2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)

# A Text Mining Approach to Discovering COVID-19 Relevant Factors

Javier Sastre
Analytics & AI
Accenture
Dublin, Ireland
j.sastre.martinez@accenture.com

Ali Hosseinzadeh Vahid
Analytics & AI
Accenture
Dublin, Ireland
ali.vahid @accenture.com

Caitlin McDonagh
Analytics & AI
Accenture
Dublin, Ireland
caitlin.mc.donagh@accenture.com

Paul Walsh
Analytics & AI
Accenture
Dublin, Ireland
paul.a.walsh@accenture.com

2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM) 978-1-7281-6215-7/20/$31.00 ©2020 IEEE DOI: 10.1109/BIBM49941.2020.9313149

*Abstract*— **This paper describes a text mining approach that utilises the PyLucene search engine and the GrapeNLP grammar engine for extracting links between temperature, humidity and the spread of COVID-19, from a vast collection of scientific publications. The approach was developed in response to a Kaggle challenge from a consortium of research groups to develop text and data mining techniques that can assist the medical community in finding answers to a series of important questions on COVID-19. For this challenge, a large corpus of scientific publications known as the COVID-19 Open Research Dataset (CORD-19) was provided by the consortium. The approach presented in this paper was winner of the competition task of extracting key insights and building summary tables of COVID-19 relevant factors such as temperature and humidity.**

***Keywords—Lucene, GrapeNLP, CORD-19, COVID-19***

## I. Introduction

The outbreak of the COVID-19 pandemic has mobilized scientists from various fields around the world to embark on research to find ways to eliminate the coronavirus and its effects. Results from this research are been published in a vast and growing number of scientific articles, making it difficult for stakeholders to keep abreast of the latest results. This brings about a growing urgency to apply recent advances in AI, such as text mining and natural language processing, to support researchers in the extraction of new insights in the ongoing fight against this infectious disease.

With this in mind, the Allen Institute for AI, in partnership with other organizations, prepared and released the COVID-19 Open Research Dataset (CORD-19) [1], which consists of over 192,000 scholarly articles about COVID-19, SARS-CoV-2, and related coronaviruses. This coalition of leading research groups announced a new challenge[1] through Kaggle and issued a call to action to the world's artificial intelligence experts to develop tools and approaches that can help the medical community develop relevant answers to high priority scientific questions. These questions were drawn from NASEM's Standing Committee on Emerging Infectious Diseases, 21st Century Health Threats[2] research topics and WHO's R&D Blueprint for COVID-19[3], and were organized in 17 Kaggle sub-tasks. More than 500 teams participated in Round 1 of the competition and feedback from medical experts identified that the most useful contributions took the form of article summary tables, so a unique tabular schema was defined for each question for Round 2, the final phase of the competition. This final phase subsequently focused on creating a set of summary tables from the literature and resulted in 100 additional submissions.

To respond to this challenge, we proposed a combined application of the PyLucene search engine [4] and the GrapeNLP grammar engine[5] for the search and extraction of summary table data and tested it on the particular question of whether temperature or humidity are relevant factors to the spread of COVID-19 or not. The system is flexible and configurable and can be further extended to a wider range of research questions. The pipeline has been assessed as the best approach for determining COVID-19 relevant factors as judged by the CORD-19 Research Dataset Challenge evaluation committee, winning the corresponding sub-task in Round 2 of the competition.

In the following sections, we describe the details of the proposed pipeline and will present sample summary tables developed by this pipeline during the CORD-19 Challenge. Our pipeline code is open sourced to help future studies.[6]

## II. Dataset

CORD-19 [1] is a large and growing collection of papers, publications and preprints on COVID-19 and related historical coronaviruses that have been gathered from multiple sources, namely PubMed Central (PMC), PubMed, the WHO's COVID-19 Database, and the preprint servers bioRxiv, medRxiv and arXiv. Some publications are in PDF format while others are in JATS XML. These publications were ingested through the Semantic Scholar literature search engine and converted into the S2ORC JSON lossy format [2], a schema designed to preserve most relevant paper structures such as paragraph breaks, section headers, inline references, and citations. The summary of the last released version of the dataset in the time of the writing this paper is: 192,509 total metadata rows with 84,426 PDF-JSON and 62,736 XML-JSON full text.

