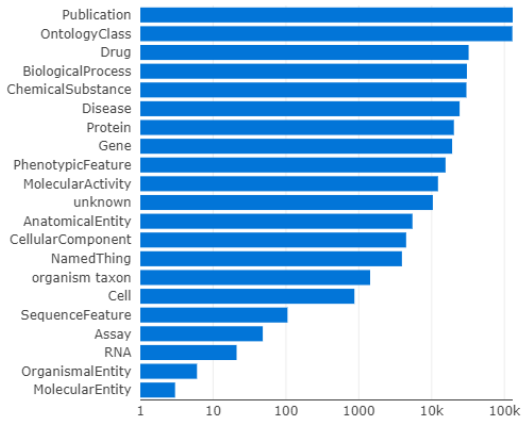

Fig. 2. Overview of all the node categories

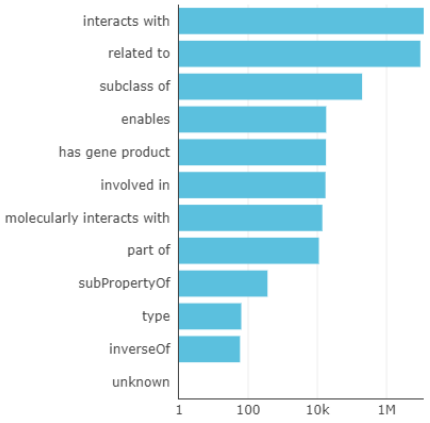

Fig. 3. Overview of all the edge categories

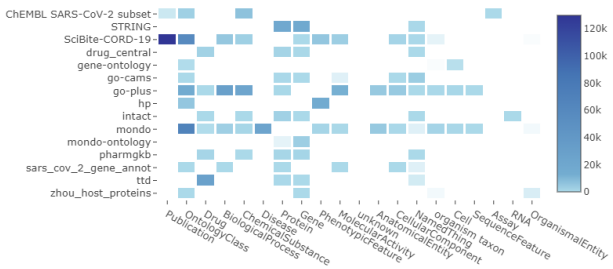

Fig. 4. Heatmap of node categories by source

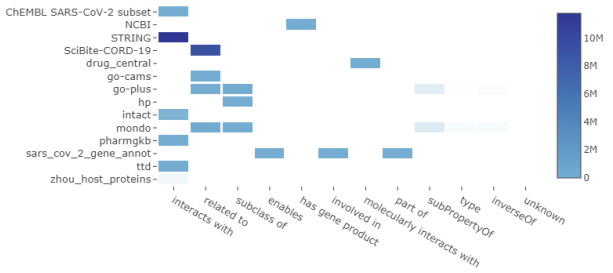

Fig. 5. Heatmap of edge categories by source (Top 20)

However this dashboard fails to provide any interactivity to the user and is fairly static. It can be useful to get a very macroscopic overview of the dataset but nothing further. Thus developing interactive visualizations which can help explore the data on a macroscopic as well as microscopic level is important.

8 Final Vizualizations

This project provides three main types of visualizations. All of them are powered by data generated from the data generation notebook. Interactions are provided in the form of on click tool-tips and features like hovering, zooming and dragging.

8.1 Bubble & Bar Charts

This vis generates a bubble chart along with the corresponding bar chart for a particular categorical attribute. The attribute can be either from nodes or edges. Hovering on any of the bubble gives a count of that particular category. The different attributes can be chosen from a drop down list of data files. An example can be seen in Fig. 6 and Fig. 7.

Fig. 6. Bubble chart for node categories attribute

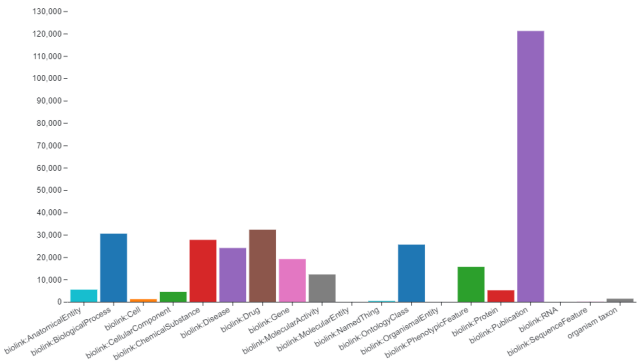

Fig. 7. Corresponding bar chart for node categories attribute

8.2 Force-Directed graph on trimmed KG

This vis also used the data file generated from the KG to render a graph. Color scheme is used to determine node colors based on their category, all the nodes belonging to a particular category will have the same color. Nodes and edges display a tool-tip on clicking on them. The tool tip contains additional information about a particular node or edge. Some of the edges have a publication associated with them, In these cases the tool tip also provides a hyperlink to the original publication mentioned in the edge e.g. Fig 8.

