Covid-19 Knowledge Graphs: Relation classification and relation prediction between chemicals, genes and diseases

Marija Chaushevska
Department of Knowledge Technologies
Jožef Stefan International Postgraduate School
Jamova 39, 1000 Ljubljana, Slovenia
e-mail: marija.chaushevska@ijs.si

ABSTRACT

Advances in machine learning and deep learning methods, together with the increasing availability of large-scale pharmacological, genomic, and chemical datasets, have created opportunities for identifying potentially useful relationships within biochemical networks. The biomedical community's collective understanding of how chemicals, genes and phenotypes interact is distributed across the text of over 24 million research articles. These interactions offer insights into the mechanisms behind higher order biochemical phenomena, such as drug-drug interactions and variations in drug response across individuals.

Therefore, in this paper, within Semantic Web Technologies course, I propose a relation classification system for classifying relations between biomedical entities such as chemicals, genes and diseases from sentences that are extracted from biomedical research articles. I have used pre-trained BERT model architecture for classifying relations between chemical-disease, chemical-gene, gene-disease and gene-gene entity types and it achieved enviable results. As well as, I have used unseen marked sentences from biomedical articles for relation prediction and it's probability, for each relationship type (chemical-gene, chemical-disease, gene-disease and gene-gene). The predicted relations are visually presented using Knowledge Graphs (KGs). However, the main focus in this seminar work is on Covid-19 disease and presenting (visualizing) predicted relations between Covid-19 disease and other chemicals and genes that Covid-19 is related to.

Key words: relation classification, biomedical relations, pretrained model, Covid-19, entities, relations, knowledge graph

1 INTRODUCTION

Biochemistry is a cross-discipline, incorporating elements of pharmacology, biology, and chemistry. The large number of disciplines associated with biochemistry makes it challenge to identify new relationships. On the other hand, computational prediction is becoming a crucial and effective strategy for identifying links, given its potentials to reduce the high failure risk of expensive and time-consuming laboratory experiments. The network of interactions among biomedical entities—chemicals, genes and diseases—has long been of interest to biomedical researchers. Structured relationships offer insights into the mechanisms behind important higher-order relationships, such as drug-drug interactions.

In this seminar work I have proposed biomedical relation classification system that classifies relations between chemicals, genes and diseases, more precisely classifying relations between chemical-disease, chemical-gene, gene-disease and gene-gene relationsgip types. Besides relation classification, I have done a relation prediction on unseen sentences, which predicts relations between two biomedical entitites and visualize them using Knowledge Graph.

Relation classification [1] is an important NLP task that identifies the type of relationship that exists between two given entities in a sentence. As well as, it is a task that has achieved great attention in recent years. It is crucial for inferring semantic relatedness between entities in a piece of text. The type of this relationship depends on other words present in the sentence and more generally on the meaning that the sentence is trying to convey. Sentences usually have multiple entities and a single entity can be related to multiple other entities in the sentence. The type of this relation can be different in each case. So, in relation classification data sets it is necessary to mark the two entities whose relation is to be classified. The reason relation classification can be an interesting and challenging problem is the types of relation labels that are specified. However, relation classification is very domain-specific and it takes a lot of effort to label data for a new domain. In the biomedical domain, a relation can exist between various entity types like chemical-gene, gene-disease, chemical-disease, gene-gene etc.

The state-of-the-art methods for relation classification are primarily based on Convolutional or Recurrent Neural Networks. Recently, the pre-trained BERT model achieves very successful results in many NLP classification/ sequence labeling tasks, so that's why I have decided for this seminar work to propose a relation classification system using a pre-trained BERT [2] model. Besides relation classification, I have done a relation prediction on unseen sentences from biomedical articles that contain at least two biomedical entitites (chemical, gene or desease).

The main motivation for this seminar work is the need to accumulate insights from large voulumes of information for the novel SARS-CoV-2 virus. COVID-19 [3] is the first global pandemic within a century and has resulted in hundreds of thousands of deaths as well as severe disruption of both economic and social structures worldwide. Thousands of new scientific articles on the virus are being published weekly, leading to a rapid increase in the cumulative knowledge about the coronavirus disease (COVID-19). COVID-19 has heightened the need for tools that enable researchers to search vast scientific corpora to find specific information, visualize connections across the data, and discover related information in the data.

