COVID-19 Research Knowledge Representation

Team Data and Depictions (D&D) Final Report

J. Scott Cardinal
cardinal.js@gatech.edu

Shad Hopson
shopson9@gatech.edu

Keegan Nesbitt
knesbitt7@gatech.edu

Tina Nguyen
tnguyen603@gatech.edu

Erhan Posluk
eposluk3@gatech.edu

Annmarie Thomas*
athomas377@gatech.edu

1 INTRODUCTION

Researchers often face difficulty parsing the large body of publications available to them, particularly those not directly related to their area of research. This can lead to an absence of knowledge of pertinent work that may be relevant[2]. It is essential for researchers, especially medical personnel, to be able to understand the intricate relationships between research publications. Visualization of those relationships can facilitate and expedite discovery and synthesis of related concepts [18]. NLP can facilitate the transformation of textual data into semantic knowledge representation [16, 22, 23], which can greatly expedite such literature review. Although this project is limited in scope to publications related to COVID-19, the larger task of processing text-based data and succinctly categorizing it through visualization can lead to more efficient ways of accessing relevant academic papers to meet the needs of researchers, medical personnel, students, and others.

2 PROBLEM DEFINITION

The aim is to ascertain and visualize a knowledge representation of the semantic conceptual domains of research for a selection of published literature. The target corpus consists of research publications related to COVID-19 [1, 27]. Our tool will allow users to select the underlying NLP techniques, resulting in different graphical representations for the output of each technique. Our method modernizes previous tools such as Jigsaw [11] to identify textual similarities with greater emphasis on semantic context. The success of this project will be in capturing the specific conceptual and semantic domains of article content as a knowledge representation.

3 BACKGROUND SURVEY

When categorizing documents, rarely will all groups will be completely distinct. There is often considerable overlap between their topical content. In Wan [26] subtopics were derived from each document by using approaches such as agglomerative clustering of sentences or TextTiling. Similarities of these topics are then calculated across documents. Fan et al. [10] applied Stanford Core NLP Parser with Word2Vec on image captions to extract semantic concepts and create concept clusters associated with the text and images. This approach precisely identified images content based on the text, but showed the limitations of NLP in capturing the contextual meaning of words.

Bidirectional Encoder Representations from Transformers (BERT) successfully classified sentences from abstracts in a corpus of academic papers about construction [6]. Sentences were separated into one of four ontologies and one network for each ontology was generated. Document and ontology nodes were linked with edges, allowing graph traversal, but reference information for the related papers was not present. Cite2Vec [3] included novelties by expanding the citation-network data to also include sentences from other papers in hich the target paper is cited. NetLens [12] used a content-actor model, where the user can navigate two submodels in parallel based on either the document content or its sources. Cite2Vec utilized an undirected node-graph of keywords and documents while NetLens used a more simplistic visualization of barcharts and flow graphs. Cite2Vec excelled at linking related papers to one another, but was less efficient at creating well defined boundaries of topics. Jigsaw [11] used a hybrid approach combining different forms of visualizations with varying computational ideas for word embeddings.

Numerous methods have been developed to identify key ideas from text-based data, such as the Latent-Dirichlet Allocation (LDA) and nonnegative matrix factorization models [4, 13]. Both result in a probability distribution of topics per document as well as the probability distribution of words [7]. BERT is a bidirectional neural network-based NLP pre-training technique [9]. It is an improvement over Word2Vec in that context is preserved in the final embedding, and improves upon previous unidirectional language representation models with bidirectionality allowing for a more complete understanding of context sensitive features. Like BERT, a graph representation of each document can be formed as described in Schenker et al. [24], in which words are represented as nodes and edges indicate their positional relationship in the document. Similarity is then calculated with graph distance measures.

Another challenge in unsupervised clustering methods is that some domain knowledge is typically required for determining how many clusters should be present. If an algorithm such as DBSCAN [15] is utilized, however, the number of clusters need not be known. Instead, a distance function is utilized to compare points that are organized into neighborhoods of a configured density or above.

4 PROPOSED METHODS

The primary innovations of our application include:

1. focus on paper contents, not citations,
2. quantifying the effectiveness of clustering for end users' exploration, and
3. relating papers to topics, then topics to each other, rather than trying to directly compare papers.