Fig. 8. Example of an edge tool tip

For nodes the name provides a link to the wikipedia article of the node if available. The id maps to the description of the particular node in the original online knowledge base e.g. Fig 9.

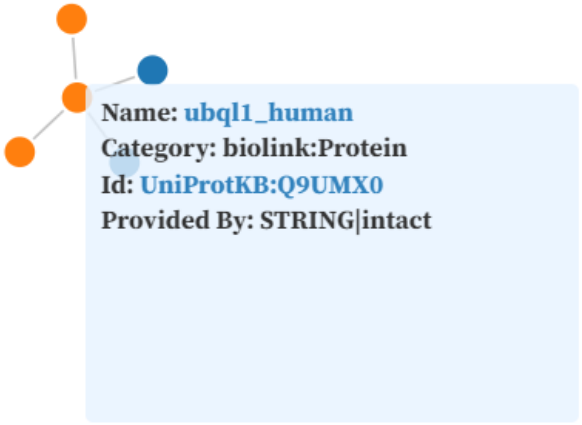

Fig. 9. Example of a node tool tip

8.2.1 Graph Filters

The above examples are of extremely trimmed down nodes. It's is difficult to find nodes which are this lightly connected in the KG. Often times there can be thousands of nodes and edges present on the graph which may make analyzing them a challenge. In order to help with this the tool provides multiple node and edge level filters which can be used to remove certain elements from consideration for time being. The list of all the available filters is present in Fig 10.

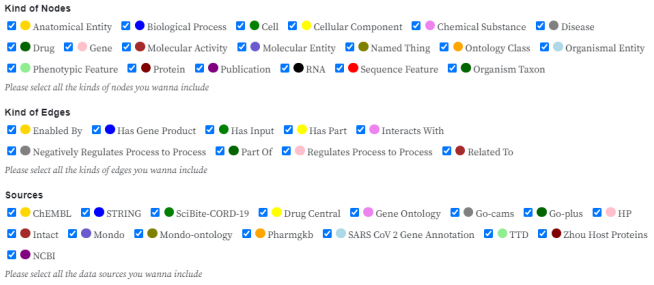

Fig. 10. Graph Filters

An example of the benefit of using filters is shown in Fig 11. As we can see the graph is huge making it difficult to identify relations between other nodes. On inspection we find that most of these nodes are of category ″biolink:Publication″. Thus if we filter that category out we may be able to identify relations between other categories. A filtered view of the same graph is show in Fig 12.

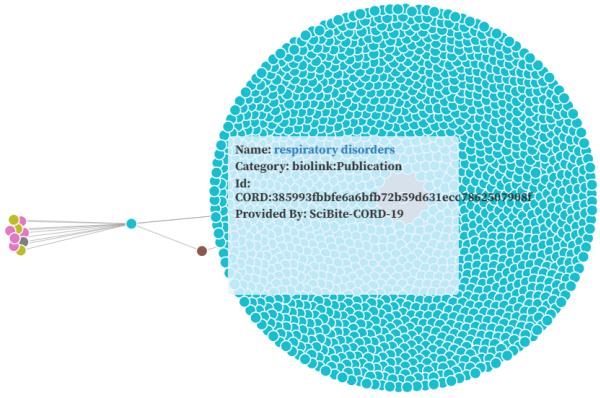

Fig. 11. Original force graph for a particular search term

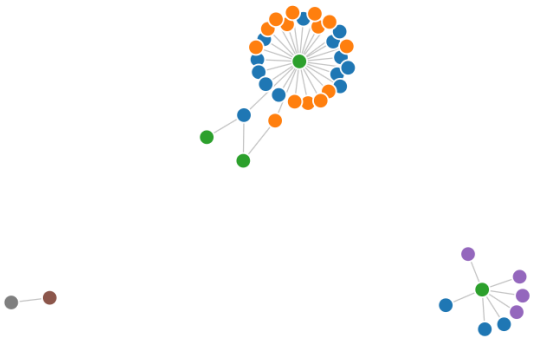

Fig. 12. Filtered graph for the same term

8.3 Node exploration graph based on Wikipedia page redirects

This vis is populated based on the node click event which display the node tool tip. It searches a subset of Wikipedia