The need to develop vaccines, therapies, and rapid tests for this virus is urgently needed. To facilitate the scientific and medical effort to stop this pandemic, most publishers are making full text of COVID-19 related manuscripts freely available. Several COVID-19 dedicated search engines have come online to address the need for information retrieval of scientific literature on the disease. Search engines like Sketch Engine COVID-19, Sinequa COVID- 19 Intelligent Search, Microsoft's CORD19 Search, and Amazon's CORD19 search use a variety of methodologies such as keyword search, natural language queries, semantic relevancy, and knowledge graphs. However, these engines return thousands of search results that overlook inherent relationships between scientific articles, such as subject topic and citations, and do not provide tools to visualize relationships, which is beneficial for knowledge discovery.

A large part of the world's population is still affected by this virus. Many people are in a great quest to find out relations between Covid-19 and chemicals that they consume on a daily basis, as well as the to find out the relations between Covid-19 and genes that this disease is related to. Therefore, the aim of this seminar work is to classify relations between biomedical entities (chemicals, genes and diseases), to make a relation prediction on unlabeled sentences and visualize relations between Covid-19 and all chemicals and genes that it is related to, using Knowledge Graphs. This work, has achieved enviable results, and I hope that it could be useful for world's population.

2 DATA

The dataset used for the purposes of this seminar work, for training and evaluating the model, is publicly available on Zenodo website (https://zenodo.org/record/1035500#.YIYLWOgzZPa).

This repository contains labeled, weighted networks of chemical-gene, gene-gene, gene-disease, and chemical-disease relationships based on single sentences in PubMed abstracts.

The Zenodo website contains two types of files for each biomedical relationship type (chemical-gene, gene-disease, chemical-disease and gene-gene), which are used for training the classification model. The first file connects dependency paths to labels, or "themes". The number of labels or "themes" for chemical-disease relationship is 7, for chemical-gene is 10, for gene-disease is 10 and for gene-gene 9. The labels for each relationship type are also shown on the Zenodo website (https://zenodo.org/record/1035500#.YIYLWOgzZPa), as well as their raw labels. Each record in the first file contains a dependency path followed by its score for each theme, and indicators of whether or not the path is part of the flagship path set for each theme (meaning that it was manually reviewed and determined to reflect that theme). The dependency path for each record looks as follows:

bleeding|nmod|start_entity bleeding|nmod|users users|nmod|inhibitors inhibitors|compound|end_entity.

The second file connects sentences to dependency paths. It consists of sentences and associated metadata, entity pairs found in the sentences, and dependency paths connecting those entity pairs. Each record in the second file contains the following information:

- PubMed ID
- Sentence number (0 = title)
- First entity name, formatted
- First entity name, location (characters from start of abstract)
- Second entity name, formatted
- Second entity name, location
- First entity name, raw string
- Second entity name, raw string
- First entity name, database ID(s)
- Second entity name, database ID(s)
- First entity type (Chemical, Gene, Disease)
- Second entity type (Chemical, Gene, Disease)
- Dependency path
- Sentence, tokenized

Dependency paths are crucial part in this data files, which connects two entities, as well as they are analyzing the grammatical structure of the sentence based on dependencies between words. They are produced automatically using the Stanford dependency parser [4]. The input to the parser is a raw Medline sentence, and the output is a dependency graph. A dependency graph shown on Figure1 is one way to represent the grammatical architecture of a sentence; the nodes are words, and the edges are grammatical dependencies (grammatical relationships between pairs of words, described in detail in the paper [5]).

A dependency path is a path through a dependency graph that connects two entities. Focusing on the dependency path helps prune out irrelevant terms and phrases and focus the algorithm's attention on the part of the sentence directly relevant to the relationship between the two entities. It is possible for a single sentence to generate multiple dependency paths if more than two entity names are present in the sentence.

