Team 4: Reda Mastouri & Kalyani

10/13/2021

Saint Peter's UNIVERSITY

The Jesuit University of New Jersey

Milestone 3

Project plan and management for the selected topic:

Disease State Conversation and Sentiment Analysis:

Examination of Digital Community Conversations Within Specific Disease States Via Reddit

Modeling

CONTENTS

Topic Modeling

Used Techniques

Algorithms in Scope

Visualization Techniques (WIP)

Perspective

App: Proof-of-Concept

appendix

TOPIC MODEL?

What is it?

A type of statistical model for discovering the abstract "topics" that occur in a collection of documents.

A document typically concerns multiple topics in different proportions; thus, in a document that is 10% about prevention and 90% about medication, there would probably be about 9 times more drugs prescription words than preventive advices words.

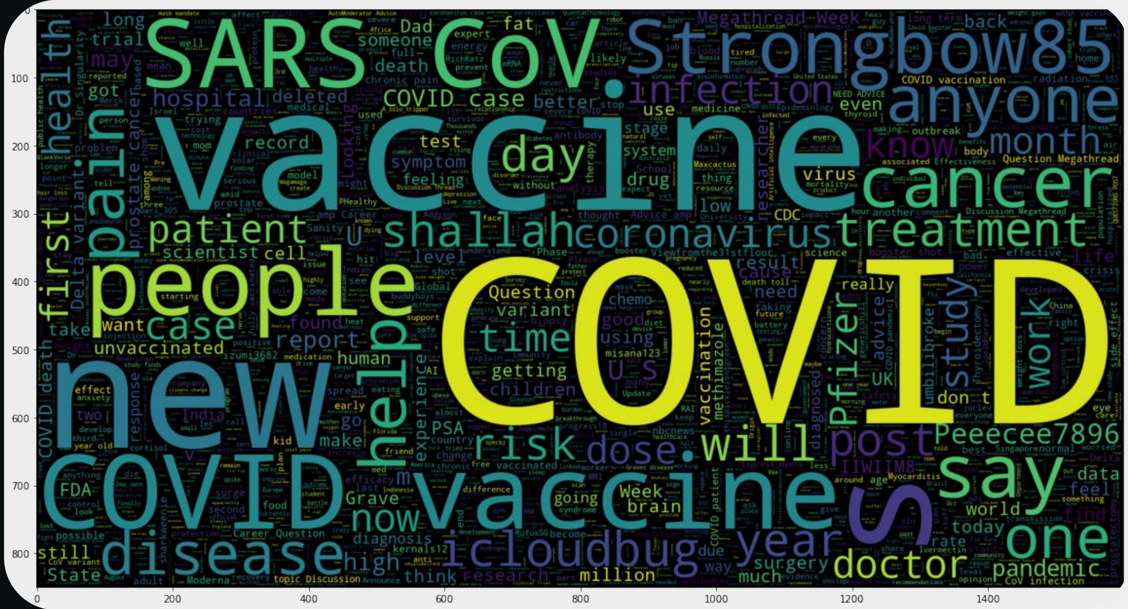
The "topics" produced by topic modeling techniques are clusters of similar words. A topic model captures this intuition in a mathematical framework, which allows examining a set of documents and discovering, based on the statistics of the words in each, what the topics might be and what each document's balance of topics is.