[1] https://www.kaggle.com/allen-institute-for-ai/CORD-19-research-challenge

[2] https://www.nationalacademies.org/

[3] https://www.who.int/teams/blueprint/covid-19

[4] https://lucene.apache.org/pylucene/

[5] https://github.com/GrapeNLP

[6] https://www.kaggle.com/javiersastre/covid-19-temperature-and-humidity-summary-tables

## III. Related works

Several research efforts for the extraction and organization of knowledge on COVID-19 have been initiated by the release of the CORD-19 dataset. NEURAL COVIDEX [3] uses a T5-base model [4], fine-tuned on biomedical text to perform unsupervised reranking on documents retrieved via BM25. KDCOVID [5] uses similarity measures based on BioSentVec [6] in order to identify sentences relevant to a query. Sarti [7] uses sentence embeddings for retrieval in a similar manner to KD-COVID, but their embeddings are generated from BERT [8] models trained on NLI datasets. Amazon, Google and Salesforce have launched publicly available IR systems for exploring the CORD-19 dataset. All these systems take as input a query in the form of natural language and return a list of documents from the CORD-19 dataset, ranked by their relevance to the given query. Amazon's system [9] first uses Amazon Comprehend Medical in order to enrich the original dataset. It then utilises Amazon Kendra's intelligent search service for indexing and searching. Google's COVID Research Explorer [10] is based on a semantic search mechanism powered by BERT [8], which was fine-tuned for the medical domain using biomedical IR datasets from the BioASQ challenges. They further enhance their system by combining term and neural-based retrieval models by balancing memorization and generalization dynamics [11]. Saleforce's CO-Search is a retriever-ranker semantic search engine designed to handle complex queries over the COVID-19 literature [12]. The retriever is built from a Siamese-BERT [13] encoder that is linearly composed with a TF-IDF vectorizer [14], and reciprocal-rank fused [15] with a BM25 vectorizer. The ranker is composed of a multi-hop question-answering module [16] that together with a multi-paragraph abstractive summarizer adjusts retriever scores.

Although, there has been a significant effort to extract entities from papers in CORD-19, such as [17] [18], less attention has been placed on extracting relations between entities. Another study [19] applies entity co-occurrences in CORD-19 to construct a graph enabling centrality-based ranking of entities (i.e. drugs, pathogens and biomolecules). CORD-19-SeVeN [20] developed pre-trained relation embeddings (nearly 100k) trained using the SeVeN pipeline [21] on CORD-19. CovEx [22] used concept extraction, knowledge graphs, and user-controlled recommendation to assist users with various levels of domain expertise in their information needs.

During the CORD-19 Research Challenge in Kaggle, more than 500 further approaches were investigated for extracting relevant information about the impact of different factors on the COVID-19 outbreak. Some solutions use information extraction techniques to surface relevant text snippets from papers. Others integrate aspects of text generation and summarization in an effort to improve the interpretability of results. A solution that won most of the challenge subtasks consisted of two main parts: a search index based on sentence embeddings and a custom BERT QA model to extract column-based answers [23]. However, grammar-based approaches also proved useful by winning two of the Kaggle CORD-19 Research Challenge subtasks.[7] This approach has the added benefit of clearer explainability and control of the results. It also has advantages in scenarios where there is a lack of annotated data for machine-learning-based information extraction [24] [25]. The approach is based on local grammars [26], which have already been used for natural language understanding in conversational agents [27] and for extracting relations between entities, especially for languages other than English [28] [29].

## IV. Architecture / methodology

The problem firstly consists of finding papers that explicitly claim or deny correlations between temperature or humidity and the spread of COVID-19. A summary table is then built with the list of extracted papers, the factor (e.g., temperature), whether a claim or denial was found, and a text excerpt for quick review. Fig. 1 illustrates the architecture of our pipeline, which consists of two core components: a document retrieval component and a grammar-based information extraction component.