Rather than focusing on the citations like prior approaches, we focus on the contents of each paper's abstract utilizing a multi-step NLP approach to perform analysis on approximately 400,000 research papers on COVID-19 [1]. The precomputed results will then be rendered to the end user, allowing them to select between the options available and drill down from a high-level overview of topics into the individual papers present in the corpus.

All team members contributed equal efforts to the final project.

- Keegan Nesbitt: Setup of compute/storage infrastructure, graph representation, research UI implementation, and creation of demo code and instructions.
- Erhan Posluk: NLP pre-processing, scaling of preprocessing with Spark/AWS, and demo code instructions.
- Annmarie Thomas: UI design and implementation, coding and testing
- Tina Nguyen: UI design and implementation, coding and testing
- James (Scott) Cardinal: Network analysis, similarity and adjacency measures, topological overlap, graph construction and analysis.
- Shad Hopson: Research and implementation of BERT and LDA topic extraction, and demo code instructions.

4.1 NLP Pre-processing

The data for this project was obtained from the Kaggle website consisting of a csv file, where each row is a unique paper and the columns are features of interest (paper id, title, authors, abstract text, etc.). While the data was plentiful, it was not in a usable state for use with natural language processing. Steps taken to transform the raw abstract data into a usable structured dataset, along with tools of implementation within PySpark are listed as follows:

1. Eliminate bad data and duplicate entries (standard library functions).
2. Convert letters to lowercase (standard library functions).
3. Expand contractions (user-defined-function with *contractions* library).
4. Replace non-alpha characters with spaces and remove single character words (reg expressions).
5. Tokenize long strings of texts into lists of words (user defined function with *NLTK* library).
6. Lemmatize all tokens to the respective base terms (user defined function with *NLTK Wordnet*).
7. Compare lemmatized tokens to list of stop words and filter accordingly.
8. Write pre-processed data to readily accessible files (parquet format).

The decision was made to use lemmatization instead of stemming because it produces more interpretable results for topical modeling. Lemmatization converts a

word to a meaningful base term, whereas stemming cuts off the last few characters producing a common root (e.g., “loves” transforms to “love” with lemmatization and “lov” with stemming). Lemmatization was completed on the list of words for each paper using Wordnet lemmatization technique from the NLTK Python package. A few additions were made to the standard stopwords list for our dataset. Words like abstract, introduction, figure, and table references were common, but not valuable to our modeling. The final step was to store this data in a readily accessible fashion – an SQL database for the demo code, and Parquet files when implemented at scale in AWS EMR. An example of the pre-processing (before and after) can be seen in the Appendix.

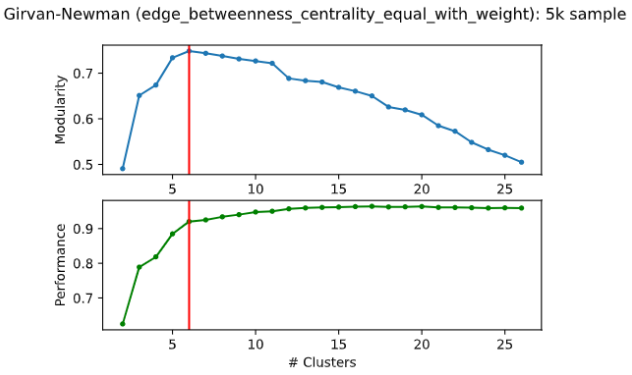

Figure 1: Topic cluster modularity and performance of Girvan-Newman communities from BERTopic graph.

4.2 Topic Extraction & Modelling

Two methods were used for topic extraction. First, we used the library BERTopic, which uses BERT to find word embeddings in a document. BERTopic clusters documents using UMAP [17] for initial dimensionality reduction, then HDBSCAN [5] to cluster along the reduced embedding map, to identify high-probability topics within the global lexicon of the corpus. Cluster distance is measured by TF-IDF. In the second method, we vectorize the abstracts with TF-IDF, cluster the data using a Gaussian mixture model, and then use LDA to find the topics of each cluster.

To select the optimal number of topics, we vary the number of topics when running the LDA code. We can compare these sets of topics quantitatively via the coherence measure [25]. Qualitative examination will occur by reviewing the words present in each topic to see how distinct each topic is and whether we believe they would aid a research in identifying relevant research.

