COVAT (COVID-19 vaccine tracker)

Akash Saurabh,FNU Vaibhav

CSCI 7200 Masters Project Report Sp 2020

Department of Computer Science, The University Of Georgia
Athens, GA, USA

ascoo@uga.edu

VaibhavViru@uga.edu

***Abstract*— This paper describes the master's project, which is a course with code CSCI 7200 being offered at The University Of Georgia, Athens. As a part of this course, we (Akash Saurabh and FNU Vaibhav) completed the project named COVAT (COVID-19 vaccine tracker). The main objective of our COVAT project is to come up with a knowledge graph for tracking the covid-19 vaccine update. These updates can help researchers with a better understanding of effective cataloging of various in trial vaccines and their related use of launched vaccines.**
We have extracted our data set from the U.S. National Library of Medicine website clinicaltrials.gov:
https://clinicaltrials.gov/ct2/results?cond=COVID-19&draw=2&rank=2#rowId1
The data from this source was in the raw HTML file format. We used this data set source containing thousands of webpages to create the knowledge graph. The main libraries used for facilitating the desired use case for the creation of the knowledge graph include BeautifulSoap, Scrapy, Minidom, NLTK, SPACY, and Stanfordcorenlp with the help of the Python framework.

***Keywords*—SPACY,Knowledge Graph,Vaccine,Covid-19,triples**

I. Introduction

We have the aim to create the knowledge graph for the triples formed by the dataset curated from approximately 4000+ webpages gathered from the U.S. National Library of Medicine website clinicaltrials.gov. The webpages in HTML format are processed by using the BeautifulSoap python library. Next, the raw text is extracted and triple formation is done by using Minidom and SPACY python libraries.

The triples formed will be used to form the knowledge graph of the COVID-19 related vaccine data. One of the ways to deal with interconnected entities is with the use of a data science tool called Knowledge Graphs. Here we know that the graph can be represented as a set of nodes and edges. Entities are represented by the nodes. The edges connect the nodes. Some examples of interconnected entities are events, people, places, and organizations.

Knowledge graphs contain entity-pairs. These entity-pairs can be traversed to uncover meaningful connections. These meaningful connections are observed in data sets that are unstructured.

Our aim through this project is to provide a more robust and concise approach towards the cataloging of vaccine development data sets. We have gathered these data as plain HTML files from the U.S. National Library of Medicine website clinicaltrials.gov. Covid-19 has become a widespread global disease that has impacted the economies and national interests of many nations worldwide due to the unfortunate breaking of this disease. Thus this project aims towards helping the researchers and doctors with a knowledge graph based overview of the meaningful relationship created by applying various Python libraries and designing different function sets to bind these libraries for a specific implementation of our desired use cases.

The meaning from the unstructured data is obtained via the implementation that we will be discussing in the following sections. Covid-19 vaccine data may have information on various aspects of it. Our project aims at extracting all of the meaning from raw Covid-19 vaccines update data sets spread across many webpages. Without this project, the analysis of these unstructured datasets would require very rich domain knowledge for the researchers to make meaning out of thousands of web pages.

Thus our paper is related to processing and structuring of Covid-19 related data spread over clinical information web pages of various web pages across the source web pages we selected. We will further understand in this paper how this data set source can be easily expanded depending on the computational capacity available to any researcher/s. Here the computational capacity refers to resources utilized for processing of Data set from the source we selected. Further on in this paper we will see implementation and result obtained for our desired use cases. This paper will also show results for better visualization of obtained knowledge graphs. Our project can be accessed at :

GitHub: https://github.com/ascoolakash/covat

Our adopted processing and structuring strategies are based on the basic understanding of triple formation from unstructured data sets and knowledge graphs that are obtained from it. The visual analysis can be done by even little understanding of visual dashboard analysis. The general structure of our project can be understood by the diagram in Fig. 1 below.

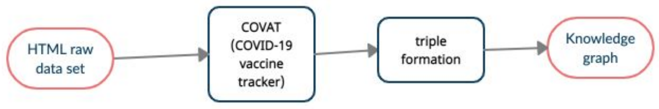


Figure : 1 Basic flow of COVAT (COVID-19 vaccine tracker).

The structuring of the rest of the paper is as follows:

Structuring and analyzing the raw unstructured Data sets has been a very old topic of research. Our project helps to add value by processing and structuring the Covid-19 vaccine-related data sets spread over various research publishing web pages.