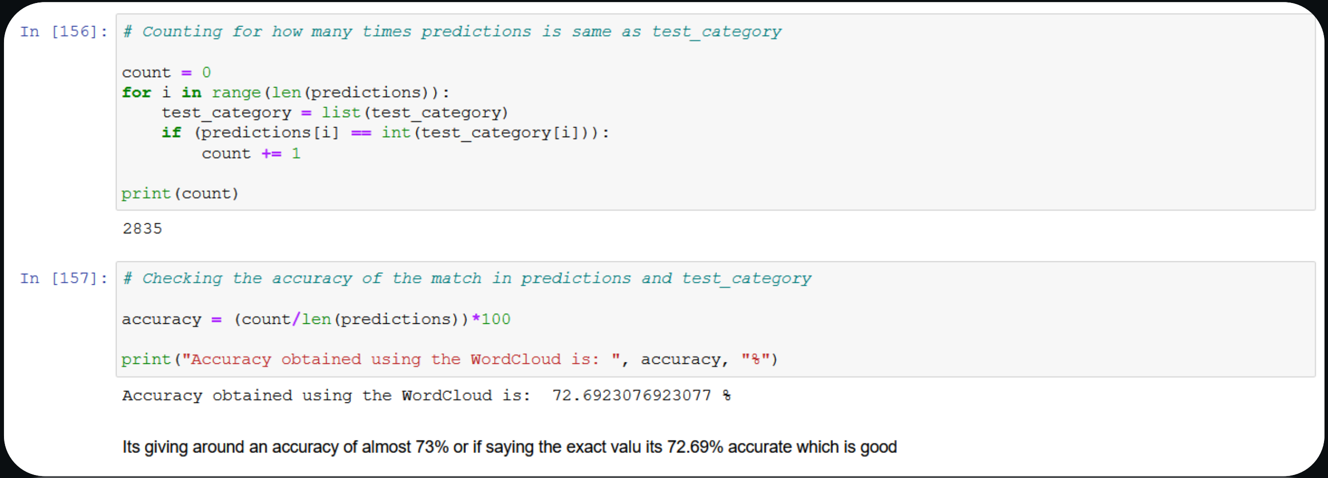
From last milestone

Complete data preprocessing and EDA

From feature engineering through word vector and data lemmatization and tokenization...

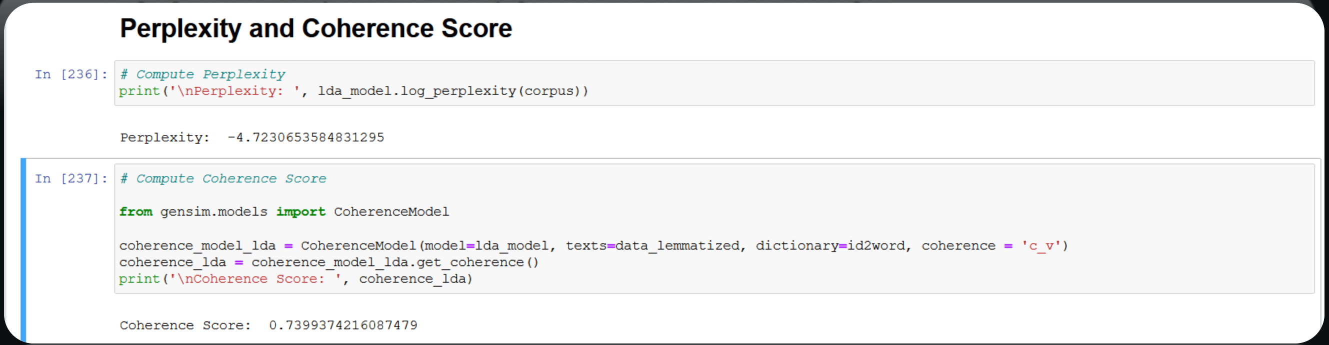
We are now about to apply the various approaches to data modeling and fine tuning.





```
In [156]: # Counting for how many times predictions is same as test_category

          count = 0
          for i in range(len(predictions)):
              test_category = list(test_category)
              if (predictions[i] == int(test_category[i])):
                  count += 1

          print(count)

          2835

In [157]: # Checking the accuracy of the match in predictions and test_category

          accuracy = (count/len(predictions))*100

          print("Accuracy obtained using the WordCloud is: ", accuracy, "%")

          Accuracy obtained using the WordCloud is:  72.6923076923077 %
```

Its giving around an accuracy of almost 73% or if saying the exact valu its 72.69% accurate which is good