based on the node name. If it finds any matching page the vis then further visits other pages mentioned the the given page to build a directed graph of all the pages. This gives the user a brief idea of the node and it's uses. Along with the graph this vis also extracts the intro paragraph of the page and display it.
Example of this vis can be found in Fig. 13 and it's exploration in Fig. 14.

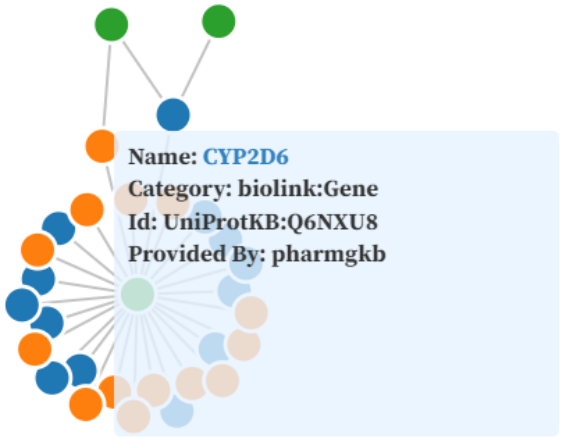

Fig. 13. Node CYP2D6 is clicked which is a gene.

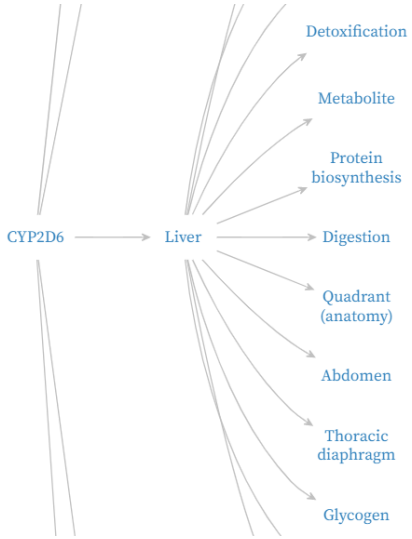

Fig. 14. Corresponding directed graph generated implies it may be related to Liver among other things.

9 Future Work

Creating a document view for each of the nodes in the KG. The DocView will use NLP techniques to help people understand technical terms using named entity recognition and it's mappings in KG-COVID-19 can be a possible expansion to this project. Automatic annotations of technical document like papers to original reference materials can be done using the nodes and then connected to the vis. Identifying similar or conflicting views/approaches and opinions between the original documents based on the vis can be an ambitious goal.

10 Conclusion

Thus we build a tool which can be used to interactively explore KG-COVID-19 which can help combat the spread and control the disease more effectively. The structure of this tool is modular enough to accomodate any other data sources with minimal changes.

References

[1] Reese J, Unni D, Callahan TJ, et al. KG-COVID-19: a framework to produce customized knowledge graphs for COVID-19 response. Preprint. bioRxiv. 2020;2020.08.17.254839. Published 2020 Aug 18. doi:10.1101/2020.08.17.254839
[2] Juan Gómez-Romero, Miguel Molina-Solana, Axel Oehmichen, Yike Guo, Visualizing large knowledge graphs: A performance analysis, Future Generation Computer Systems, Volume 89, 2018, Pages 224-238, ISSN 0167-739X, https://doi.org/10.1016/j.future.2018.06.015.
[3] https://towardsdatascience.com/automated-adverse-drug-event-ade-detection-from-text-in-spark-nlp-with-biobert-837c700f5d8c
[4] https://www.kaggle.com/allen-institute-for-ai/CORD-19-research-challenge
[5] Md. Yasin Kabir, Sanjay Madria CoronaVis: A Real-time COVID-19 Tweets Data Analyzer and Data Repository Preprint. arxiv. 2020; 2004.13932 https://arxiv.org/pdf/2004.13932.pdf
[6] https://observablehq.com/explore