Team 13: COVID-19 Research Articles Analysis and Visualization - Final Report

Dongjin Choi, Tina Yuting Guo, Seil Kwon, Isaac Thomas Rehg, Andrew Wang, Chaerin Yoo

1 INTRODUCTION

Recently, CORD-19 [22], a dataset of over 200,000 COVID-related scientific documents, was assembled to promote research into new techniques to search this ever-growing number of papers. Our group is interested in designing a **dashboard system** with an **automated pipeline** that summarizes COVID-19 research article data set and to encourage user's exploration.

1.1 Problem Definition

We have unstructured text data set of **over 200,000** bio-medical research articles' abstracts and body texts which has a size of 20 gigabytes in raw format. This includes meta data of authors, affiliations, citations, publication venues. Given the data set, we aim to build an interactive visualization system while answering following Questions:

(1) (Cluster View) How can we provide inclusive visualization of clusters of large number of documents?
(2) (Citation View) How can we provide visualization of large citation graph of articles?
(3) (Text View) How can we efficiently search relevant articles from large number of documents?
(4) (QA View) How can we retrieve relevant answers for a natural language query?
(5) (QA View) How can we efficiently log a user's feedback for the answers?

2 SURVEY

Graph Visualization. Argo Lite [12] proposed a fast graph visualization system by rendering via WebGL. This system will be a useful reference for visualizing large citation network in our project. We will improve it to represent textual similarity between articles in visualization. [13] proposed a method to visualize research papers considering both topics and citation network. They proposed a novel method to calculate the shape of node clusters and edge bundling. This paper will be directly related to our project. To build our citation network, we could refer to other citation network tools such as CitNetExplorer [5] and try to apply some of the network building methods that were used. E. Segel and J. Heer [18] also show various strategies to effectively narrate the story behind visualization.

Document Clustering. The CORD-19 [22] dataset consists of 200,000 journal articles and research papers. For clustering, approaches such as LDA [3] or TF-IDF indexing [6] are commonly used. [6] utilized TF-IDF indexing to obtain clusters for biomedical research documents, finding them to provide meaningful groupings when validated by an expert. Other popular methods include K-Means [23] and Nonnegative Matrix Factorization (NMF) [21]. NMF creates easily explainable results [21]. In addition, Tucker Decomposition utilizes tensor decomposition to conduct static malware analysis on a large scale [14]. Moreover, [17] makes an attempt to extract factual relations from scientific papers.

Question Answering System. Question Answering is a relatively new in NLP field. This new research has led to the development of high quality datasets for generic question answering [15]. In QA, curating new datasets is costly, especially when experts are required. Recent efforts have led to the development of QA dataset for COVID-19 [19] by drawing from a large body of related papers. This dataset can be used as a test set to validate performance of existing models. We plan on leveraging models trained in generic biomedical domains [2, 11] to conduct Question Answering over the documents we visualize, after validating its performance on CovidQA. [19].

Recommendation System. One possible future extension of this project is to implement a recommendation system based on the citation network of papers [24]. The paper introduces an Eigenfactor Recommends (EFrec) that uses a citation network to find which papers are relevant to the seed paper and decides which ones are important. As of now, this method seems uneasy to implement due to its complex nature. However, it could be an interesting feature that could be added on the citation network that we will build for this project in the future.

Information Extraction From Data Cloud and Word Detection. With dataset consisting of 200,000 journal articles and research papers, the method in [16] to extract information might be useful. Information is usually clear to the author but not to computers. The extractors aim to bring such a structure. We can group papers with similar information. In [1], the author aims to group data together, determine the stem of words and also identify the actual insight of the word/sentence. With that, we can group articles better. In [7], a novel method to index a document by SBERT, TF-IDF and BM25 was proposed. It's ranking mechanism is highly relevant to our topic. However, this method might not be available in real-time.

3 PROPOSED METHOD

3.1 Document Clustering

For the cluster view, we must project the documents into a low-dimensional manifold where related documents are close together so that the user can identify groups of topically similar documents. To validate this projection and provide another medium for visualizing document similarity, we cluster documents and color-code points in the plot by corresponding document's cluster.

Preprocessing. Before conducting our analysis we remove any documents with missing abstracts or body text. We remove documents written in languages other than english for simplicity and because such documnets constitute a very small portion of the dataset ($< 1\%$). Our document encodings leverage word-occurrence statistics, so next we conduct preprocessing to remove common words from each document that would not be useful in characterizing a document's topic.