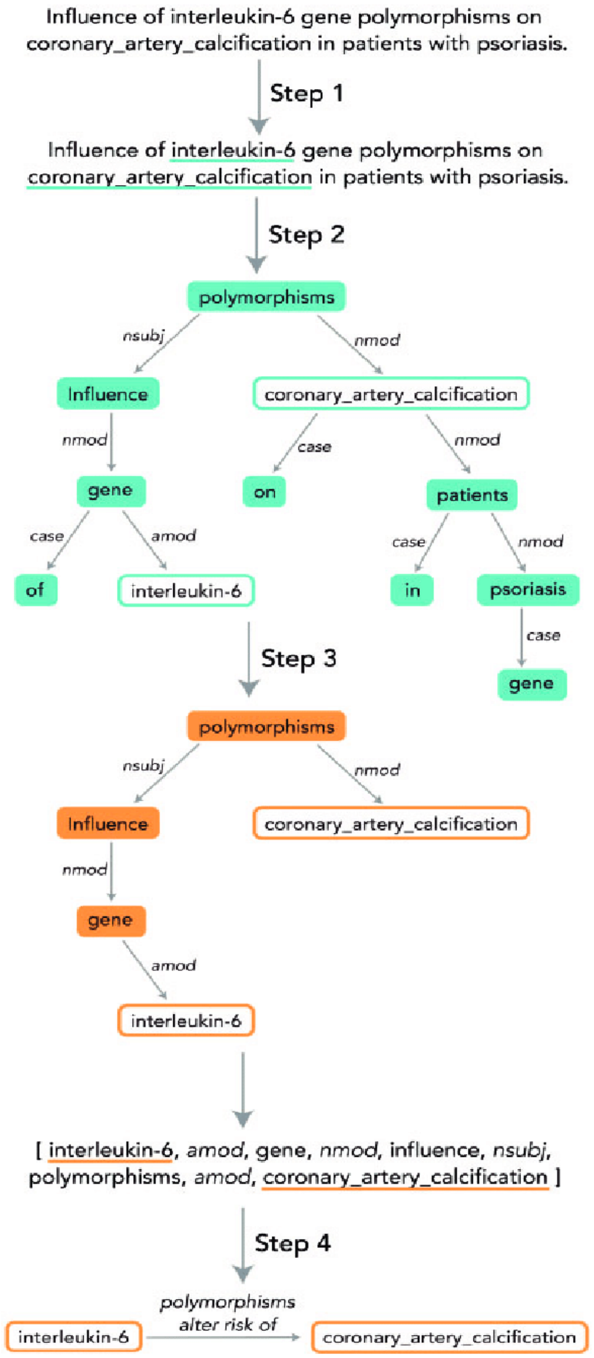
Influence of interleukin-6 gene polymorphisms on coronary_artery_calcification in patients with psoriasis.
Step 1
Influence of interleukin-6 gene polymorphisms on coronary_artery_calcification in patients with psoriasis.
Step 2
polymorphisms
nsubj
nmod
Influence
coronary_artery_calcification
nmod
case
nmod
gene
on
patients
case
amod
case
nmod
of
interleukin-6
in
psoriasis
case
gene
Step 3
polymorphisms
nsubj
nmod
Influence
coronary_artery_calcification
nmod
gene
amod
interleukin-6
[interleukin-6, amod, gene, nmod, influence, nsubj, polymorphisms, amod, coronary_artery_calcification]
Step 4
polymorphisms alter risk of
interleukin-6
coronary_artery_calcification

Figure 1. Process of converting a sentence to structured relationship [6]

On the Table 1 below are shown the number of records (instances) in both data files, for each relationship type which are used for training and evaluating the relation classification model. Whereas, Figure2 presents an architecture diagram of all steps taken in this seminar work, which will be clearly described later in this paper.