Our approach to modelling the extracted topics is essentially one of biclustering. Topics retrieved during the extraction are clustered by co-expression in documents across the corpus, while documents are clustered by the distribution of their probability of association with topic vectors. Using correlation as a similarity measure derived from document probabilities of extracted topics, the adjacency between topics or papers was calculated either by signum function or exponentially damped weight. From that adjacency, we calculated the network’s topological overlap [14, 28], with a quantile threshold cutoff for the final graph adjacency matrix. From that topological overlap matrix (TOM), we form an undirected acyclic graph of all topics identified from the papers utilizing Python’s NetworkX package. Each node in the graph represents a single topic, while the edges represent the similarity measure between topics.

With the subsequent graph, we utilize the Girvan-Newman algorithm [19, 20] to identify network communities indicating clusters of topics. This algorithm utilizes the TOM edge weights to group the paper topics into different clusters by identifying edges with the highest betweenness. At each iteration, an edge is removed to separate communities of nodes and the remaining edge with highest betweenness is recalculated. The iterative approach of removing edges to form clustered communities results in a hierarchical clustering that could be visualized as a dendrogram. Output of the clustering assigns each topic to a given cluster. As the number of clusters can vary depending upon how many iterations are executed, a continuous range of cluster counts are produced.

To assist selection of the number of clusters, we calculate the effectiveness of the clustering using NetworkX. Specifically, we utilize the modularity metric [8, 21] to identify community structure within the topic network. A clustering with many within-cluster connections but few connections with different clusters will have a high modularity value. When viewing the results, the user can vary the number of topic clusters he or she wants to visualize. Since the modularity value is specific to the number of clusters, the value is displayed to the end

Figure 2: Team DnD COVID-19 topic modelling user interface.

user as additional information and used to recommend a default clustering.

We produce the graph clustering output twice, once with BERTopic output and once with LDA. For each combination of topic modeling and topic similarity approach, the graph will be constructed and clusters formed. Selection of these approaches gives the user more flexibility to select the option that seems to perform best for his or her use case while allowing quick interaction, since these options are all precomputed. However, our end user is not expected to be knowledgeable in the field of topic modeling, similarity metrics, or graph clustering, so the options we present to the user for selection are adapted to plain-English descriptions instead. The cluster count selection is intuitive, so it will remain unmodified.

4.3 UI Design

The UI is implemented with Bokeh on a data backend of MySQL. Bokeh was selected to allow for integration with other Python libraries such as Pandas or NetworkX. MySQL was chosen as the backend as it would be appropriate for a larger scale implementation of a hosted application. In our prototype, we opted to run the user interface and the MySQL server locally, employing Docker containers for a consistent development environment amongst team members.

The interface starts with a network graph of topics. Once a user clicks on a topic node, this displays a table with the papers associated with that topic. From there, a user can decide to explore their topic of interest further with bar charts, which can be filtered based on topic choice. One bar chart displays the top ten words and their probability of association with that topic. Another chart displays the top 5 papers associated with the selected topic, with each paper's most highly related topics and probability of association. This feature helps the user see connections between topics, papers, and other topics more clearly rather than being overwhelmed by the entire network. All graphics have hover tools that display related information and metrics about the feature of interest.

5 EXPERIMENTS & EVALUATION

This application was aimed at answering two main questions through the various coding scripts and the design of an interactive Bokeh tool. At a high level, some of these questions included:

1. Comparison of the results from BERT vs LDA techniques (see Appendix for one example).
2. Sensitivity to varying numbers of clusters as evaluated by the Girvan-Newman algorithm.

Initially, we tested the NLP pre-processing on a sample dataset of approximately 5,000 papers locally using regular Python and NLTK. However, this was not effective for the full set of 400,000 documents. To handle the full-scale data, we needed to utilize a distributed solution involving AWS EMR and Apache Spark. After converting the NLTK code to work in Spark, we were able to process the full dataset on an EMR cluster consisting of five *m5.xlarge* nodes in approximately 1 hour and 40 minutes. During this processing, we eliminated approximately 100,000 documents of the original 400,000 due to bad data, duplication, or other errors. Experiments for this step were completed in a few iterations, progressively adding different project-specific stopwords and comparing resulting tokens of stemming vs lemmatization. Upon completion, we successfully produced the tokenized output for ~ 300,000 candidate papers to be passed to BERTopic and LDA.