Perplexity and Coherence Score

```
In [236]: # Compute Perplexity
          print('\nPerplexity: ', lda_model.log_perplexity(corpus))

          Perplexity:  -4.7230653584831295

In [237]: # Compute Coherence Score

          from gensim.models import CoherenceModel

          coherence_model_lda = CoherenceModel(model=lda_model, texts=data_lemmatized, dictionary=id2word, coherence = 'c_v')
          coherence_lda = coherence_model_lda.get_coherence()
          print('\nCoherence Score: ', coherence_lda)

          Coherence Score:  0.7399374216087479
```

73%

Accuracy based on WordCloud prediction

2835

topics

74%

Coherence Score

-4.72

Perplexy

Algorithms in Scope

Viable set of algorithms

Popular topic modeling algorithms include Latent Semantic Analysis (LSA) a.k.a Latent Semantic Indexing , Hierarchical Dirichlet Process (HDP), Latent Dirichlet Allocation (LDA) and Non-negative Matrix factorization among which LDA has shown great results in practice and therefore widely adopted.

1. Latent Semantic Analysis
2. Latent Dirichlet Allocation
3. Hierarchical Dirichlet Process
4. Non-negative Matrix factorization

Techniques implemented

From StopWords to Perplexy computation

LDA represents documents as mixtures of topics that spit out words with certain probabilities.
Latent Dirichlet Allocation (where documents are represented by latent topics, and topics are represented by a distribution over words) Non-negative Matrix Factorization (where a document-term matrix is approximately factorized into term-feature and feature-document matrices).