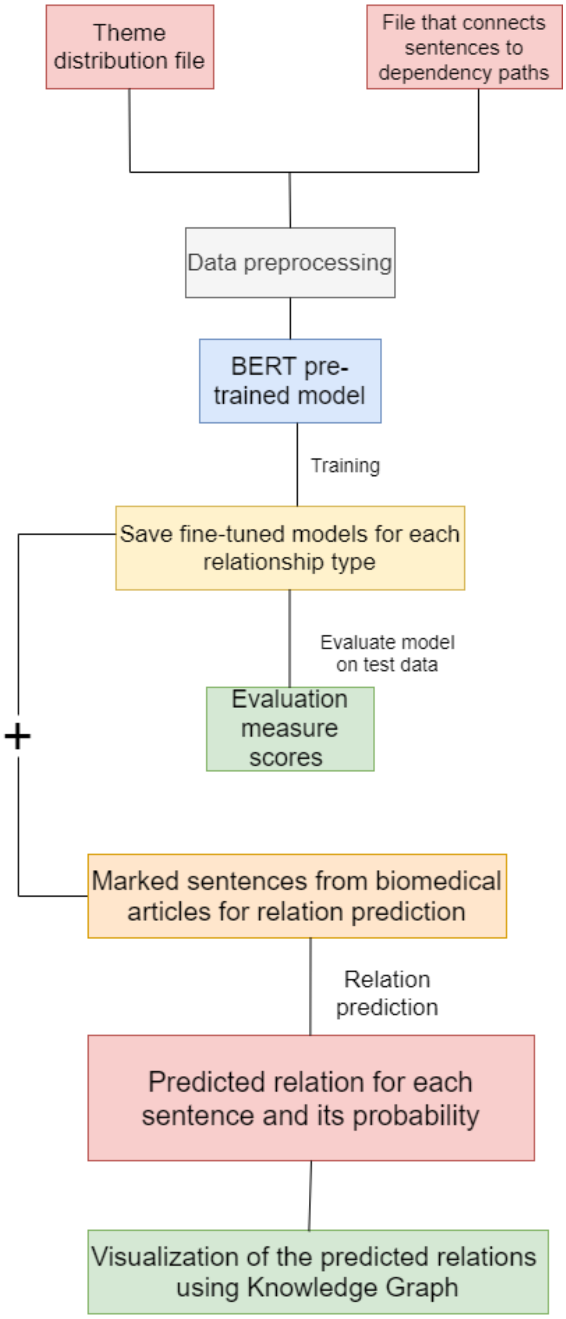
Theme distribution file
File that connects sentences to dependency paths
Data preprocessing
BERT pre-trained model
Training
Save fine-tuned models for each relationship type
Evaluate model on test data
Evaluation measure scores
+
Marked sentences from biomedical articles for relation prediction
Relation prediction
Predicted relation for each sentence and its probability
Visualization of the predicted relations using Knowledge Graph

Figure 2. Architecture design of the seminar work

Relationship type	Number of instances in the theme distribution file	Number of instances in the file that connects sentences to dependency paths
Chemical-disease	2166529	4302710
Chemical-gene	765928	1572376
Gene-gene	1635545	4240702
Gene-disease	1867278	3469081

Table 1. Number of instances in data files for each relationship type

3 Data preprocessing

Before building a model architecture and train it, it is very important to prepare the raw data because Machine Learning models do not work with raw text. The main piece of functionality that I need for data preparation with BERT is how to tokenize inputs and convert them into their corresponding ids in BERT's vocabulary. At the moment, the Hugging Face library (https://huggingface.co/) seems to be the most widely accepted and powerful PyTorch interface for working with BERT. In addition to supporting a variety of different pre-trained transformer models, the library also includes pre-built modifications of these models suited to your specific task. Huggingface has added very nice functionality to both the BertModel and BertTokenizer class where we can just put in the name of the pre-trained model we want to use. In this seminar work it is the 'bert-base-cased' model which works better than bert-base-uncased and consists of 12 Transformer blocks (layers), 768 hidden vectors, 12 self-attention heads and 109M parameters in total .

After downloading both files for each biochemical relationship type (chemical-disease, chemical-gene, gene-disease, gene-gene) from Zenodo website, I have built a Data Loader. Data Loader is a client application for the bulk import or export of data. Use it to insert, update, delete, or export salesforce records. From the first file, which contains dependency path and the score for each label ('theme') I have extracted the highest score of each instance (dependency path). From the second file, that contains tokenized sentences and connects them with dependency, I have extracted the tokenized sentences and merge both files (data frames) on dependency path.

The next step is to encode labels using preprocessing.LabelEncoder() class that encodes target labels with value between 0 and num_classes -1. Next I map labels into themes. Words in the sentences (strings) must be converted into a single integer id, using the vocabulary and convert_tokens_to_ids() function. Before I train the model it is important to split data into train and test data, where x-array are ids of the tokenized sentences and y-array are labels or "themes" of the sentences.

4 Training and Evaluating Classification Model

Relation classification, the task of this seminar work, is a well known task in NLP. It classifies relations that occur between two biomedical entities (chemicals, genes or diseases) in sentences by assigning a label from a pre-defined set of abstract relation labels.

Huggingface (https://huggingface.co/) is the most well-known library for implementing state-of-the-art transformers in Python. It offers clear documentation and tutorials on implementing dozens of different transformers for a wide variety of different tasks. For training the model in this seminar work I am using Pytorch framework. The model used for training in this seminar work is BERT [2] model, which stands for Bidirectional Encoder Representations from Transformers and it is used to train a classifier. It is designed to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context. As a result, the pre-trained BERT model can be fine-tuned with just one addition output layer to create state-of-the art models for wide range of NLP tasks. BERT is a transformers model pretrained on a large corpus of English data in a self-supervised fashion. This means it was pretrained on the raw texts only, with no humans labelling them in any way (which is why it can use lots of publicly available data) with an automatic process to generate inputs and labels from those texts. The Transformer is the basic building block of most current state-of-the-art architectures of NLP. Its primary advantage is its multi-head attention mechanisms which allow for an increase in performance and significantly more parallelization than previous competing models such as recurrent neural networks.