BERTopic and LDA underwent a similar scaling solution, as a local implementation was not possible. Here, we utilized AWS EC2 instead of AWS EMR since BERTopic and LDA libraries were not implemented for distributed

processing. We used the largest EC2 instance type available in the student accounts, *m5.2xlarge*, and attempted to run BERTopic on the full dataset. After 30 hours, the execution was still incomplete. From the CPU and memory usage metrics observed in the AWS EC2 console, however, we did not seem to be bound by resource constraints. We were able to successfully execute using approximately 100,000 papers, therefore we proceeded to use this collection in future steps.

With LDA, we also ran into scaling difficulties when trying to go beyond 100,000 papers. In this case, we seemed to be limited by memory constraints when utilizing PCA on the data prior to the LDA modeling. To maintain parity with the BERTopic output, we produced the LDA output for the same 100,000 papers to enable the user of our application to explore and compare both algorithms' results in the user interface. With BERTopic, the run time appears to increase exponentially with the number of papers – ~ 5,000 documents taking a few minutes, ~ 100,000 taking a few hours, and ~ 300,000 documents running for over 27 hours without finishing.

Using the set of 100,000 papers, we were able to effectively experiment with the number of topics produced by LDA and see which was most effective. The BERTopic output was generating around 800 topics, so we tried some similar values and others for greater exploration. Unfortunately, we learned we needed to use the *gensim* package's implementation of LDA rather than *scikit-learn*'s to obtain the coherence measure. Due to the time remaining combined with the additional runtime this calculation would add, we were unable to reimplement our approach using the *gensim* package (rough estimates from online forums stated gensim can take 9 times as long to run).

With BERTopic, some of this metric exploration is abstracted and handled for us. It utilizes a similar metric to coherence called class-based cTF-IDF, which extracts the most important words per cluster, to help select the number of topics identified. It also automatically reduces topics by merging those over a certain similarity threshold.

With a visual check of the created topics of BERTopic and the LDA model, we found that BERTopic had much clearer and more specific topics than the LDA model. This remained true even as we increased the number of topics in the LDA model. A visual inspection of the BERTopics output showed broad categorization into computation, education, vaccinations, many groups related to specific countries, and documents related to other types of coronavirus including those found in animals. This precise clustering led to some topic groups being as small as 10 documents, while those related to education were in the thousands. In general, visual inspection shows BERTopic doing very well. The table below shows examples of the topic-words both models have for their topic-groups.

After topic extraction, each document had a probability of association with each topic. Topic extraction determines the most disjoint set of keywords or phrases from the total document corpus and assigns topics to individual documents accordingly. In many cases, documents were strongly associated with one particular topic with much smaller probabilities for all others. Many documents, however, do not fall cleanly into only one topic. In our sample corpus of 5,253 journal articles, only 56% (n=2,949) of the documents could be definitively assigned a topic. The remaining 44% of the documents were indeterminate, with multiple feasible topic probabilities.

The goal of our secondary topic clustering was to find the associations between topics by exploring their co-expression within subsets of documents. Similarly, a large number of those indeterminate documents could be cross associated by identifying correlations in their topic probability profiles. We found that only a small portion of the ~ 5k documents sampled (less than 4%) could not be associated with a topic cluster at all (i.e., outliers). Conversely, we found that although some 80 disjoint topics had been extracted, those topics were strongly organized into only six discernable communities with no network isolates.

Unfortunately, we found that community detection algorithms were not computationally feasible for the much larger and highly-connected graph of documents even with the limited test sample. This may be an inherent limitation of the NetworkX implementation rather than of the method itself, however. Traditional clustering using the complement of the graph's adjacency matrix proved very feasible, identifying ~ 26 sub-communities of documents within the 5k test sample. The solution was found too late in the process to reasonably incorporate into the final project.

The user interface (UI) was presented to four participants not involved with the project's development.

Table 1: Comparison of BERTopic and LDA topics extracted.

Topic #	LDA	BERT
1	health, covid, disease, research, study	teacher, learning, teach, teaching
2	student, online, learn, medical, covid	mrna, replication, genome, ribosome
3	immune, ifn, virus, covid	graph, algorithm, neural, semantic
4	group, laparoscopic,surgery, study	vaccination, vaccinate, immunization, antigen

Backgrounds of these participants were varied, including a physics professor, biology professor, nurse with research background, and historian. These users were asked to explore the UI and provide feedback around the following questions:

1. How intuitive is the interface to use?
2. How does the ability to search for papers with this interface compare to traditional searches? What are perceived benefits and drawbacks?
3. Is it helpful to use the different topic modeling versions (BERT vs LDA interfaces)?
4. Suggestions for additional features.