Will check this on the jupyter Notebook

Visualization Techniques

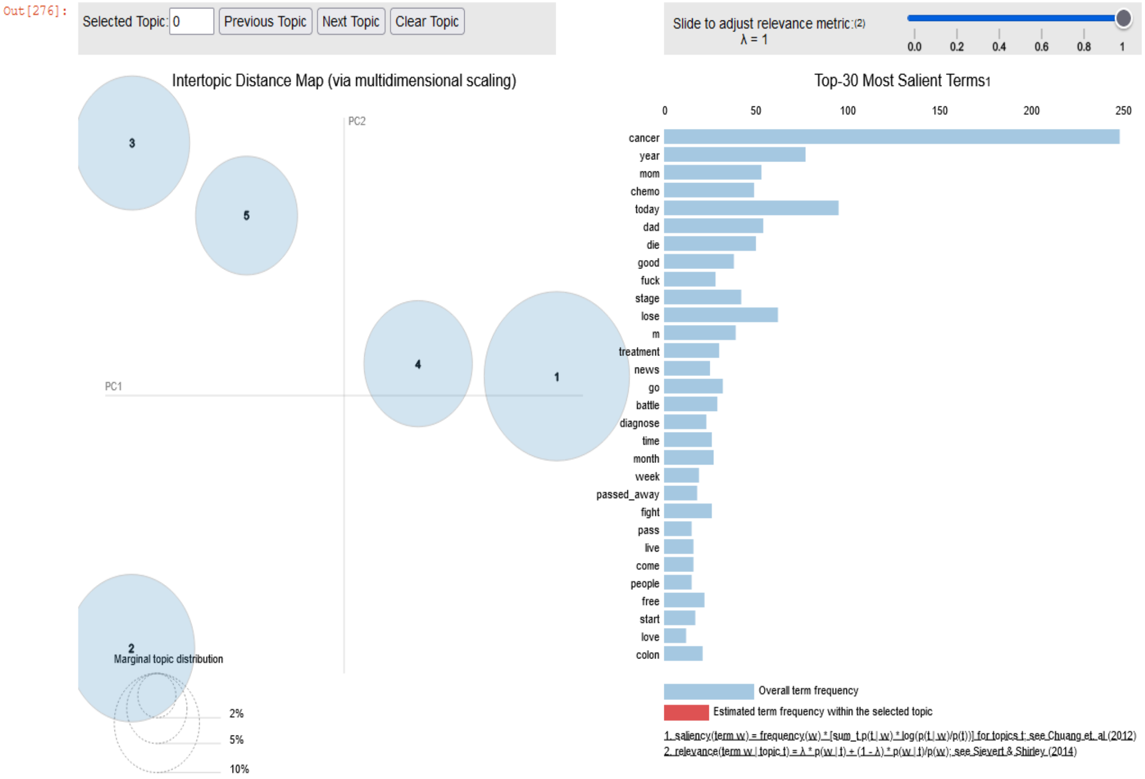
From StopWords to Perplexy computation

Livestream Reddit via PRAW

1. *Livestream comments submitted within Subreddits or by Redditors*
2. *Livestream submissions submitted within Subreddits or by Redditors*

Analytical tools for scraped data

1. *Generate frequencies for words that are found in submission titles, bodies, and/or comments*
2. *Generate a wordcloud from scrape results*



1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)

2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)