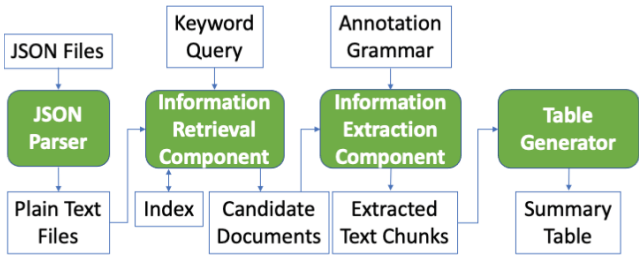


*Fig. 1 Proposed architecture pipeline*

### A. Information Retrieval Component

We used the PyLucene [30] indexing and search library, a Python extension for accessing Apache Lucene [31], to perform a first keyword-based search of papers mentioning COVID-19 as well as mandatory words in the grammar. Due to the use of an inverted index structure, this search is faster than applying a grammar, although it is limited to searching for words without structure. Apache Lucene is well known, and documentation can be found at [31].

### B. Information Extraction Component

We developed a local grammar [26] for finding claims related to temperature or humidity and the spread of COVID-19 (Figs. 4 to 7), using the Unitex [32] grammar editor. We then used the GrapeNLP grammar engine[8] to apply the grammar to the pre-selected papers by the information retrieval component.

Instead of specific words, one can use lexical masks in the grammar to match any token that complies with a set of lexical, syntactic and/or semantic restrictions. A special lexical mask <TOKEN> allows for matching any token. By combining these masks, we have developed a "fuzzy" grammar that partially describes the expressions to match. To solve ambiguity, GrapeNLP implements a scoring system of token matches where each matched token adds a greater score to the overall match as more information is used by the lexical mask that matched it, ranging from 0 for the <TOKEN> mask to 14 for specific words in the grammar. Additionally, the grammar can specify XML tags in order to annotate portions

[7] The one presented in this paper along with the one presented at https://www.kaggle.com/bobirakova/grammar-based-cord-19-approach

[8] GrapeNLP is the rebranded grammar engine used in [27] after its publication at https://github.com/GrapeNLP with additional improvements

of text that are to be extracted (e.g., the factor). GrapeNLP builds an efficient representation of every possible output (tags and scores) as a *filtered-popping network* [33], which is a kind of weighted finite-state automaton with output, then uses a Viterbi-like algorithm (see chapter 18 of [34]) to efficiently extract the top-scoring match.

Local grammars are equivalent to recursive transition networks (RTNs) with output [35]. Unitex represents these grammars as a type of *graph* for improved readability. Sequences recognized by the grammar are those that can be read from left to right by following the grammar paths. Boxes represent components that can recognize any of the sequences listed inside. Text in bold below a box represents the output. Grey boxes represent calls to other sub-grammars, whereas <E> represents the empty string and makes all the content of a box optional. Triangular boxes do not consume input and red text outside the boxes are comments and have no effect on the grammar application.

## V. Use- cases/ empirical experiments

The following subsections explain how we configured PyLucene to index and pre-select the papers. Then, the grammar and the corresponding scoring approach will be described. Finally, samples of summary tables generated by the proposed pipeline are presented.

### A. Configuration of Information Retrieval Component

First of all, the JSON format of the papers provided in the dataset were parsed and converted into a table containing the metadata and full text of the papers. Lucene filters were used to convert the tokens to lower case and to remove English stop words. To search for publications that contain mentions of the target factors as well as COVID-19, we used the Boolean query in Fig. 2. The query matches any document that contains at least one synonym of either temperature or humidity, and at least one synonym of COVID-19.

```
(air OR clammy OR climate OR cool OR cold OR hotne
ss OR humid OR humidity OR precipitation OR rainfall
OR temperature OR temperatures OR warm) AND ("cor
onavirus disease 19" OR "sars cov 2" OR "2019 nco" O
R 2019ncov OR "coronavirus 2019" OR "wuhan pneum
onia" OR  "wuhan virus" OR "wuhan coronavirus" OR
covid19 OR covid-19)
```

*Fig. 2 The query used for searching documents about the effects of temperature and humidity on the transmission of COVID-19*

### B. Configuration of Information Extraction Component

GrapeNLP performs exact matches of grammars to given pieces of text. Since we were to find papers that either claimed or denied the causal relation between the targeted factors and the spread of COVID-19, we developed a grammar (Fig. 3) that matched the entire text of a paper provided that at least one of these expressions appeared in the paper. This grammar calls the sub-grammar *causal_relation* (Fig. 5) to recognize our targeted expressions, and grammar *null_insert* (Fig. 4 left) to match all the paper text before and after the expression. It also generates tags <excerpt> before and after call to *causal_relation* to extract the expression.