All participants stated that the interface's features appeared intuitive, though on further use not all features worked quite as expected and the availability of some tools were unclear (e.g, sorting the table by column, or the highlighting tool to zoom in on charts). Two participants suggested including descriptors of what tools were available to explore various plots.

Participants thought the topic modeling was interesting, but three stated the network graph would not necessarily make it easier to find papers that met their desired search criteria. The fourth user felt the hover tool with words associated to topics made it quick and easy to scan for topic clusters of interest. One user stated, "the network graph is useful if your goal is general exploration, but most researchers are looking for very specific topics and the network does not allow for easy identification of papers aligned to a search."

Users generally agreed the ability to perform a keyword search and have the interface return topics only aligned with those keywords would greatly improve the interface. The LDA component was generally viewed as not useful due to topic clusters being too broad, but that, on the BERT model, once a topic of interest was located, the ability to quickly scan related topics for relevance was perceived as an improvement over traditional search results.

6 DISCUSSION

Overall, we faced the most challenges in getting our topic modeling approaches to effectively scale up to the full data sets. Due to the time spent here, we were unable to do much parameter tuning and exploration with different settings of BERTopic or LDA. In future work, exploring these parameters in greater depth and with more scalable algorithms could yield better results. Another area for improvement is trying to better distribute other portions of the logic. For example, with the LDA code, there are implementations of PCA, Gaussian mixture models, and LDA natively in PySpark. This may allow for scaling of our solution to the full set of 300,000 preprocessed papers.

Applying a secondary clustering following topic extraction to detect communities within the topic graph appears to have been successful but did not adequately scale for the larger graph of documents. The results, however, are promising for the general approach. Further work might leverage more computationally efficient methods for community detection or alternative clustering methods. It's clear from the overall outcomes of this exploratory work that it is at least theoretically feasible to address topic modelling and document search through connected communities and evaluating document co-expression networks.

Finally, the design of the user interface as a tool to navigate topic modeling has some distinct advantages over traditional search engines for academic papers, while still having areas where it can be improved for ease of use. The visualization of connected topics helps the user see relationships between topics as intended. The ability to have the most relevant terms populate for each topic cluster allows the user to quickly assess a cluster's relevance to their search. Further development of a user interface should focus on deeper inquiries into a topic cluster's features as well as the ability to search by keyword for desired content.

REFERENCES

[1] Allen Institute. 2020. COVID-19 Open Research Dataset Challenge(CORD-19). https://www.kaggle.com/allen-institute-for-ai/CORD-19-research-challenge

[2] Gábor Berend and Richárd Farkas. 2013. Keyphrase-Driven Document Visualization Tool. In *The Companion Volume of the Proceedings of IJCNLP 2013*, Kentaro Torisawa and Hang Li (Eds.). Asian Federation of Natural Language Processing, Nagoya, Japan, 17–20. https://www.aclweb.org/anthology/I13-2005

[3] Matthew Berger, Katherine McDonough, and Lee M. Seversky. 2017. cite2vec: Citation-Driven Document Exploration via Word Embeddings. *IEEE Transactions on Visualization and Computer Graphics* 23, 1 (Jan. 2017), 691–700. https://doi.org/10.1109/tvcg.2016.2598667

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

[5] Ricardo J. G. B. Campello, Davoud Moulavi, Arthur Zimek, and Jörg Sander. 2015. Hierarchical Density Estimates for Data Clustering, Visualization, and Outlier Detection. *ACM Transactions on Knowledge Discovery from Data* 10, 1 (July 2015), 1–51. https://doi.org/10.1145/2733381

[6] Hainan Chen and Xiaowei Luo. 2019. An automatic literature knowledge graph and reasoning network modeling framework based on ontology and natural language processing. *Advanced Engineering Informatics* 42 (Oct. 2019), 100959. https://doi.org/10.1016/j.aei.2019.100959

[7] Jaegul Choo, Changhyun Lee, Chandan K. Reddy, and Haesun Park. 2013. UTOPIAN: User-Driven Topic Modeling Based on Interactive Nonnegative Matrix Factorization. *IEEE Transactions on Visualization and Computer Graphics* 19, 12 (Dec. 2013), 1992–2001. https://doi.org/10.1109/tvcg.2013.212