Models: LDA

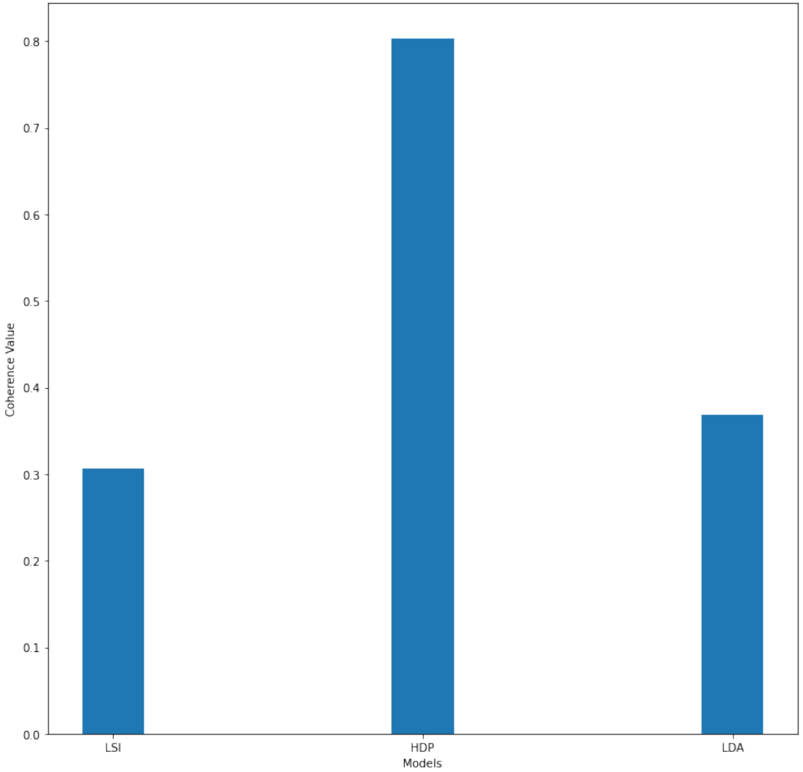
Implementation

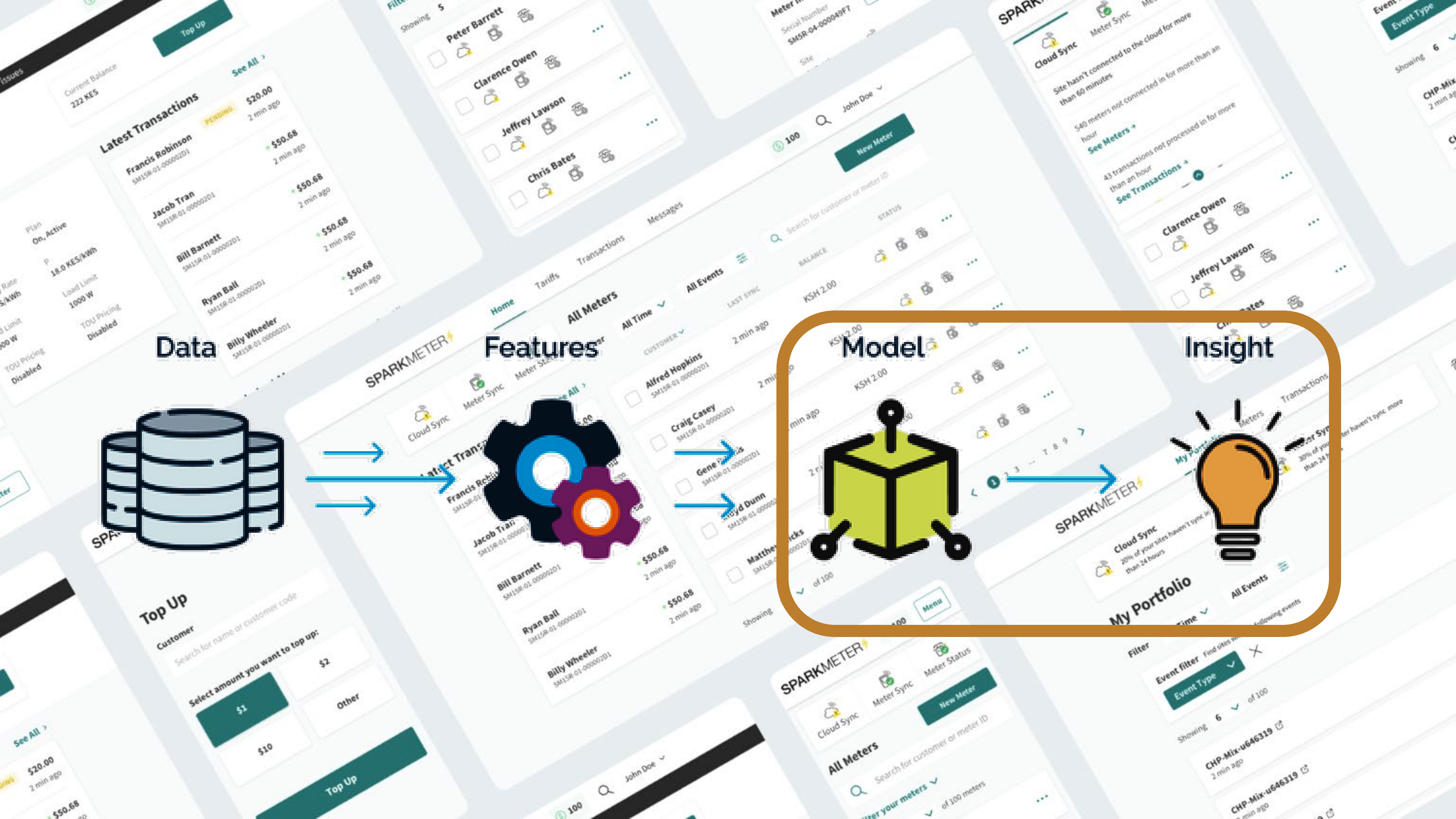
Check the notebook

Evaluation

Comparative study

1. Latent Semantic Analysis
2. Latent Dirichlet Allocation
3. Hierarchical Dirichlet Process
4. Non-negative Matrix factorization





Data

Features

Model

Insight

Perspective

What's next?