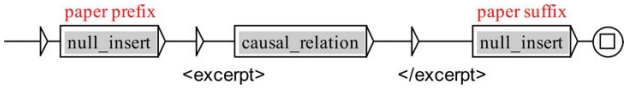


*Fig. 3 Grammar axiom*

Grammar *null_insert* (Fig. 4 left) recognizes 0, 1 or more arbitrary tokens thanks to the shortcut between the start and end boxes and to the <TOKEN> mask inside a loop.

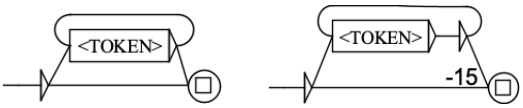


*Fig. 4 Grammars null_insert (left) and penalizing_insert (right)*

Grammar *causal_relation* (Fig. 5) detects expressions of claims or denials of the causal relations between a list of 1 or more factors and an effect. In case the relation is negated, it generates an empty tag <negation/> to notify it. Grammar *penalizing_insert* (Fig. 4 right) is used to recognize arbitrary token inserts, increasing the grammar recall at the expense of precision. To limit the length of these inserts, a negative score of -15 points is generated per token. An overall score threshold of -600 points is used to allow for at least 40 of these tokens (recall explicit grammar words add 14 points). Grammar *list_of_factors* (Fig. 6 top right) annotates the list of factors with a couple of <factors> tags. It uses grammar *temp_or_humidity* (Fig. 6 left) to detect single relevant factors. These are the same as the temperature and humidity synonyms used in the search engine query (Fig. 2). Finally, grammar *effect* (Fig. 6 bottom right) detects different expressions of effects such as spread, transmission or death.

## VI. Results

TABLE 1 illustrates some of the results from the grammar extraction. The column “factor” contains the matched factor or list of factors. Column “influential” indicates if the expression found is a claim (“Y”, <negation> tag absent) or a denial (“N”, <negation> tag present) of correlation. Column “score” contains the top match score. Finally, column “excerpt” presents the full matched statement along with some left and right context. Factors in the excerpt are highlighted in red, and the rest of the statement in green.

At the time of submitting our solution to the Kaggle challenge, it was tested on a version of the CORD-19 dataset that comprised 40,771 papers. The search engine found 32.7% of those (13,331) papers as having mentions to either temperature or humidity as well as to COVID-19. The grammar engine found claims or denials of causal relations in 41.92% of those (5,589), of which 2.67% (149) had a best match score above the threshold. Of those best matches, 91.95% (137) were claims and 8.05% (12) denials. The full table of results can be found at the Kaggle notebook page.[9]

It must be noted that the grammar can be improved in order to increase its coverage. This process can be done in quick

[9] https://www.kaggle.com/javiersastre/covid-19-temperature-and-humidity-summary-tables