The tokenizer object allows the conversion from charater strings to tokens understood by different models. Each model has its own tokenizer, and some tokenizing methods are different across tokenizers. For the tokenizer, I used 'bert-base-case' version of BertTokenizer, that consists of 12 Transformer blocks (layers), 768 hidden vectors, 12 self-attention heads and 109M parameters in total. As well as, I am using BertForSequenceClassification library, that inherits from PreTrainedModel, to construct the model. This model is also a PyTorch subclass. This is the normal BERT model with an added single linear layer on top for classification that I will use as a relation classifier. As I feed input data, the entire pre-trained BERT model and the additional untrained classification layer is trained on the specific task.

The replacement optimization algorithm for stochastic gradient descent used for training the model is Adam. Adam combines the best properties of the AdaGrad and RMSProp algorithms to provide an optimization algorithm that can handle sparse gradients on noisy problems. In PyTorch we need to set the gradients to zero before starting backpropagation because PyTorch accumulates the gradients on subsequent backward pass. The learning rate is 1e-5. The data files from Zenodo website are split so that, 70% of the instances of each relationship type are for training and 30% for evaluation (test) the model. The number of epochs is 10 and the training batch size is 128, whereas evaluation batch size is 64. When training a model using PyTorch framework, in each epoch there are two passes: forward pass and backward pass. The forward pass computes predicted outputs (labels) by passing input to the model, which are ids of the sentences. On the other hand, backward pass (backpropagation) compute gradient of the loss with respect to all the learnable parameters of the model. Then I need to update the weights (parameters) using gradient decent with optimizer.step() function.

This model architecture, with the same parameters, was trained four times, for each relationship type (chemical-gene, chemical-disease, gene-disease and gene-gene) separately and according to that each fine-tuned model was saved, so I could lately evaluate the model and make predictions on unseen sentences.

For evaluating the model on test data I have calculated the following multi class evaluation metrics: macro f1 score, weighted f1 score, accuracy, as well as, precision ,recall and f1-score for each label of the relationship types. On Table 2 below are shown results of the evaluation on test data, such as: accuracy score, macro f1 score and weighted f1 score for each relationship type. The class distribution for each relationship type is imbalanced, so that's why I have chosen weighted f1 measure and it gives good scores for each relationship type. The f1 scores are calculated for each label for each relationship type and then their average is weighted by support - which is the number of true instances for each label.

Fine-tuned model	Accuracy on test data	Macro f1 score	Weighted f1 score
Chemical-disease	74.37 %	72.57%	74.66%
Chemical-gene	75.068%	70.08%	75.71%
Gene-disease	72.396%	50.19%	70.55%
Gene-gene	81.04%	77.99%	81.89 %

Table 2. Evaluation measures scores

5 Relation Predictions

After training the model on train data and evaluating on test data for each relationship type separately, I have done a relation prediction and it's probability on new, unseen sentences from biomedical articles. Relation prediction is the processes of using a saved, fine-tuned model to make predictions for the relation theme (label) between two entity types (chemical, gene or disease) for a given sentence (unseen data) from biomedical article and retrieve the probability for the predicted relation. In paper [6], authors revealed 10 broad themes for chemical-gene relations, 7 for chemical-disease, 10 for gene-disease and 9 for gene–gene in Medline abstracts.

The dataset that I used for making predictions is available on kaggle website and it contains sentences which are extracted from biomedical articles and contain more than two biomedical entities. The task is to predict the relation between two entitites for each sentence and retrieve the probability for the predicted relation. The number of instances (sentences) that this dataset consists is 40813. This dataset is in .csv format and consists of five columns, where the first column is the start entity (it can be any chemical, gene or disease), the second column is end entity that also can be some gene, chemical or disease, third and fourth column consist the type of the start and end entity (chemical, gene or disease) and the last column has the marked sentence from a Medline papers, for which sentence I have to predict the relation between two given entities using the saved, fine-tuned model for the appropriate relationship type. As we cannot confirm order of entities (for example chemical-disease or disease-chemical), I predict two possibilities, initial prediction and reverse prediction, and retrieve the higher one.

