A Deep Learning Approach to Identifying Missing Hierarchical Relations in SNOMED CT

Rashmie Abeysinghe[1], Olivier Bodenreider, MD, PhD[2]
[1]Department of Computer Science, University of Kentucky, Lexington, KY
[2]Lister Hill National Center for Biomedical Communications, National Library of Medicine, National Institutes of Health, Bethesda, MD

Abstract

Inconsistencies such as missing hierarchical relations in terminologies like SNOMED CT could be problematic to downstream applications that use these terminologies as knowledge sources. Structural-lexical methods based on non-lattice subgraph have been recently introduced to identify hierarchical relation inconsistencies. Such methods are rule-based and only a subset of non-lattice subgraphs exhibit these patterns. In this work, we explore whether such rule-based methods could be replaced by a deep learning approach. Our aim is to train a neural network that predicts hierarchical relations between concepts. To train the model we employ hierarchical relations and non-relations that exist in SNOMED CT. After training, we apply the model to SNOMED CT non-lattice subgraphs so that missing hierarchical relations can be identified.

1 Introduction

SNOMED CT is the largest clinical terminology in the world. Quality issues that exists in SNOMED CT could propagate to downstream applications that use it[1]. Therefore, auditing SNOMED CT is necessary to make sure that it produces an accurate representation of its clinical content. Due to the size and the structural complexity of SNOMED CT, manually reviewing is impractical to audit the terminology. Therefore, automated approaches are widely investigated to perform quality assurance efficiently and effectively.

In this work, we introduce a deep learning-based approach to identify missing hierarchical (is-a) relations in SNOMED CT. We aim to not only identify the existence of a missing is-a, but also identify the direction of the relation. Recently, there have been a number of investigations that take into account the lexical features of concept labels in graph fragments known as Non-Lattice Subgraphs that are extracted in terminologies[2–4]. These approaches leverage lexical patterns that are identified between concepts in non-lattice subgraphs to suggest missing is-a relations. The patterns have been manually curated based on observations researchers have made in NLSs. The patterns introduced do not cover all the NLSs. Hence, lexical pattern-based approaches are neither exhaustive, nor sustainable. The goal of this work is to leverage dep learning techniques to automatically identify missing is-a relations in NLSs from SNOMED CT.

2 Background

2.1 SNOMED CT

Maintained and distributed by SNOMED International, SNOMED CT is the largest, most comprehensive clinical healthcare terminology in the world containing around 350,000 concepts. SNOMED CT covers clinical medicine, including findings, diseases, and procedures for use in electronic medical records[7]. It has 19 top-level sub-hierarchies including Clinical finding, Procedure, Body structure etc. The aim of SNOMED CT is to improve patient care through the development of systems to record health care encounters accurately[8]. Importantly, SNOMED CT enables consistent, processable representation of clinical content in Electronic Health Records (EHR)[6].

2.2 Non-Lattice Subgraph-based Structural-Lexical Auditing Methods

Being a lattice is considered as a desirable property for a well-formed terminology[10]. A terminology is a lattice if any pair of concepts in the terminology has a unique maximal shared descendant and a unique minimal shared ancestor. If a concept pair has more than a single maximal shared descendant (or a single minimal shared ancestor), it is known as a non-lattice pair[?, 10], which may disclose quality issues in terminologies.

Examining multiple non-lattice pairs that share the same maximal shared descendants separately is time-consuming

and may involve a significant amount of redundant work. Therefore, Non-Lattice Subgraphs (NLSs) have been introduced[2] to inspect them together. An NLS can be acquired by a non-lattice pair (c_1, c_2) as follows. Firstly, maximal common descendants of the non-lattice pair, $mcd(c_1, c_2)$, named as the lower bounds, is computed. Then, the minimal common ancestors of the lower bounds, $mca(mcd(c_1, c_2))$, named as the upper bounds, is computed. Finally, all the concepts as well as relations between (and including) lower and upper bounds is aggregated to generate the NLS. The size of an NLS is the number of concepts it contains.

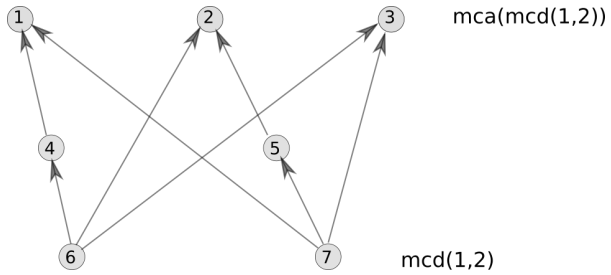

Figure 1: An example of an NLS. Nodes of the graph are concepts. The edges indicate hierarchical IS-A relations where the arrowheads point to the parent concept.