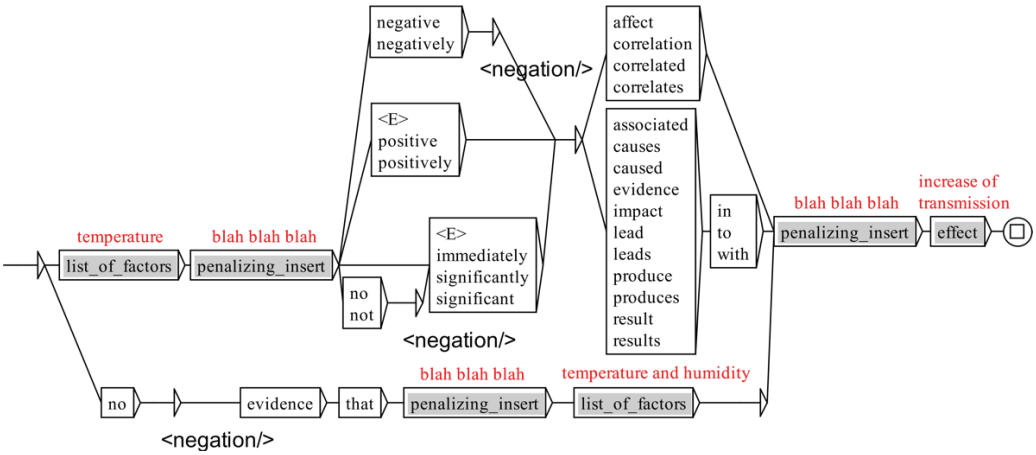


*Fig. 5 Grammar causal_relation*

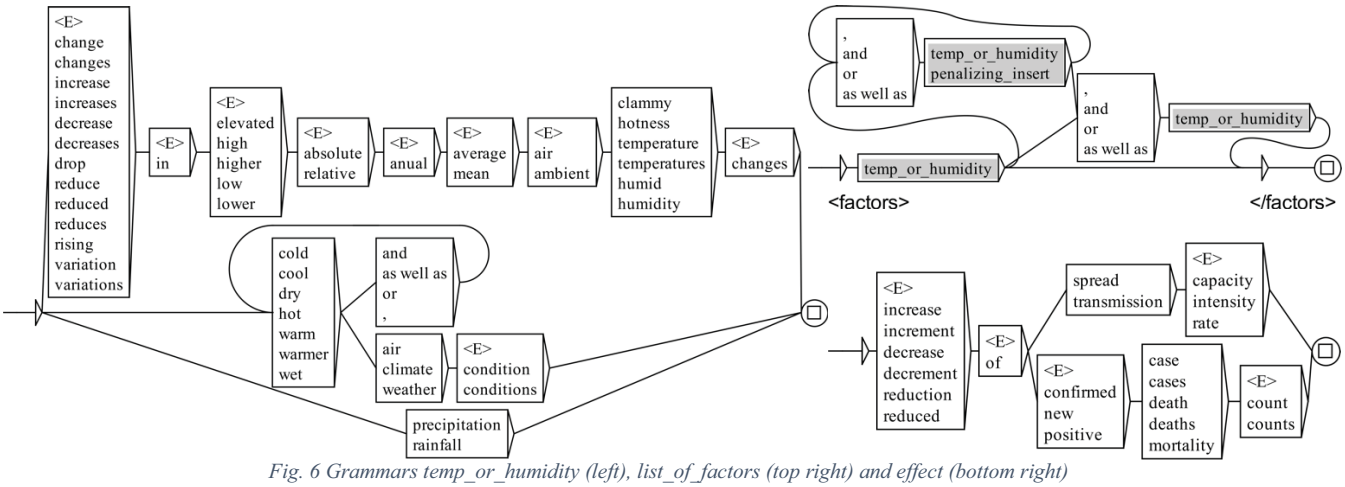


*Fig. 6 Grammars temp_or_humidity (left), list_of_factors (top right) and effect (bottom right)*

TABLE 1 SAMPLES OF EXTRACTED TEXT ILLUSTRATING POSITIVE/NEGATIVE CAUSAL IMPACT OF TEMPERATURE AND HUMIDITY ON COVID-19 TRANSMISSION

| Factor | Influential | Score | Excerpt |
|---|---|---|---|
| Changes in temperature | N | 68 | ic values are provided in Table 1 . ii. Changes in temperature shows no significant correlation with cases transmitted, deaths or recovered:Linear |
| Absolute humidity | N | 68 | Absolute humidity is negatively associated with daily death counts of COVID-19.\n\nG R A P H I C A L A B S |
| Mean Temperature | Y | -186 | Mean temperature of last two weeks (when b 3°C) was positively associated with newly confirmed COVID-19 cases. • 1°C rise in the mean temperature |
| high temperatures as well as low humidity | Y | 109 | climate such as Saudi-Arabia 14, 15 and high temperatures as well as low humidity were associated with increased disease spread rate 16 .Given the variant vulnerability of |
| increase in average temperature | Y | -97 | mean values for the other variables, an increase in average temperature from 1°C to 9°C was associated with a decrease in predicted cases at ADM1 level from 24 cases to 19 cases |

iterations by reviewing the papers found by the search engine but rejected by the grammar, then reviewing the portions of text where the relevant factors appeared, in order to locate target expressions that could have been missed. Then the grammar can be quickly updated and re-run at once since, in contrast with machine learning techniques, there is no model retraining step: the grammar itself is the model.