Before making predictions it is necessary to prepare prediction data (same as I prepared train and test data), which includes converting tokens into ids, using BertTokenizer and loading saved, fine-tuned models and predicting two probabilities: initial prediction and reverse prediction, and retrieve the higher one.

I have made relation prediction on the marked sentences, for each relationship type (chemical-gene, chemical-disease, gene-disease and gene-gene) and save them in four different .csv data files, so I could lately visualize them using Knowledge Graphs.

6 Visualization of the predicted relations using Knowledge Graphs

Human knowledge provides a formal understanding of the world. Therefore, from my own point of view, the most suitable way of representing the predicted relations predictions for each relationship type, from the previous section, are knowledge graphs (KGs) [7-9]. Knowledge graphs that represent structural relations between entities have become an increasingly popular research direction towards cognition and human-level intelligence. They are structural representations of relations between real-world entities where relations are defined as triplets containing a head entity, a tail entity, and the relation type connecting them. KG based information retrieval has shown great success in the past decades. Knowledge graphs are graph-structured knowledge bases, in which facts are represented as relations (edges) between entities (nodes). Resource Description Framework (RDF) is a common way of representing knowledge graphs. RDF defines relationships in the form of triplets comprising head entity, relation, and tail entity. Biochemical networks can be treated as knowledge graphs, where vertexes are the entities, and edges are the relations from head entity to tail entity. Thus, a link prediction task in biochemical networks is equivalent to a knowledge graph completion task.

A graph is made of two sets — the set of nodes (also called vertices) and the set of edges (also called arcs). Each edge itself connects a pair of nodes indicating that there is a relation between them. Graphs can be either homogeneous or heterogeneous. In a homogeneous graph, all the nodes represent instances of the same type and all the edges represent relations of the same type. The KGs that are constructed in this seminar work are heterogeneous, because nodes represent instances of two different types of entites and edges (relations) are different types. As well as, nodes and relation types have domain specific semantics, which is biomedical.

By showing the predicted relations using Knowledge Graph, I got a visual representation of the relations between biochemical entities (chemicals, genes and disease). Therefore, by this representations of the predicted relations we can easily show relations between a given entity, for example Covid-19, and other entities that this disease is related to. For example if we are interested into a relation between Covid-19 disease and chemical Daptomycin, which is chemical-disease relationship type, we should look at the knowledge graph (Figure3) that shows relations between Covid-19 and chemicals and see what is the predicted relation between that two entities. If we hold the coursor on the edge that connects those two entities that we want to know relation, KG will show what is the predicted relation between those entities. On the Figure4 below we can see that the predicted relation between Covid-19 and Daptomycin is ‘C’ label which means ‘inhabits cell grow’. Figure5 shows Knowledge Graph which represents predicted relations between Covid-19 and genes that this disease is related to. Also we could visualize KG, according to our queries related to relations between some entitites. For example, if we want to know which genes have “Mutations that affect disease course” (label Ud) and “Polymorphysms alter risk” (label Y) to Covid-19, we can easily do that by extracting (from the predicted relations between gene-disease relationship type) all sentences that have Covid-19 disease as an entitiy and where predicted relations are “Ud” or “Y”. The result from this query, visualized with KG is shown on Figure6. We could notice that KG on Figure6 is much smaller than KG on Figure5, because in the KG on the Figure6 we extracted only those entities (genes) that have “Mutations that affect disease course” (label Ud) and “Polymorphysms alter risk” (label Y) on Covid-19 disease. In the same way we can construct many KGs, according to our needs and interests related to any disease, chemical or gene and their relations.