Considering more of Livestream Reddit via PRAW

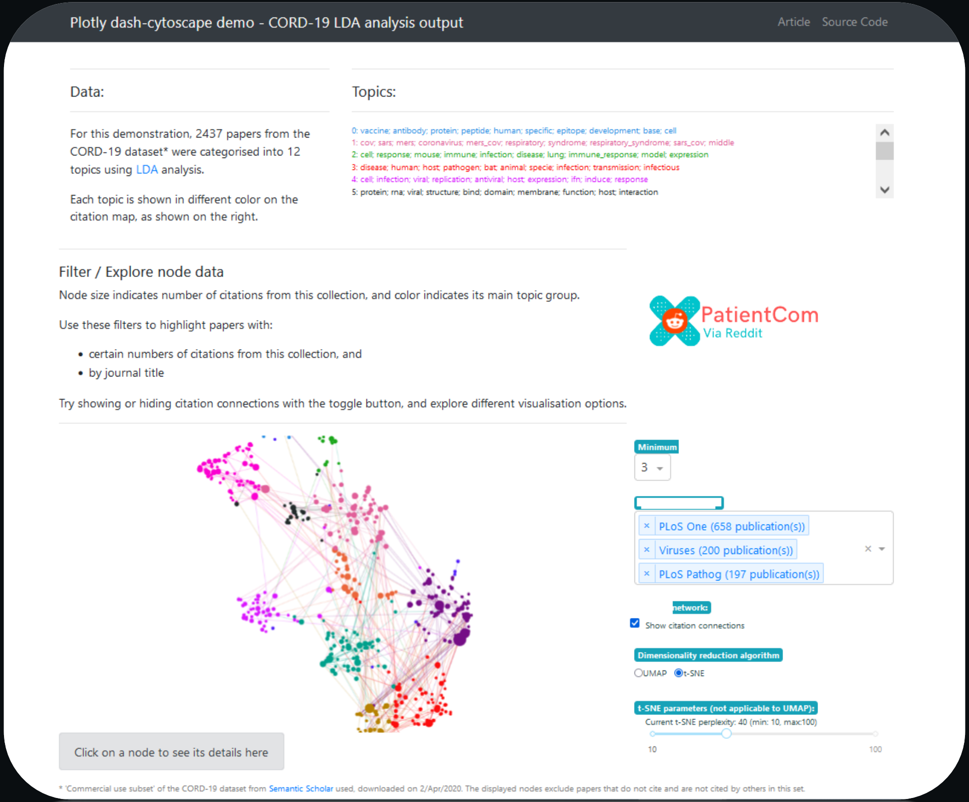
1. *Livestream comments submitted within Subreddits or by Redditors*
2. *Livestream submissions submitted within Subreddits or by Redditors*

Analytical tools for scraped data

1. *Generate frequencies for words that are found in submission titles, bodies, and/or comments*
2. *Generate a wordcloud from scrape results to be included in the final app*

App: Proof-of-Concept

Via Dashly



Plotly dash-cytoscape demo - CORD-19 LDA analysis output

Data:

For this demonstration, 2437 papers from the CORD-19 dataset* were categorised into 12 topics using LDA analysis.

Each topic is shown in different color on the citation map, as shown on the right.

Topics:

0: vaccine; antibody; protein; peptide; human; specific; epitope; development; base; cell
1: cov; sars; mers; coronavirus; mers_cov; respiratory; syndrome; respiratory_syndrome; sars_cov; middle
2: cell; response; mouse; immune; infection; disease; lung; immune_response; model; expression
3: disease; human; host; pathogen; bat; animal; specie; infection; transmission; infectious
4: cell; infection; viral; replication; antiviral; host; expression; ifn; induce; response
5: protein; rna; viral; structure; bind; domain; membrane; function; host; interaction

Filter / Explore node data

Node size indicates number of citations from this collection, and color indicates its main topic group.

Use these filters to highlight papers with:

- certain numbers of citations from this collection, and
- by journal title

Try showing or hiding citation connections with the toggle button, and explore different visualisation options.

PatientCom
Via Reddit

Minimum: 3

× PLoS One (658 publication(s))
× Viruses (200 publication(s))
× PLoS Pathog (197 publication(s))

network:

☑ Show citation connections

Dimensionality reduction algorithm

○UMAP ●t-SNE

t-SNE parameters (not applicable to UMAP):

Current t-SNE perplexity: 40 (min: 10, max:100)

Click on a node to see its details here

* 'Commercial use subset' of the CORD-19 dataset from Semantic Scholar used, downloaded on 2/Apr/2020. The displayed nodes exclude papers that do not cite and are not cited by others in this set.

Thank You

Please feel free to ask any question