II. Objective

The objective of our project is to create a visually and analytically meaningful structure out of unstructured Covid-19 vaccine-related data spread over 4000+ web pages from our data set source. The structured data sets will be displayed by use of the knowledge graph. The knowledge graph can be used for making the meaning out of unstructured web page data sets available in the source of datasets we selected for this project. We will be creating specific rules and methods to reach our objective of this project.

III. Related Work

In the process of knowledge graph creation, the most important part is triple extraction or knowledge extraction. In this section, we are going to discuss three different domains related to our research i.e. knowledge extraction. The first approach is supervised; it involves labeling entities in text corpus then training the classifiers to capture relations between pairs of entities in a sentence or a combination of sentences [1,2,3]. This approach is expensive therefore the algorithm would be biased towards a particular corpus. The second approach is unsupervised, in this approach we use extracted data directly to model and generate triples using a rule-based mechanism[4,5]. The last approach is heterogeneous supervision in this we create triples based on rule-based tagging[5,7,8,9,10]. In this project, we are working on an unsupervised machine learning algorithm to create triples.

In the past, there are several attempts to extract triples/semantic relationships using unsupervised learning, but none have been specifically aimed to create multiple triplets in a single sentence by figuring out the total number of in entity chunks. A part of that past work people are using ROOT as a relation tag main character in this paper we are discussing both ROOT and ACL dependency parsing. The triplets created in the manner of noun-verb-noun relationships that we are targeting. Some of the most relevant studies are indicated here.

Carlson et al. [11,12] present a framework for semi-supervised learning in a similar context (NELL). Specifically, their Coupled Pattern Learner (CPL) methodology lays out the broad framework of how one can extract word-relationships given an ontology and a text corpus. The CPL's precision ranges from 70% to 100% for a variety of relationships (For example, 'X is a building', 'Y is a conference', 'Company P produces product Q' etc.), averaging a precision of 89%. Pantel et al. present an algorithm [13] that is designed to harvest "Is-A" (protein::biopolymer), "Part-Of" (oxygen::air) relations along with succession (Ford::Nixon), reaction (boron::fluorine), and production (kidney::kidney stones) pairs. They achieve a precision of 80% on news articles-text and 91% precision on a chemistry textbook. Similar to our COVAT algorithm, the three core steps of their algorithm are Text Extraction, Triple Creation, and Rule Management.

IV. Data Source

We use the data set available in the U.S. National Library of Medicine website: clinicaltrials.gov (https://clinicaltrials.gov/ct2/results?cond=COVID-19&draw=2&rank=2#rowId1)

This data set is used to extract raw text data. On the website, there is more than 4000 research going on the direction of COVID-19. We see the number of research increase daily in this direction therefore the list has been updated on a regular basis. After the extraction of data, we get text data of almost 20MB. This data set will further be used to extract meaning from it.

V. Architecture

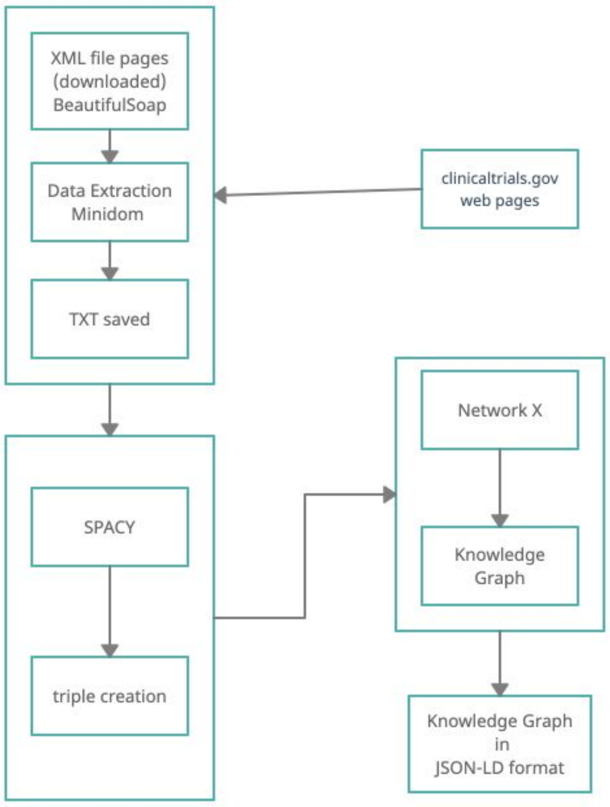


Figure 2: Architecture of COVAT (COVID-19 vaccine tracker)

In this section, we are going to discuss the architecture and functioning of COVAT. We are first extracting the data from clinicaltrials.gov website using pythons library described in the above figure. Then we use pythons Spacy library to extract the semantic relationships using dependency parsing. These saved triples were then sent to pythons networkx library to create a knowledge graph which further helped us to save the knowledge graph in json-ld format.