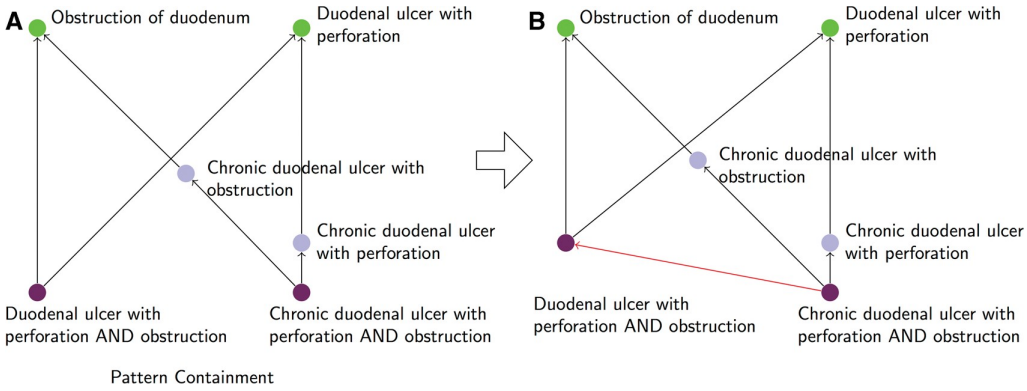

Figure 2: An NLS in SNOMED CT exhibiting Union lexical pattern[2].

NLSs have been utilized to effectively identify defects in biomedical terminologies. Cui et al.[2] have proposed four lexical patterns found in NLSs which suggest missing hierarchical relations and missing concepts in SNOMED CT. Figure 2 displays an NLS which exhibits a lexical pattern called "Containment" that they introduced. Here they suggest a missing relation between the two lower bound concepts based on the fact that, the set of words of the lower bound concept *Duodenal ulcer with perforation AND obstruction* is contained in the lower bound concept *Chronic duodenal ulcer with perforation AND obstruction*. The modifier "Chronic" makes the latter concept more specific than the former, and hence a missing relation in the form of *Chronic duodenal ulcer with perforation AND obstruction* is-a *Duodenal ulcer with perforation AND obstruction* is suggested. Cui et al.[2] have introduced two more lexical patterns: "Intersection" which suggests missing relations in upper bound and "Union" which suggests missing relations in lower bound.

Lexical patterns in NLSs such as "Containment" in Figure 2 are useful in not only identifying inconsistencies, but also in suggesting remediation measures to fix the inconsistency. However, these patterns have been manually curated based on observations that researchers have made in NLSs. Manual curation of the patterns is time consuming. Some complex lexical patterns may not be identified during manual curation at all. Furthermore, only a smaller percentage of NLSs have exhibited the lexical patterns that have been introduced by Cui et al[2]. Hence, the lexical pattern-based approaches are neither exhaustive, nor sustainable.

3 Methods

In this work, we investigate a deep learning approach to automatically learn from existing relations in terminologies to identify potential missing relations in its NLSs. We apply our approach to the Clinical Finding subhierarchy of the March 2019 US edition of SNOMED CT.

3.1 The architecture of the model

A Graph Neural Network (GNN) architecture is used in this project to address the challenges posed by irregularities in graph data such as each node having a variable number of neighbors, and the inability of the traditional neural network architectures to cope with such data[9, 11]. GNNs learn features for each node in a graph by performing neighborhood aggregation: nodes aggregating information from their neighbors. Based on neighborhood aggregation scheme, there exists different types of GNNs. We used Graph Convolutional Network (GCN), a type of GNNs in our work[12]. Architecture of our model is given in Figure 3. We have two GCN layers followed by three Fully-Connected (FC) layers in our model. Input to the first GCN layer is a graph with feature embeddings for each of its nodes. Output of the second GCN layer is also a graph with the same structure, but different node embeddings that are learnt by neighborhood aggregation at both GCN layers. The feature embeddings of the pair of concepts where we are training to predict the existence/non-existence of a relation is then concatenated and passed on to the first FC layer. The concatenation is done so that always the child is followed by the parent. The output of the final FC layer is passed through a Sigmoid function. The output of the sigmoid is a probability of the concept pair having a relation. In the training phase, the error will be calculated based on the output probability and the expected probability (positive:1, negative:0) and backpropogated to update the weights of the model. In the validation, testing and application phases, we consider a probability above or equal to 0.5 to denote the existence of an is-a relation while a probability below 0.5 will denote the nonexistence of an is-a relation.