Document Representation. We choose to represent documents by term-frequency inverse-document-frequency (TF-IDF) index vectors. This representation models each document as a vector of word-occurrences for that document relative to word-occurrences across the set of all documents. Specifically, for the vector representation $\mathbf{v}^{(g)}$ of document g, we have

$$\mathbf{v}_i^{(g)} = f_{g,i} \times \log\left(\frac{N}{\sum_{h=0}^{N} \mathbf{1}(f_{h,i} > 0)}\right) \tag{1}$$

where $f_{h,i}$ is the number of occurrences of word$_i$ in document h.

Because this vector of normalized word occurrences lives in a very high-dimensional space, we apply PCA and then embed document vectors using t-SNE to make the 2-D visualization more informative. Using PCA on the TF-IDF vectors we can reduce dimensionality from 4096 to 1982 while retaining 95% of the original variance. Visualizing these 1982-D vectors in 2 dimensions is still ineffective, so we further reduce dimensionality using t-SNE embeddings [20]. t-SNE fits a lower-dimensional embedding vector $\mathbf{y}_i$ for each data point $\mathbf{x}_i$, such that the conditional distribution $\mathrm{p}(\mathbf{y}_j|\mathbf{y}_i) = \mathrm{p}(\mathbf{x}_j|\mathbf{x}_i)$ is preserved. The resulting embeddings preserve the local manifold structure of the data while existing in a much lower-dimensional space.

Clustering. We explore both k-means clustering on the PCA-compressed TF-IDF vectors and clustering by Latent-Dirichlet Allocation (LDA). In LDA a latent topic variable is modeled to maximize likelihood of document-word occurrences. We assign each document to the cluster topic with maximum likelihood in the predicted topic distribution for that document.

3.2 Graph Analysis

Our other goal is to find the citation networks between articles and authors, perform an analysis on the graphs, and eventually effectively visualize the graphs to users. We define our target graph as follows:

Definition 3.1. Let $\mathbf{G}$ a directed graph representing citation between papers.
$\mathbf{G}_p.node = \{paper | \forall paper \in \mathcal{D}\}$ where $\mathcal{D}$ denotes our data set, and $paper$ is each paper that has been cited at least once by another node.
$\mathbf{G}_p.edge = \{(source, target) | \forall (source, target) \in \mathcal{D}.citations\}$ where $(source, target)$ denotes a citation from $source$ paper to the $target$ paper.

Preprocessing We preprocessed our dataset to find a unique identifier of papers to match each paper's bibliography into each paper. One problem we encountered was that the raw bibliography textual data is quite noisy, so titles sometimes include html tags and punctuations. We transformed paper titles and bibliographic items' titles by taking lowercases and eliminating punctuations. We assumed two paper titles as same if the edit distance between the two is less that 5. We construct a set of edges based on their citation relationships.

Network Centrality Analysis For network analysis, we mainly looked into degree centrality/in-degree centrality. Degree centrality is a network centrality measure regarding the degrees of the nodes. Having a high degree indicates that the node has a high degree

centrality.[8] In-degree centrality uses the same definition, but only takes the in-degrees into account. These two centrality values would help analyze the most cited papers in the network.

We also used VoteRank centrality, which reflects the node's capacity for spreading news or influence over a network. It utilizes the concept of votes between connected nodes, and the voting score of a node is determined by the sum of the voting ability of its neighbors.[26] So the VoteRank centrality will also take into account the influence that a node could have on its neighbors, which could determine which papers were truly influential based on the citation network.

To calculate the VoteRank, we needed to take an additional step during the process. In a directed graph, the VoteRank algorithm uses the out-degree as having voting ability.[26] On the other hand, the out-degree of our citation network denotes when a source paper is citing a target paper. This means that a paper which is cited by many other papers would have a large number of in-degrees. Therefore, in order to properly apply the VoteRank algorithm, we first needed to reverse all the edge directions.

All of these centrality measures were calculated using the NetworkX Python package.[10] To help visualize the network analysis data, we utilized other additional plotting tools such as MatPlotLib and Seaborn.

Graph Visualization To visualize processed graph, we utilize two different external tools. First, we used networkx [9] with a constant node size, and showing directions as arrows. We further utilized ArgoLite, which we learned from the course. This tool has an advantage that it automatically computes the pageRank scores of nodes and scale nodes' sizes based on the score. We will compare different visualizations and utilize appropriate visualization method for our final model.