VI. Knowledge Extraction Procedure

In this section, we are going to discuss relation extraction and heterogenous supervision, including the rule-based mechanism.

1) Relation Extraction

Here we see that the Relation extraction is conducted at the sentence level [14]. Here for a given sentence ‘s’, then an entity which is a token span in ‘s’ can be represented as (e1,..., en).

The relation mentions a triple (e1, e2, s) which consists of an ordered entity pair (e1, e2) in sentence s. Here the relation extraction task is to categorize relation mentions into a given set of relation types R, or Not-Target-Type (None).

This means the type of relation mentioned does not belong to R.

2) Heterogeneous Supervision

Similar to [15], we employ labeling functions as basic units to encode supervision information and generate annotations. As we know different supervision information may have different proficient subsets. So to make these subsets even more meaningful we require each labeling function to encode every other supervision information. Mainly, in the relation extraction scenario, we require each labeling function to only annotate one relation type which is based on one elementary piece of information. We should see that knowledge-based labeling functions are also considered to be noisy. This is as relation extraction is conducted at sentence-level.

An example of this is although the president of (Biden, USA) exists in KB, it should not be assigned with “Biden was born in Newyork, Newyork, USA”, because the “ president of “ is irrelevant to the context.

3) Features and Pre Processing

We see here that the entity-relation-entity triads are the equivalent of “features” for the given method. These are extracted using the Syntactic Tree output from the Stanford Parser [16]. The Parts-of-Speech tagger from the SpaCy Python libraries is used. Moreover, the SpaCy lemmatizer [17] is used to lemmatize nouns and also get to their tense-less root forms. The example for this is, ‘poured’, ‘pouring’ and ‘pours’ are all converted to ‘pour’. Further same steps are used for completing the pre-processing.

VII. Manual Data Creation

In this section here, we are going to discuss manual triple creation and the related rule management. We are using spaCy display function and dependency parsing to check parts of speech of each word in a sentence which we can see in the below figure:

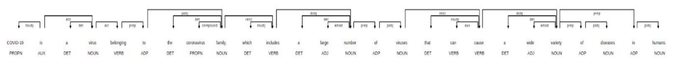
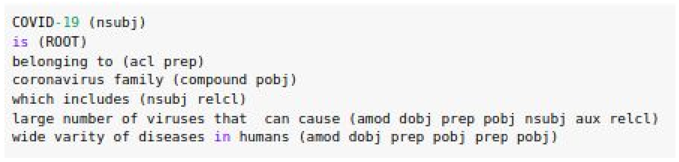


Figure 3: Dependency parsing in a complex sentence

In the above figure, we can see that the noun chunks in a complex sentence can be more than two in respect of a simple sentence that has two or fewer noun chunks. In this paper, we provide a solution to resolve this problem. We created a method in which we create multiple triples from a complex sentence.



```
COVID-19 (nsubj)
is (ROOT)
belonging to (acl prep)
coronavirus family (compound pobj)
which includes (nsubj relcl)
large number of viruses that  can cause (amod dobj prep pobj nsubj aux relcl)
wide varity of diseases in humans (amod dobj prep pobj prep pobj)
```

Figure 4: Manual identification of entity and relation

In the above figure 4, we choose words to generate meaningful entities and relations. We do manual identification of entities and relations for getting relevant entities and their related relationship. For creating entities we choose the format of ...adj-verb-noun. Here for relation, we choose both ROOT and ACL dependency parsing. For checking POS we choose the spacy dependency parsing method.

VIII. NLP Library

This section discusses the libraries used to create this project. We selected three popular python libraries used for natural language processing to compare and select the best library to match the requirements of our project’s use cases. These libraries are spaCy, CoreNLP, and NLTK. Let us have a look at their comparative efficiency and feature performance.

Below Table 1 shows the comparative speed of Key functionalities of spaCy, CoreNLP, and NLTK libraries. The functionalities discussed are Tokenizer, Tagging, and Parsing.