## VII. CONCLUSION

This paper described a new approach to uncovering key insights and discoveries on COVID-19 from a large corpus of scientific publications known as [1] the COVID-19 Open Research Dataset (CORD-19). It was developed in response to a call for new techniques that can assist the medical community in finding answers to a series of important scientific questions on COVID-19. Providing automated answers to these questions can help clinical experts to rapidly validate the results of recent research and propose new experimental hypotheses to help manage and reduce the impact of the COVID-19 pandemic.

The approach is based on two open-source components: the Apache Lucene search engine for keyword search, and the GrapeNLP grammar engine for further refinement of the matches found, enabling search functionality for more complex linguistic structures.

The approach was also validated by the CORD-19 Kaggle competition jury panel, who selected it as the best for creating summary tables on COVID-19 relevant factors. The approach was demonstrated using the effect of temperature and humidity on transmission as a use case. Grammars can be further refined via the Unitex graphical user interface, which is also open source, and extended to a wider range of

questions. Indeed, the approach was also demonstrated on identifying relevant COVID-19 risk factors.[10]

While deep learning approaches have also gained traction for addressing complex information retrieval tasks, they have short comings, such as the reliance on labelled data and the need for long training times and expensive hardware. They also suffer from lack of explainability and interpretability. The approach presented in this paper is efficient and has added benefit of clearer explainability in terms of the retrieval mechanism, as the grammar can easily be applied and reviewed by non-specialists. This COVID-19 pipeline is open sourced and is freely available for users, researchers and developers that may require to search papers using potentially complex linguistic structures.

REFERENCES


[1] L. L. Wang, K. Lo, Y. Chandrasekhar, R. Reas, J. Yang, D. Eide, K. Funk, R. Kinney, Z. Liu, W. Merrill and P. Mooney, "CORD-19: The Covid-19 Open Research Dataset.," *ArXiv,* 2020.

[2] K. Lo, L. L. Wang, M. Neumann, R. Kinney and D. S. Weld, "S2orc: The semantic scholar open research corpus.," *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics ,* pp. 4969-4983, July 2020.

[3] E. Zhang, N. Gupta, R. Nogueira, K. Cho and J. Lin, "Rapidly Deploying a Neural Search Engine for the COVID-19 Open Research Dataset: Preliminary Thoughts and Lessons Learned," *arXiv preprint arXiv:2004.05125,* 2020.

[4] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text trans-former," *arXiv preprint arXiv:1910.10683 ,* 2019.

[5] [Online]. Available: https://github.com/nmonath/kdcovid/. [Accessed August 2020].

[6] Q. Chen, Y. Peng and Z. Lu, "Biosentvec: creating sentence embeddings for biomedical texts," *2019 IEEE International Conference on Healthcare Informatics (ICHI)*, pp. 1-5, June 2019.

[7] [Online]. Available: https://github.com/gsarti/covid-papers-browser. [Accessed August 2020].

[8] J. Devlin, M. W. Chang, K. Lee and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *arXiv preprint arXiv:1810.04805.,* 2018.

[9] T. A. Kass-Hout and B. Snively, "AWS launches machine learning enabled search capabilities for COVID-19 dataset," 20 April 2020. [Online]. Available: https://aws.amazon.com/blogs/publicsector/aws-launches-machine-learning-enabled-search-capabilities-covid-19-dataset/. [Accessed 2020 August].

[10] [Online]. Available: https://ai.googleblog.com/2020/05/an-nlu-powered-tool-to-explore-covid-19.html. [Accessed August 2020].

[11] Z. Jiang, C. Zhnag, K. Talwar and M. C. Mozer, "Characterizing Structural Regularities of Labeled Data in Over parameterized Models," *arXiv:2002.03206,* 2020.

[12] A. Esteva, A. Kale, R. Paulus, K. Hashimoto, W. Yin, D. Radev and R. Socher, "Co-search: Covid-19 information retrieval with semantic search, question answering, and abstractive summarization," *arXiv preprint arXiv:2006.09595.,* 2020.

[13] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks.," *preprint arXiv:1908.10084,* 2019.

[14] O. Shahmirzadi, A. Lugowski and K. Younge, "Text similarity in vector space models: a comparative study.," *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA),* no. IEEE, pp. pp. 659-666, 2019.