3.3 Question Answering

We used a deep-learning based pre-trained model for automated question answering. The model is provided by HuggingFace [25] library, and the model utilized a BERT [4] model which was pre-trained on the PubMed corpora. The QA model takes a pair (question, context) as an input and it outputs the most probable output snippet from the context answering the question. It's nearly intractable to generate all possible input (question, context) pairs from data set and evaluate a deep model to generate desired output. Thus, filtering original document set and taking a relevant subset of data is very important. We propose a method to select a subset based on TF-IDF similarity between query sentence and papers' abstracts. We select top-7 articles based on the cosine similarity and only use those articles as candidates of answers.

4 PLAN & WORK DISTRIBUTION

Previous Activities. All team members worked on the proposal and progress report. Jin wrote the skeleton of this progress report. Isaac pre-processed the initial CORD-19 data, and worked with Seil and Andrew on clustering. Tina and Chaerin worked with Jin on the citation network and Q&A system. Everyone will work on the dashboard system and final presentation. All team members have contributed similar amount of effort.

Final Activities & Work Distribution. Final project activities include updating and implementing the code, conducting usability test, writing the final report, and creating the poster. The work distribution breakdown is as follows:

Activities	Main Owners	Secondary Owners
Implementation	Isaac, Jin	Tina, Sherry
Final Report	Andrew, Tina	Jin, Seil
Poster	Sherry, Seil	Isaac, Andrew
Usability Test	All	–

5 EXPERIMENTS & EVALUATION

To evaluate our product, we performed a usability study with five evaluation questions. Participants for the usability test were able to choose their answer on a 5 point scale where 5 is strongly agree and 1 is strongly disagree.

Usability Test Questions.

(1) I think I would like to use this system frequently
(2) I found the QA tools very helpful
(3) I found the various functions in this system to be well integrated
(4) I think most people could learn to use this system quickly
(5) I found the visualization tools very helpful

Experiment Observations. Based on 23 participants, over 70% found that our solution is easy to use and navigate. The majority of participants also found each individual functionality to be helpful for finding COVID-19

information and exploring papers with over 75% finding the clustering and graph visualizations to be very helpful.

One area of improvement for our product is the integration of functionalities for a more seamless user experience. While users found the product easy to learn and use, many were looking for more summarized responses from the article search and wanted to see the loading wheel as functionalities took time to load.

Figure 1: Usability Test Q1(left) Q2(right) Results

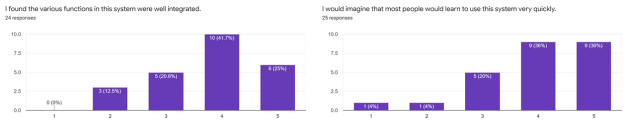

Figure 2: Usability Test Q3(left) Q4(right) Results

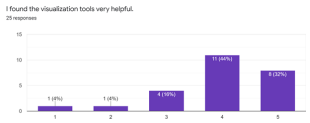

Figure 3: Usability Test Q5 Results

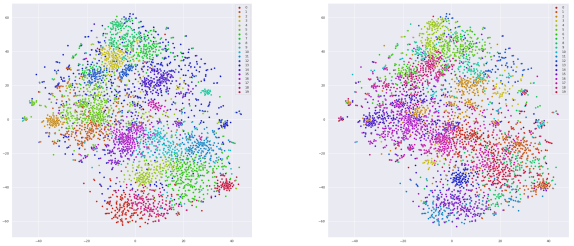

Figure 4: K-means and LDA clustering. Documents are clustered into 20 different groups by using K-means(left) and LDA(right) methods.

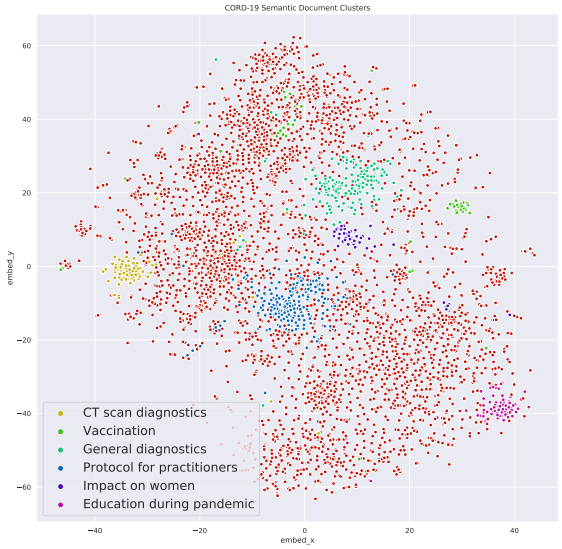

Figure 5: Tightest clusters produced by K-means clustering, labeled by the corresponding article topic.