Library Package	Tokenizer	Tagging	Parsing
spaCy	0.2ms	1ms	19ms
CoreNLP	2ms	10ms	49ms
NLTK	4ms	443ms	–

Table 1: Comparative Speed of Key Functionalities (Tokenizer, Tagging, Parsing)

We can see from this table that spaCy is better when compared to CoreNLP and NLTK for Tokenizer, Tagging, and Parsing. So selecting spaCy is the best option when based on table 1.

Next, we have a look at the accuracy of entity extraction that will have an impact on triple formation from the raw HTML data sets.

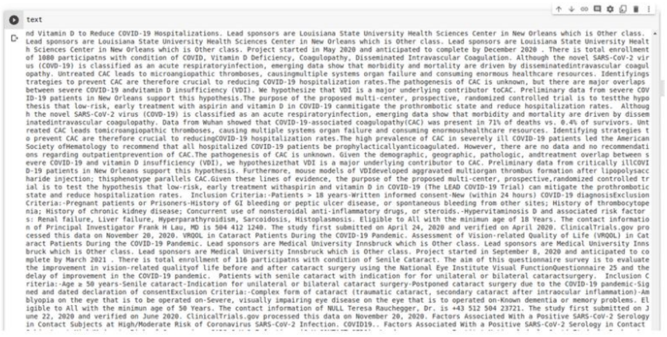
Library Package	Precision	Recall	F-Score
spaCy	0.72	0.65	0.69
CoreNLP	0.79	0.73	0.76
NLTK	0.51	0.65	0.58

Table 2: Relative accuracy for Entity Extraction (Precision, Recall, F-Score)

While analyzing table 2 we see that spaCy is a better selection as the tokenizer, tagging and parsing are better for spaCy so comparable differences between spaCy and CoreNLP for precision, recall and f-score can be ignored. Thus we go ahead with the spaCy python library for our project implementation. While analyzing these libraries we came to know that dependency parsing is not an inbuilt function. For creating dependency parsing for the analysis of parts of speech we have to pipeline rules for dependency parsing in stanfordcorenlp. Due to a lack of appropriate understanding of linguistics, we choose to try different libraries. We check the spaCy python library which provides better algorithms than NLTK.

IX. RESULTS

We extracted the data from the clinicaltrials.gov website which is shown in the below figure.