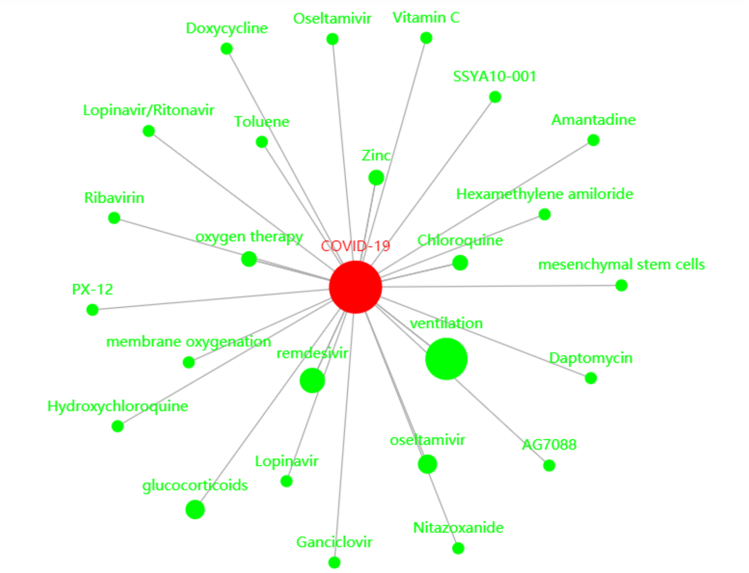

Figure 3. Chemical - Covid19 Knowledge Graph

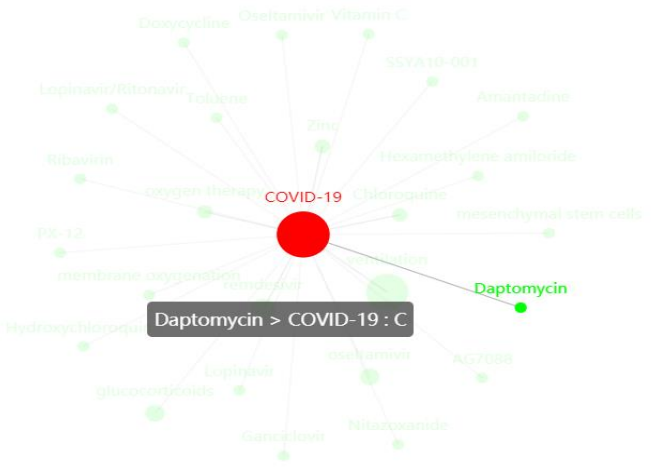

Figure 4. Predicted relation between Covid-19 and Daptomycin shown on KG

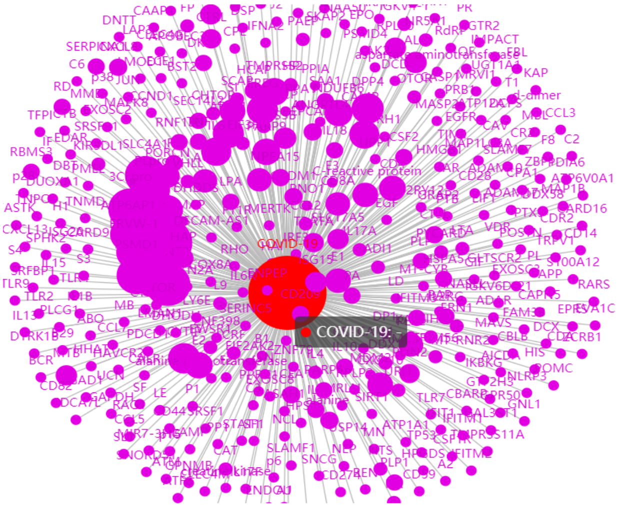

Figure 5. Gene - Covid-19 Knowledge Graph

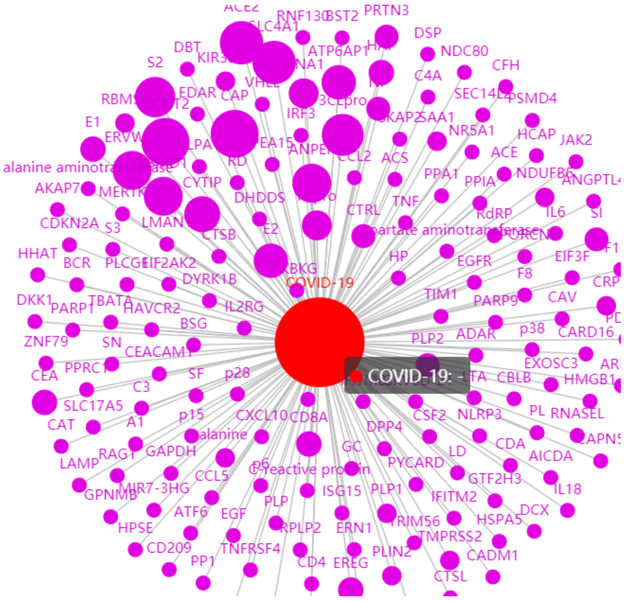

Figure 6. Genes that have "Mutations affect disease course" and "Polymorphisms alter risk" with Covid-19