6 RESULTS & DISCUSSION

Document Clustering. For our pilot version, we have conducted clustering using a variety of algorithms on a smaller (10k sample) subset of the entire CORD-19 dataset. We have evaluated document clustering using K-means clustering and LDA topic modeling approaches. See 3.1 for detailed description of each of these approaches.

While both clustering approaches can isolate tight groups of data points in the embedding space, we find k-means to produce cluster assignments with less overlap, overall. Figure 5, highlights the 6 article topics for specific clusters. While clustering was able to identify topics for tight clusters, it was difficult to identify a unified topic for less tight-knit clusters.

Graph Analysis. We have leveraged the same 10K subset of CORD-19 data to create the target article and author graphs using the approach from 3.2.

Our analysis shows that an overwhelming number of articles referenced the paper, "Epidemiological and Clinical Characteristics of the Early Phase of the Covid-19 Epidemic in Brazil", which was referenced a total of 156 times. In comparison, the second most referenced citation has 38 references which is only 1/4 that of the

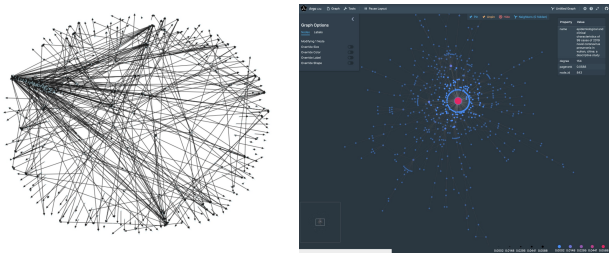

Figure 6: Visualizations of citation graphs. Network's visualization (left) visualizes nodes as same size, edges' direction with arrows, while ArgoLite's visualization (right) shows nodes' size proportional to their pageRank scores and edges as undirected.

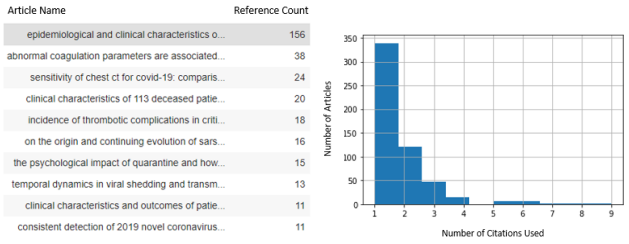

Figure 7: Top-10 Cited Articles (left). Citation # Distribution (right).

top citation. These results were also reflected in the in-degree centrality values shown in Figure 8(left). The highest in-degree centrality value was approximately 0.114, while all the following values were measured to be almost negligible compared to the highest centrality value.

VoteRank results were also similar for the top 3 papers, but had some different results in the following papers, which can be seen in Figure 8(right). When trying to determine the influential papers based on the citation network, it could be more adequate to use VoteRank instead of degree centrality. In the end, we hope to incorporate all these varying analysis measures in our final product, as different means to interpret and analyze the citation network.

We then analyzed the number of citations used by each article. Surprisingly, most articles leveraged a small number of citations with the majority using 3 or less and very few leveraging more than 6 total references. The mean number of citations per article is only 1.68.

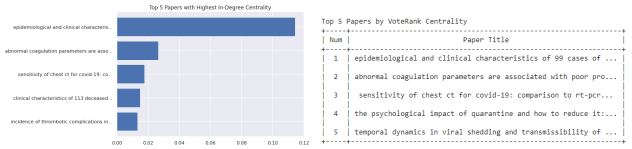

Figure 8: Top 5 Papers with In-Degree Centrality (left) and VoteRank Centrality (right).

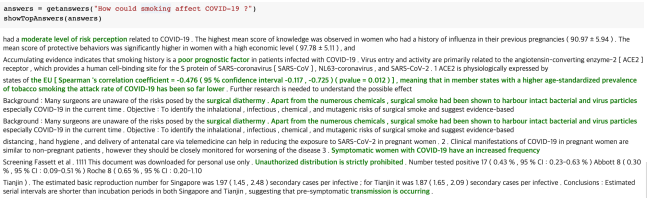

Figure 9: Results of QA response for an example question.