[15] G. V. Cormack, C. L. Clarke and S. Buettcher, "Reciprocal rank fusion out-performs condorcet and individual rank learning methods," *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval,* 2009.

[16] A. Asai, K. Hashimoto, H. Hajishirzi, R. Socher and C. Xiong, "Learning to retrieve reasoning paths over wikipedia graph for question answering," *arXiv preprint arXiv:1911.10470,* 2019.

[17] X. Wang, X. Song, Y. Guan, B. Li and J. Han, "Compre-hensive named entity recognition on cord-19 with distant or weak supervision," *arXiv preprint arXiv:2003.12218,* 2020.

[18] Y. Liang and P. Xie, "Identifying ra-diological findings related to covid-19 from medical literature," *arXiv preprint arXiv:2004.01862 ,* 2020.

[19] S. Ahamed and M. Samad, "Information Mining for COVID-19 Research From a Large Volume of Scientific Literature," *arXiv preprint arXiv:2004.02085,* 2020.

[20] [Online]. Available: https://github.com/luisespinosaanke/cord-19-seven. [Accessed August 2020].

[21] L. Espinosa-Anke and S. Schockaert, "SeVeN: Augmenting word embeddings with unsupervised relation vectors," *arXiv preprint arXiv:1808.06068,* 2018.

[22] B. Rahdari, P. Brusilovsky, K. Thaker and H. K. Chau, "CovEx: An Exploratory Search System for COVID-19 Scientific Literature".

[23] K. Team, "With sports (and everything else) cancelled, this data scientist decided to take on COVID-19 | A Winner's Interview with David Mezzetti," 2020. [Online]. Available: https://medium.com/kaggle-blog/when-his-hobbies-went-on-hiatus-this-kaggler-made-fighting-covid-19-with-data-his-mission-a-e306419b99a5. [Accessed Auguat 2020].

[24] Y. Wang, L. Wang, M. Rastegar-Mojar, S. Moon, F. Shen, N. Afzal, S. Liu, Y. Zeng, S. Mehrabi, S. Sohn and H. Liu, "Clinical information extraction applications: A literature review," *Journal of biomedical informatics,* no. 77, pp. pp.34-49, 2018.

[25] A. B. Abacha and P. Zweigenbaum, "Automatic extraction of semantic relations between medical entities: a rule based approach," *Journal of biomedical semantics 2, no. S5,* no. S4, 2011.

[26] M. Gross, "1 The Construction of Local Grammars," *Finite-state language processing,* p. 329, 1997.

[27] J. M. Sastre Martinez, J. Sastre and J. Garcia-Puga, "Boosting a Chatterbot Understanding with a Weighted Filtered-Popping Network Parser," 2009.

[28] F. B. Mesmia, F. Zid, K. Haddar and D. Maurel, "ASRextractor: A Tool extracting Semantic Relations between Arabic Named Entities," *Procedia Computer Science 117,* pp. 55-62, 2017.

[29] A. Pierson and C. Fairon, "Study of lexical aspect in the French medical language. Development of a lexical resource," *Proceedings of the 2nd Clinical Natural Language Processing Workshop ,* pp. pp.55-64, 2019.

[30] "Lucene," [Online]. Available: https://lucene.apache.org/pylucene/.

[31] "Apache Lucene Core," [Online]. Available: https://lucene.apache.org/core/.

[32] P. Sebastien, T. Nakamura and S. Voyatzi, "Unitex, a corpus processing system with multi-lingual linguistic resources," in *eLexicography in the 21st century: new challenges, new applications (eLEX'09)*, 2009.

[33] J. M. Sastre Martínez, "Efficient parsing using filtered-popping recursive transition networks," in *International Conference on Implementation and Application of Automata,*, Springer, Berlin, Heidelberg, 2009.

[34] J. M. Sastre Martinez, Efficient finite-state algorithms for the application of local grammars, Doctoral dissertation of Université Paris Est & Universidad de Alicante, 2011.

[35] J. M. Sastre and M. Forcada, "Efficient parsing using recursive transition networks with output," *3rd Language & Technology Conference (LTC'07),* pp. pp.208-284, 2007.


[10] https://www.kaggle.com/pauljireland/covid-19-risk-factor-summary-tables