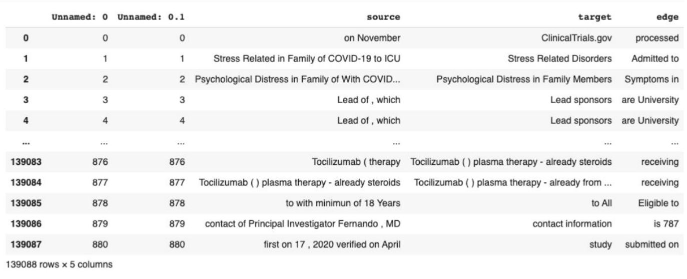


```
text
nd Vitamin D to Reduce COVID-19 Hospitalizations. Lead sponsors are Louisiana State University Health Sciences Center in New Orleans which is Other class.
Lead sponsors are Louisiana State University Health Sciences Center in New Orleans which is Other class. Lead sponsors are Louisiana State University Healt
h Sciences Center in New Orleans which is Other class. Project started in May 2020 and anticipated to complete by December 2020 . There is total enrollment
of 1080 participatns with condition of COVID, Vitamin D Deficiency, Coagulopathy, Disseminated Intravascular Coagulation. Although the novel SARS-CoV-2 vir
us (COVD-19) is classified as an acute respiratoryinfection, emerging data show that morbidity and mortality are driven by disseminatedintravascular coagul
opathy. Untreated CAC leads to microangiopathic thromboses, causingmultiple systems organ failure and consuming enormous healthcare resources. Identifyings
trategies to prevent CAC are therefore crucial to reducing COVID-19 hospitalization rates.The pathogenesis of CAC is unknown, but there are major overlaps
between severe COVID-19 andvitamin D insufficiency (VDI). We hypothesize that VDI is a major underlying contributor toCAC. Preliminary data from severe COV
ID-19 patients in New Orleans support this hypothesis.The purpose of the proposed multi-center, prospective, randomized controlled trial is to testthe hypo
thesis that low-risk, early treatment with aspirin and vitamin D in COVID-19 canmitigate the prothrombotic state and reduce hospitalization rates.  Althoug
h the novel SARS-CoV-2 virus (COVD-19) is classified as an acute respiratoryinfection, emerging data show that morbidity and mortality are driven by dissem
inatedintravascular coagulopathy. Data from Wuhan showed that COVID-19-associated coagulopathy(CAC) was present in 71% of deaths vs. 0.4% of survivors. Unt
reated CAC leads tomicroangiopathic thromboses, causing multiple systems organ failure and consuming enormoushealthcare resources. Identifying strategies t
o prevent CAC are therefore crucial to reducingCOVID-19 hospitalization rates.The high prevalence of CAC in severely ill COVID-19 patients led the American
Society ofHematology to recommend that all hospitalized COVID-19 patients be prophylacticallyanticoagulated. However, there are no data and no recommendati
ons regarding outpatientprevention of CAC.The pathogenesis of CAC is unknown. Given the demographic, geographic, pathologic, andtreatment overlap between s
evere COVID-19 and vitamin D insufficiency (VDI), we hypothesizethat VDI is a major underlying contributor to CAC. Preliminary data from critically illCOVI
D-19 patients in New Orleans support this hypothesis. Furthermore, mouse models of VDIdeveloped aggravated multiorgan thrombus formation after lipopolysacc
haride injection; thisphenotype parallels CAC.Given these lines of evidence, the purpose of the proposed multi-center, prospective,randomized controlled tr
ial is to test the hypothesis that low-risk, early treatment withaspirin and vitamin D in COVID-19 (The LEAD COVID-19 Trial) can mitigate the prothrombotic
state and reduce hospitalization rates.  Inclusion Criteria:-Patients > 18 years-Written informed consent-New (within 24 hours) COVID-19 diagnosisExclusion
Criteria:-Pregnant patients or Prisoners-History of GI bleeding or peptic ulcer disease, or spontaneous bleeding from other sites; History of thrombocytope
nia; History of chronic kidney disease; Concurrent use of nonsteroidal anti-inflammatory drugs, or steroids.-Hypervitaminosis D and associated risk factor
s: Renal failure, Liver failure, Hyperparathyroidism, Sarcoidosis, Histoplasmosis. Eligible to All with the minimun age of 18 Years. The contact informatio
n of Principal Investigator Frank H Lau, MD is 504 412 1240. The study first submitted on April 24, 2020 and verified on April 2020. ClinicalTrials.gov pro
cessed this data on November 20, 2020. VRQOL in Cataract Patients During the COVID-19 Pandemic. Assessment of Vision-related Quality of Life (VRQOL) in Cat
aract Patients During the COVID-19 Pandemic. Lead sponsors are Medical University Innsbruck which is Other class. Lead sponsors are Medical University Inns
bruck which is Other class. Lead sponsors are Medical University Innsbruck which is Other class. Project started in September 8, 2020 and anticipated to co
mplete by March 2021 . There is total enrollment of 116 participatns with condition of Senile Cataract. The aim of this questionnaire survey is to evaluate
the improvement in vision-related qualityof life before and after cataract surgery using the National Eye Institute Visual FunctionQuestionnaire 25 and the
delay of improvement in the COVID-19 pandemic.  Patients with senile cataract with indication for for unilateral or bilateral cataractsurgery.  Inclusion C
riteria:-Age ≥ 50 years-Senile cataract-Indication for unilateral or bilateral cataract surgery-Postponed cataract surgery due to the COVID-19 pandemic-Sig
ned and dated declaration of consentExclusion Criteria:-Complex form of cataract (traumatic cataract, secondary cataract after intraocular inflammation)-Am
blyopia on the eye that is to be operated on-Severe, visually impairing eye disease on the eye that is to operated on-Known dementia or memory problems. El
igible to All with the minimun age of 50 Years. The contact information of NULL Teresa Rauchegger, Dr. is +43 512 504 23721. The study first submitted on J
une 22, 2020 and verified on June 2020. ClinicalTrials.gov processed this data on November 20, 2020. Factors Associated With a Positive SARS-CoV-2 Serology
in Contact Subjects at High/Moderate Risk of Coronavirus SARS-CoV-2 Infection. COVID19.. Factors Associated With a Positive SARS-CoV-2 Serology in Contact
```

Figure 5: Extracted data from U.S.National Library of Medicine website

From that text data, we created 139,088 numbers of triples.