Question Answering. Using proposed method, we made attempts to generate answers for important questions for COVID-19. Figure 9 shows results for one example question. For the question "How could smoking affect COVID-19 ?", the model generates answers like "Accumulating evidence indicates that smoking history is a poor prognostic factor in patients infected with COVID-19 ." We are not bio-medical experts, but we could see that the results make sense.

UI and Visualization. The user interface for the final dashboard was built using Node.js with the React web framework. We were able to successfully visualize the document clusters and the citation network displayed in Figure 10 and Figure 11. In addition to the graph visualizations, users are able to further interact with the dashboard through search capabilities. Tables were created to display the results from the article search and question-answer functionalities.

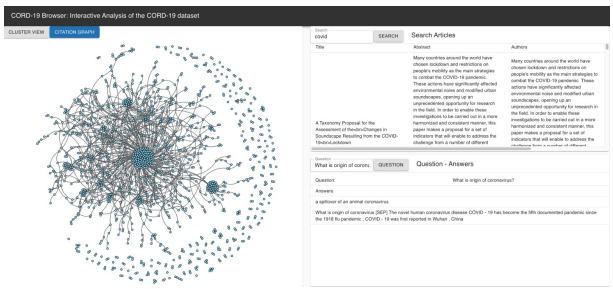

Figure 10: Dashboard Citation Graph View with Article Search QA

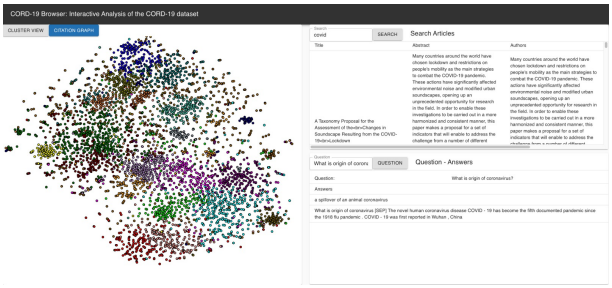

Figure 11: Dashboard Document Clustering View with Article Search QA

7 CONCLUSION

We were able to create and visualize clustering of articles and citation networks as well as implement a question-answer deep learning model. Our results indicate that K-means was able to create cleaner clusters with less overlapping groups than the LDA method and that when comparing centrality and in-degree metrics in the citation network, a handful of papers were heavy referenced by other research articles. An area of improvement is to further explore our clustering approach since our current approach is unable to label clusters that are not tight-knit. The usability report indicates that our pilot product is overall easy to use and informative with good visualizations. However, we can further improve our solution by refining the details for the dashboard to offer better integration of functionalities. This will provide a seamless user experience for those leveraging the product.

REFERENCES

[1] Gayatri Behera. [n. d.]. Synonym or similar word detection in assignment papers. *International Journal of Engineering and Technical Research* 7, 8 ([n. d.]).

[2] Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A Pretrained Language Model for Scientific Text. arXiv:cs.CL/1903.10676

[3] David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research* 3, Jan (2003), 993–1022.

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[5] Nees Jan Van Eck and Ludo Waltman. 2014. CitNetExplorer: A new software tool for analyzing and visualizing citation networks. *Journal of Informetrics* 8, 4 (2014), 802–823. https://doi.org/10.1016/j.joi.2014.07.006

[6] Maksim Ekin Eren, Nick Solovyev, Edward Raff, Charles Nicholas, and Ben Johnson. 2020. COVID-19 Kaggle Literature Organization. *Proceedings of the ACM Symposium on Document Engineering 2020* (Sep 2020). https://doi.org/10.1145/3395027.3419591

[7] Andre Esteva, Anuprit Kale, Romain Paulus, Kazuma Hashimoto, Wenpeng Yin, Dragomir Radev, and Richard Socher. 2020. Co-search: Covid-19 information retrieval with semantic search, question answering, and abstractive summarization. *arXiv preprint arXiv:2006.09595* (2020).

[8] Jennifer Golbeck. [n. d.]. Analyzing the Social Web. https://www.sciencedirect.com/book/9780124055315/analyzing-the-social-web

[9] Aric Hagberg, Dan Schult, Pieter Swart, D Conway, L Séguin-Charbonneau, C Ellison, B Edwards, and J Torrents. 2013. Networkx. High productivity software for complex networks. *Webová strá nka https://networkx. lanl. gov/wiki* (2013).