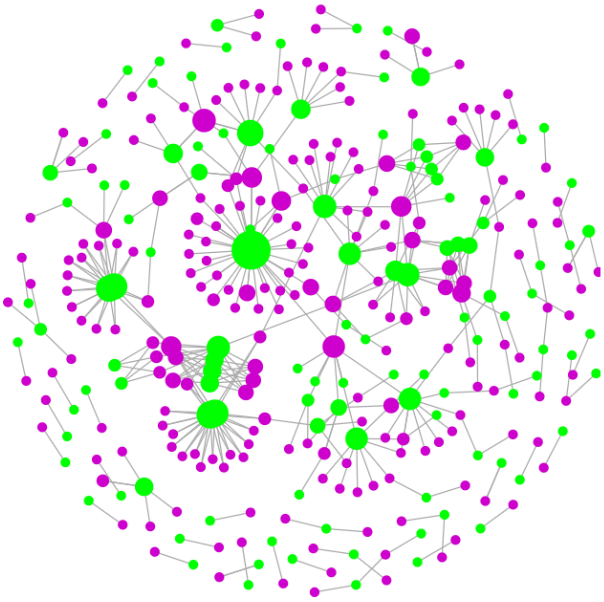

Figure 7. Chemical-Gene Knowledge Graph

7 Conclusion

In this paper I proposed a relation classification system, which classifies relations between biomedical entities, such as chemicals, genes and diseases. The model architecture used for classifying relations is pre-trained BERT model. The proposed seminar work demonstrates that with a pre-trained BERT model you can quickly and effectively create a high quality model using the pytorch interface, regardless of the specific NLP task you are interested in.

Using the saved, fine-tuned models for each relationship type, I have also proposed a relation prediction on unseen sentences from biomedical articles. This step predicts the relation between two biomedical entities, that could be chemical, gene or disease, from unseen sentences extracted from biomedical articles and retrieve the probability for the predicted relation. To be more challenging, useful and visually representative, I have visualized the predicted relations using Knowledge Graphs. The emphasis is on visualizing the predicted relations between Covid-19 disease and chemicals and genes that this disease is related to. We could definitely agree that KG helps a lot into visualizing predicted relations between entities and gives us better understanding into domain specific semantics that we are analyzing. In this paper it is biomedical domain, where the main emphasis is on Covid-19 disease and it's relations with other entities such as chemicals and genes. Besides that, KGs have gradually become an important resource for many knowledge-driven applications, such as semantic search, question answering and recommender systems.

References

[1] ShanchanWu, Yifan He, Enriching Pre-trained Language Model with Entity Information for Relation Classification

[2] Jacob Devlin Ming-Wei Chang Kenton Lee Kristina Toutanova, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, 1810.04805v2, 24 May 2019

[3] Todd L. Cherry, Alexander G. James, James Murphy, The impact of public health messaging and personal experience on the acceptance of mask wearing during the COVID-19 pandemic, Journal of Economic Behaviour and Organization, May 2021

[4] De Marneffe,M.C. and Manning,C.D. (2008a) The Stanford typed dependencies representation. In: COLING Workshop on Cross-Framework and Cross-Domain Parser Evaluation, pp. 1–8.

[5] De Marneffe,M.C. and Manning,C.D. (2008b) Stanford typed dependencies manual. Technical report. Stanford University; 2008.

[6] Bethany Percha, Russ B. Altman, A global network of biomedical relationships derived from text, Bioinformatics, 34(15), 2018, 2614–2624

[7] A. Singhal. (2012). Introducing the Knowledge Graph: Things, not Strings. [Online]. Available: https://googleblog.blogspot.com/2012/05/ introducing-knowledge-graph-things-not.html

[8] L. Ehrlinger andW.Wöÿ, ``Towards a de_nition of knowledge graphs,'' in Proc. Joint Posters Demos Track 12th Int. Conf. Semantic Syst. (SEMANTiCS),
1st Int. Workshop Semantic Change Evolving Semantics (SuCCESS) Co-Located 12th (SEMANTiCS), 2016, p. 4. [Online]. Available: http://ceur-ws.org/Vol-1695/paper4.pdf

[9] J. Yan, C. Wang, W. Cheng, M. Gao, and A. Zhou, ``A retrospective of knowledge graphs,'' *Frontiers Comput. Sci.*, vol. 12, no. 1, pp. 55_74, Feb. 2018.