[8] Aaron Clauset, M. E.J. Newman, and Cristopher Moore. 2004. Finding community structure in very large networks. *Physical Review E - Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics* 70, 6 (2004), 6. https://doi.org/10.1103/PhysRevE.70.066111 arXiv:cond-mat/0408187

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Mlm (Oct. 2018). arXiv:1810.04805

[10] Mengdi Fan, Wenmin Wang, Peilei Dong, Ronggang Wang, and Ge Li. 2018. Unsupervised Concept Learning in Text Subspace for Cross-Media Retrieval. In *Advances in Multimedia Information Processing – PCM 2017*. Springer International Publishing, 505–514. https://doi.org/10.1007/978-3-319-77380-3_48

[11] C. Gorg, Zhicheng Liu, Jaeyeon Kihm, Jaegul Choo, Haesun Park, and J. Stasko. 2013. Combining Computational Analyses and Interactive Visualization for Document Exploration and Sensemaking in Jigsaw. *IEEE Transactions on Visualization and Computer Graphics* 19, 10 (Oct. 2013), 1646–1663. https://doi.org/10.1109/tvcg.2012.324

[12] Hyunmo Kang, Catherine Plaisant, Bongshin Lee, and Benjamin B. Bederson. 2007. NetLens: Iterative Exploration of Content-Actor Network Data. *Information Visualization* 6, 1 (Jan. 2007), 18–31. https://doi.org/10.1057/palgrave.ivs.9500143

[13] Da Kuang, Jaegul Choo, and Haesun Park. 2014. Nonnegative Matrix Factorization for Interactive Topic Modeling and Document Clustering. In *Partitional Clustering Algorithms*. Springer International Publishing, 215–243. https://doi.org/10.1007/978-3-319-09259-1_7

[14] Ai Li and Steve Horvath. 2007. Network neighborhood analysis with the multi-node topological overlap measure. *Bioinformatics* 23, 2 (2007), 222–231. https://doi.org/10.1093/bioinformatics/btl581

[15] Soumaya Louhichi, Mariem Gzara, and Hanene Ben Abdallah. 2014. A density based algorithm for discovering clusters with varied density. In *2014 World Congress on Computer Applications and Information Systems (WCCAIS)*. IEEE. https://doi.org/10.1109/wccais.2014.6916622

[16] Souheyl Mallat, Emna Hkiri, Mohsen Maraoui, and Mounir Zrigui. 2015. Semantic Network Formalism for Knowledge Representation: Towards Consideration of Contextual Information. *International Journal on Semantic Web and Information Systems* 11, 4 (2015), 64–85. https://doi.org/10.4018/IJSWIS.2015100103

[17] Leland McInnes, John Healy, Nathaniel Saul, and Lukas Großberger. 2018. UMAP: Uniform Manifold Approximation and Projection. *Journal of Open Source Software* 3, 29 (Sept. 2018), 861. https://doi.org/10.21105/joss.00861 arXiv:1802.03426

[18] Rina Nakazawa, Takayuki Itoh, and Takafumi Saito. 2018. Analytics and visualization of citation network applying topic-based clustering. *Journal of Visualization* 21, 4 (March 2018), 681–693. https://doi.org/10.1007/s12650-018-0483-5

[19] M.E.J. Newman and M. Girvan. 2003. Mixing Patterns and Community Structure in Networks. In *Statistical Mechanics of Complex Networks*, Romualdo Pastor-Satorras, Miguel Rubi, and Albert Diaz-Guilera (Eds.). Springer, Chapter 5, 66–87. https://doi.org/10.1007/978-3-540-44943-0_5 arXiv:cond-mat/0210146

[20] M. E. J. Newman. 2004. Fast algorithm for detecting community structure in networks. *Physical Review E* 69, 6 (June 2004), 066133. https://doi.org/10.1103/PhysRevE.69.066133 arXiv:cond-mat/0309508

[21] M. E. J. Newman. 2006. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences* 103, 23 (June 2006), 8577–8582. https://doi.org/10.1073/pnas.0601602103 arXiv:physics/0602124

[22] Judea Pearl. 1986. Fusion, propagation, and structuring in belief networks. *Artificial Intelligence* 29, 3 (1986), 241–288. https://doi.org/10.1016/0004-3702(86)90072-X

[23] Walid S. Saba. 2007. Language, logic and ontology: Uncovering the structure of commonsense knowledge. *International Journal of Human Computer Studies* 65, 7 (2007), 610–623. https://doi.org/10.1016/j.ijhcs.2007.02.002 arXiv:cs/0610067