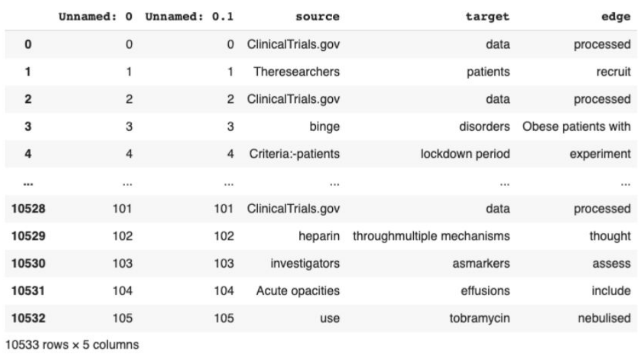
We also calculated data from neuralcoref python library and found out that we are getting 10533 numbers of triples has been generated. By using our code we are getting almost thirteen times more triples than the given Neuralcoref library earlier.



	Unnamed: 0	Unnamed: 0.1	source	target	edge
0	0	0	on November	ClinicalTrials.gov	processed
1	1	1	Stress Related in Family of COVID-19 to ICU	Stress Related Disorders	Admitted to
2	2	2	Psychological Distress in Family of With COVID...	Psychological Distress in Family Members	Symptoms in
3	3	3	Lead of , which	Lead sponsors	are University
4	4	4	Lead of , which	Lead sponsors	are University
...	...	...	...	...	...
139083	876	876	Tocilizumab (therapy	Tocilizumab ( ) plasma therapy - already steroids	receiving
139084	877	877	Tocilizumab ( ) plasma therapy - already steroids	Tocilizumab ( ) plasma therapy - already from ...	receiving
139085	878	878	to with minimun of 18 Years	to All	Eligible to
139086	879	879	contact of Principal Investigator Fernando , MD	contact information	is 787
139087	880	880	first on 17 , 2020 verified on April	study	submitted on

139088 rows × 5 columns

Figure 6: Created triples 139088 rows x 5 column



	Unnamed: 0	Unnamed: 0.1	source	target	edge
0	0	0	ClinicalTrials.gov	data	processed
1	1	1	Theresearchers	patients	recruit
2	2	2	ClinicalTrials.gov	data	processed
3	3	3	binge	disorders	Obese patients with
4	4	4	Criteria:-patients	lockdown period	experiment
...	...	...	...	...	...
10528	101	101	ClinicalTrials.gov	data	processed
10529	102	102	heparin	throughmultiple mechanisms	thought
10530	103	103	investigators	asmarkers	assess
10531	104	104	Acute opacities	effusions	include
10532	105	105	use	tobramycin	nebulised

10533 rows × 5 columns

Figure 7: Created triples after using Neuralcoref library (10533 rows x 5 columns)

By analyzing 5% of total triples obtained by our own methods that we created we find that 70% of triples were meaningful to the human evaluators. The triples extracted by our code provide a more meaningful result which we can check in the figure. Then we use the networkx python library to plot the knowledge graph on the jupyter notebook. Some results of knowledge graph created are in below figures:

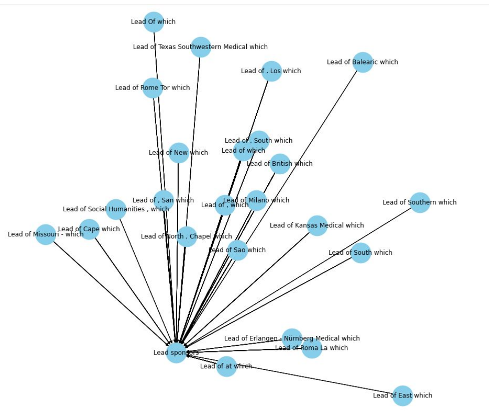


Figure 8: Knowledge graph ("are University")

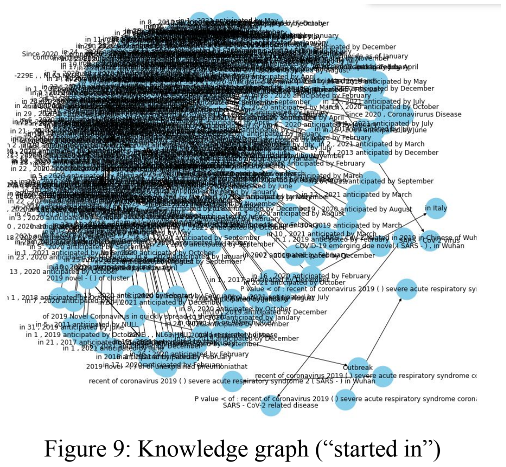


Figure 9: Knowledge graph ("started in")