[10] Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. 2008. Exploring Network Structure, Dynamics, and Function using NetworkX. In *Proceedings of the 7th Python in Science Conference*, Gaël Varoquaux, Travis Vaught, and Jarrod Millman (Eds.). Pasadena, CA USA, 11–15.

[11] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2019. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics* (Sep 2019). https://doi.org/10.1093/bioinformatics/btz682

[12] Siwei Li, Zhiyan Zhou, Anish Upadhayay, Omar Shaikh, Scott Freitas, Haekyu Park, Zijie J Wang, Susanta Routray, Matthew Hull, and Duen Horng Chau. 2020. Argo Lite: Open-Source Interactive Graph Exploration and Visualization in Browsers. *arXiv preprint arXiv:2008.11844* (2020).

[13] Rina Nakazawa, Takayuki Itoh, and Takafumi Saito. 2015. A visualization of research papers based on the topics and citation network. In *2015 19th International Conference on Information Visualisation*. IEEE, 283–289.

[14] Charles Nicholas. 2020. Mr. Shakespeare, Meet Mr. Tucker, Part II. *UMBC Faculty Collection* (2020).

[15] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text. arXiv:cs.CL/1606.05250

[16] Giuseppe Rizzo, Raphaël Troncy, Sebastian Hellmann, and Martin Bruemmer. 2012. NERD meets NIF: Lifting NLP Extraction Results to the Linked Data Cloud. *LDOW* 937 (2012).

[17] Ulrich Schäfer, Hans Uszkoreit, Christian Federmann, Torsten Marek, and Yajing Zhang. 2008. Extracting and Querying Relations in Scientific Papers on Language Technology.. In *LREC*. Citeseer.

[18] E. Segel and J. Heer. 2010. Narrative Visualization: Telling Stories with Data. *IEEE Transactions on Visualization and Computer Graphics* 16, 6 (2010), 1139–1148.

[19] Raphael Tang, Rodrigo Nogueira, Edwin Zhang, Nikhil Gupta, Phuong Cam, Kyunghyun Cho, and Jimmy Lin. 2020. Rapidly Bootstrapping a Question Answering Dataset for COVID-19. arXiv:cs.CL/2004.11339

[20] Laurens van der Maaten and Geoffrey Hinton. 2008. Viualizing data using t-SNE. *Journal of Machine Learning Research* 9 (11

2008), 2579–2605.

[21] Fei Wang, Tao Li, Xin Wang, Shenghuo Zhu, and Chris Ding. 2010. Community discovery using nonnegative matrix factorization. *Data Mining and Knowledge Discovery Volume 22* (July 2010). https://doi.org/10.1007/s10618-010-0181-y

[22] Lucy Lu Wang, Kyle Lo, Yoganand Chandrasekhar, Russell Reas, Jiangjiang Yang, Doug Burdick, Darrin Eide, Kathryn Funk, Yannis Katsis, Rodney Kinney, Yunyao Li, Ziyang Liu, William Merrill, Paul Mooney, Dewey Murdick, Devvret Rishi, Jerry Sheehan, Zhihong Shen, Brandon Stilson, Alex Wade, Kuansan Wang, Nancy Xin Ru Wang, Chris Wilhelm, Boya Xie, Douglas Raymond, Daniel S. Weld, Oren Etzioni, and Sebastian Kohlmeier. 2020. CORD-19: The COVID-19 Open Research Dataset. arXiv:cs.DL/2004.10706

[23] Shenghui Wang and Rob Koopman. 2017. Clustering articles based on semantic similarity. *Scientometrics (2017) 111:1017–1031* (February 2017). https://doi.org/10.1007/s11192-017-2298-x

[24] J. D. West, I. Wesley-Smith, and C. T. Bergstrom. 2016. A Recommendation System Based on Hierarchical Clustering of an Article-Level Citation Network. *IEEE Transactions on Big Data* 2, 2 (2016), 113–123.

[25] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. HuggingFace's Transformers: State-of-the-art Natural Language Processing. *ArXiv* (2019), arXiv–1910.

[26] Jian-Xiong Zhang, Duan-Bing Chen, Qiang Dong, and Zhi-Dan Zhao. 2016. Identifying a set of influential spreaders in complex networks. *Scientific Reports* 6, 1 (2016). https://doi.org/10.1038/srep27823