[24] Adam Schenker, Mark Last, Horst Bunke, and Abraham Kandel. 2003. Comparison of Distance Measures for Graph-Based Clustering of Documents. In *Graph Based Representations*

in Pattern Recognition. Springer Berlin Heidelberg, 202–213. https://doi.org/10.1007/3-540-45028-9_18

[25] Hanna M. Wallach, Iain Murray, Ruslan Salakhutdinov, and David Mimno. 2009. Evaluation methods for topic models. In *Proceedings of the 26th Annual International Conference on Machine Learning - ICML '09*, Vol. 167. ACM Press, Montreal, Quebec, Canada, 1–8. https://doi.org/10.1145/1553374.1553515

[26] Xiaojun Wan. 2007. A novel document similarity measure based on earth mover's distance. *Information Sciences* 177, 18 (Sept. 2007), 3718–3730. https://doi.org/10.1016/j.ins.2007.02.045

[27] 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* (April 2020). arXiv:2004.10706

[28] Andy M Yip and Steve Horvath. 2007. Gene network interconnectedness and the generalized topological overlap measure. *BMC bioinformatics* 8 (2007), 22. https://doi.org/10.1186/1471-2105-8-22

A EXAMPLE OF TOPIC EXTRACTION

Shown below (Figure 3) is an example of the progression of an abstract through the DnD text processing procedure. These same results can be seen in the various outputs along each step (see Readme in folder download). An example with a short abstract was selected here. One interesting conclusion that might be drawn from this data is that BERT appears to provide much more specific topics than that of LDA. The LDA and BERT scales differ in their respective packages, but in both cases a higher value indicates a stronger word association.

Figure 3: Example output for Document ID `e1xm39x8`, showing unprocessed paper abstract, lemmatized text, and topics extracted by LDA and BERTopic.

Paper ID: e1xm39x8
Title: The Strategy of Picornavirus Evading Host Antiviral Responses: Non-structural Proteins Suppress the Production of IFNs
Authors: Wang, Yining; Ma, Lina; Stipkovits, Laszlo; Szathmary, Susan; Li, Xuerui; Liu, Yongsheng

Raw Unprocessed Text

Viral infections trigger the innate immune system to produce interferons (IFNs), which play important role in host antiviral responses. Co-evolution of viruses with their hosts has favored development of various strategies to evade the effects of IFNs, enabling viruses to survive inside host cells. One such strategy involves inhibition of IFN signaling pathways by non-structural proteins. In this review, we provide a brief overview of host signaling pathways inducing IFN production and their suppression by picornavirus non-structural proteins. Using this strategy, picornaviruses can evade the host immune response and replicate inside host cells.

Pre-processed Text (Tokenized, Lemmatized, etc.)

['viral', 'infection', 'trigger', 'innate', 'immune', 'system', 'produce', 'interferon', 'ifns', 'play', 'important', 'role', 'host', 'antiviral', 'response', 'co', 'evolution', 'virus', 'host', 'favor', 'development', 'various', 'strategy', 'evade', 'effect', 'ifns', 'enable', 'virus', 'survive', 'inside', 'host', 'cell', 'one', 'strategy', 'involve', 'inhibition', 'ifn', 'signal', 'pathway', 'non', 'structural', 'protein', 'review', 'provide', 'brief', 'overview', 'host', 'signaling', 'pathway', 'induce', 'ifn', 'production', 'suppression', 'picornavirus', 'non', 'structural', 'protein', 'use', 'strategy', 'picornaviruses' 'evade', 'host', 'immune', 'response', 'replicate', 'inside', 'host', 'cell']

LDA Output

Topic 18
Top 10 Word Associations
('virus', 76.53662)
('response', 75.55084)
('infection', 25.18072)
('hcv', 24.66671)
('cell', 15.89236)
('mouse', 12.39168)
('ifns', 9.92372)
('produce', 7.54982)
('treatment', 6.87384)
('type', 6.32834)

BERT Output

Topic 69
Top 10 Word Associations
('cmv', 0.15534)
('immunocompetent', 0.03387)
('sepsis', 0.02854)
('patient', 0.02569)
('cytomegalovirus', 0.01759)
('immunocompromised', 0.01513)
('fda', 0.01419)
('lymphopenia', 0.01382)
('haplotype', 0.01379)
('infection', 0.01371)