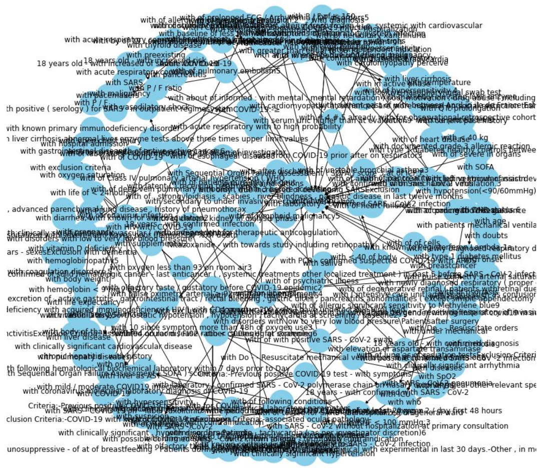


Figure 10: Knowledge graph(“associated with”)

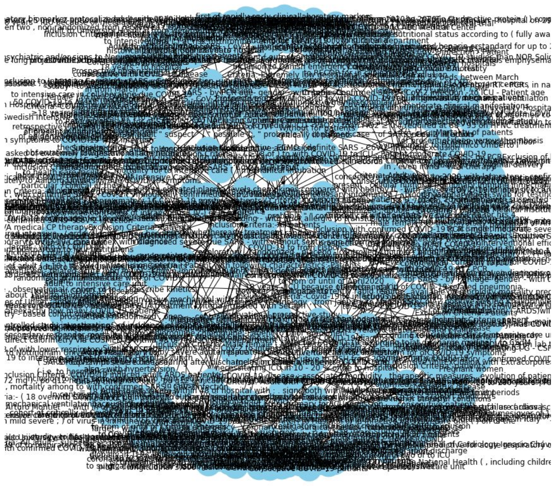


Figure 11: Knowledge graph (“admitted to”)

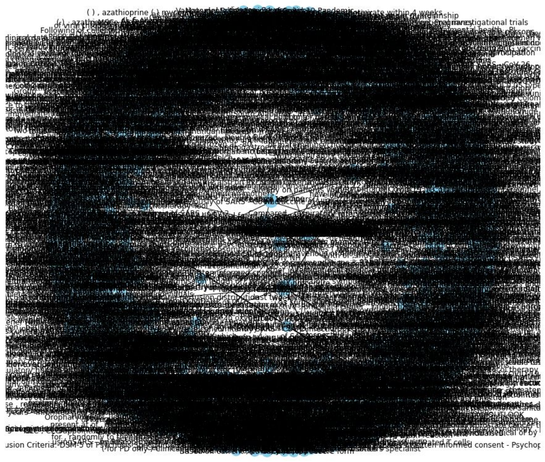


Figure 12: Knowledge graph(“is”)

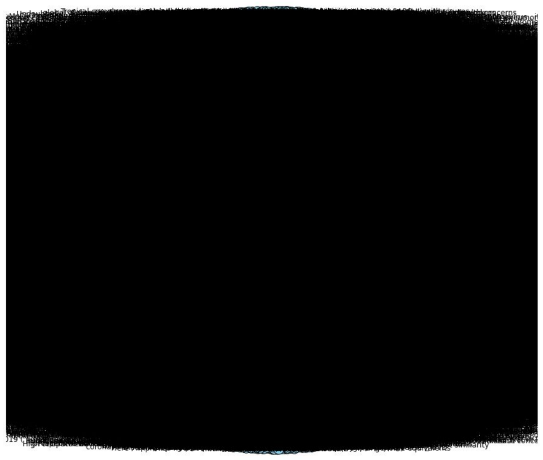


Figure 13: Knowledge graph with complete dataset

X. CONCLUSION

The main objective of our project was to create a knowledge graph using raw HTML data sets from the U.S. National Library of Medicine website clinicaltrials.gov:

https://clinicaltrials.gov/ct2/results?cond=COVID-19&draw=2&rank=2#rowId1

We extract XML pages from HTML pages by using BeautifulSoap. Beautifulsoup is a python package used for the parsing of HTML and XML documents. This XML is parsed using minidom. Minidom is a Python library. It stands for Minimal DOM because it is a minimal implementation of the Document Object Model interface. The resultant here is data in the form of a raw text file of size 20MB. We apply spaCy python library to this text file obtained for finding the triples. Once the triple formation was done we applied network X on these triples to form knowledge graphs. The obtained knowledge graph was saved for better view in json-ld format. Different density knowledge graphs were obtained depending on the amount of triples that we considered. As we saw that by analyzing 6% of total triples obtained by our own methods that we created we find that 70% of triples were meaningful to the human reviewers.

XI. FUTURE WORK

This project can be implemented in many different use cases where the visualization can be implemented by using Neo4j. We have implemented a basic overview by using Neo4j. Future researchers can expand this technology to expand further scopes related to knowledge graph visualization. Furthermore, we want to create our own library using the methods that we have created as our methods showed better for triple clarity when compared to the pre-existing neuralcoref python library.The use case of our own library is promising as our methodology found by analyzing 6% of total triples obtained by our own methods that we created we find that 70% of triples were meaningful to the human reviewers. This is the right method to work on in our project which can also add promising functionalities in the future.

Acknowledgment

We would like to express special thanks of gratitude to my Master's Project Guide Dr. Arpinar. This project also helped me in doing a lot of research and I came to know about so many new things. We are thankful to him. The Nobel thought of creating a knowledge graph for vaccine tracking is important and crucial in the current situation. It helps a lot of researchers to work against COVID-19.

References

[1] GuoDong, Z., Jian, S., Jie, Z., & Min, Z. (2005, June). Exploring various knowledge in relation extraction. In Proceedings of the 43rd annual meeting on association for computational linguistics (pp. 427-434). Association for Computational Linguistics.

[2] Surdeanu, M., & Ciaramita, M. (2007, March). Robust information extraction with perceptrons. In Proceedings of the NIST 2007 Automatic Content Extraction Workshop (ACE07).

[3] Shinyama, Y., & Sekine, S. (2006, June). Preemptive information extraction using unrestricted relation discovery. In Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics (pp. 304-311). Association for Computational Linguistics.

[4] Banko, M., Cafarella, M. J., Soderland, S., Broadhead, M., & Etzioni, O. (2007, January). Open information extraction for the web. In IJCAI (Vol. 7, pp. 2670-2676).

[5] Etzioni, O., Cafarella, M., Downey, D., Popescu, A. M., Shaked, T., Soderland, S., ... & Yates, A. (2005). Unsupervised named-entity extraction from the web: An experimental study. Artificial Intelligence, 165(1), 91- 134.

[6] Bunescu, R. C., & Mooney, R. (2007, June). Learning to extract relations from the web using minimal supervision. In Annual meeting-association for Computational Linguistics (Vol. 45, No. 1, p. 576).

[7] Rozenfeld, B., & Feldman, R. (2008). Selfsupervised relation extraction from the Web. Knowledge and Information Systems, 17(1), 17-33.

[8] R. E. Sorace, V. S. Reinhardt, and S. A. Vaughn, "High-speed digital-to-RF converter," U.S. Patent 5 668 842, Sept. 16, 1997.

[9] (2002) The IEEE website. [Online]. Available: http://www.ieee.org/

[10] M. Shell. (2002) IEEEtran homepage on CTAN. [Online]. Available: http://www.ctan.org/tex-archive/macros/latex/contrib/supported/IEEEtran/

[11] Carlson, A., Betteridge, J., Kisiel, B., Settles, B., Hruschka Jr, E. R., & Mitchell, T. M. (2010, July). Toward an Architecture for Never-Ending Language Learning. In AAAI (Vol. 5, p. 3).

[12] Carlson, A., Betteridge, J., Wang, R. C., Hruschka Jr, E. R., & Mitchell, T. M. (2010, February). Coupled semi-supervised learning for information extraction. In Proceedings of the third ACM international conference on Web search and data mining (pp. 101-110). ACM.

[13] Pantel, P., & Pennacchiotti, M. (2006, July). Espresso: Leveraging generic patterns for automatically harvesting semantic relations. In Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics (pp. 113-120). Association for Computational Linguistics.

[14] Junwei Bao, Nan Duan, Ming Zhou, and Tiejun Zhao. 2014. Knowledge-based question answering as machine translation. Cell, 2(6)

[15] Alexander J Ratner, Christopher M De Sa, Sen Wu, Daniel Selsam, and Christopher Re. 2016. Data ´ programming: Creating large training sets, quickly. In Advances in Neural Information Processing Systems, pages 3567–3575.

[16] Marie-Catherine de Marneffe, Bill MacCartney and Christopher D. Manning. 2006. Generating Typed Dependency Parses from Phrase Structure Parses. In LREC 2006.

[17] Bird, Steven, Edward Loper and Ewan Klein (2009), Natural Language Processing with Python. OReilly Media Inc.

[18] https://www.analyticsvidhya.com/blog/2017/04/natural-language-processing-made-easy-using-spacy-%E2%80%8Bin-p