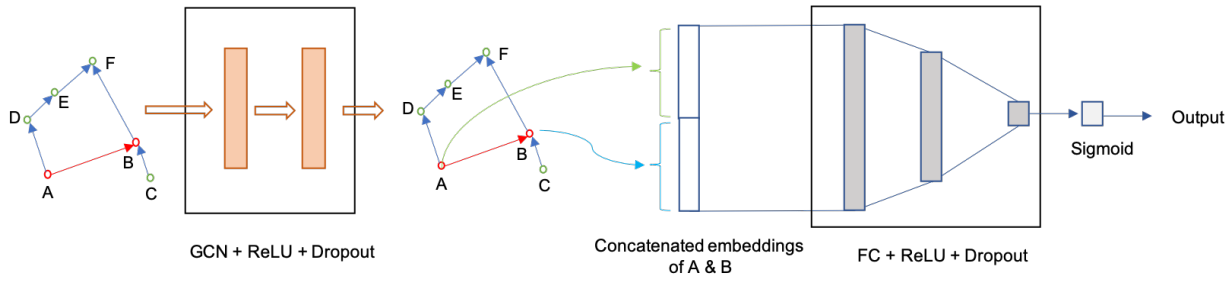

Figure 3: The architecture of our GNN-based classifier.

3.2 Sample generation

Since this is a supervised learning task, the model has to be trained using labeled data. We generate samples from existing relations (concept-pairs that has an is-a relation, considered as positive samples) and existing non-relations (concept-pairs that does not have an is-a relation considered as negative samples) of SNOMED CT. Note that our negative samples does not contain non-relations in NLS lower and upper bounds as these are where, after training, we expect to apply the trained model to predict missing relations. For each concept in a sample, we define a context around the concept with its ancestors and descendants up to n levels. We set $n = 2$ in this work. The concept-pair of the sample together with their contexts generates a subgraph from the terminology. Figure 4 denotes a subgraph obtained for the positive sample concepts: *Toxic shock syndrome caused by methicillin susceptible Staphylococcus aureus* and *Staphylococcal toxic shock syndrome*. Similarly Figure 5 denotes a subgraph obtained for the negative sample concepts: *Brucellosis of skin* and *Cutaneous diphtheria*. However, for negative samples we further process the subgraphs as follows. First, we artificially introduce an is-a relation between the concept-pair. This is important so that the positive and negative samples will have the same link existence information and the classifier will not optimize on this part of information to classify. Because of this step, each sample irrespective of negative or positive, has a child concept and a parent concept. For the negative samples, which concept is the child is chosen randomly. Next, we will

remove any relations in the subgraph that could be inferred with the introduction of the above mentioned is-a relation. Such redundant relations does not exists in the terminology and hence, the classifier may optimize on this information to separate negative samples from positives. Figure 6 denotes the resultant subgraph after the two processing steps.

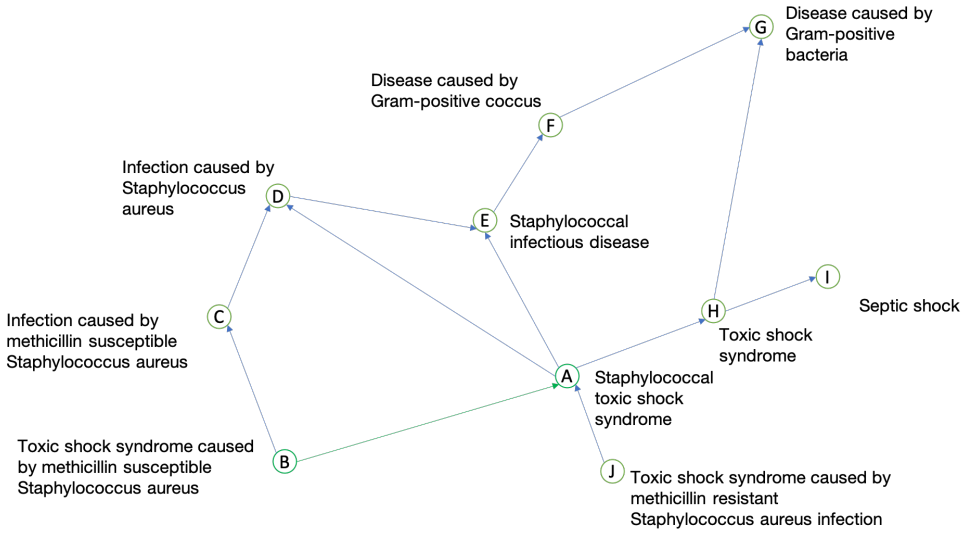

Figure 4: Subgraph formed by positive sample concepts: *Toxic shock syndrome caused by methicillin susceptible Staphylococcus aureus* and *Staphylococcal toxic shock syndrome* together with the concepts in their context.

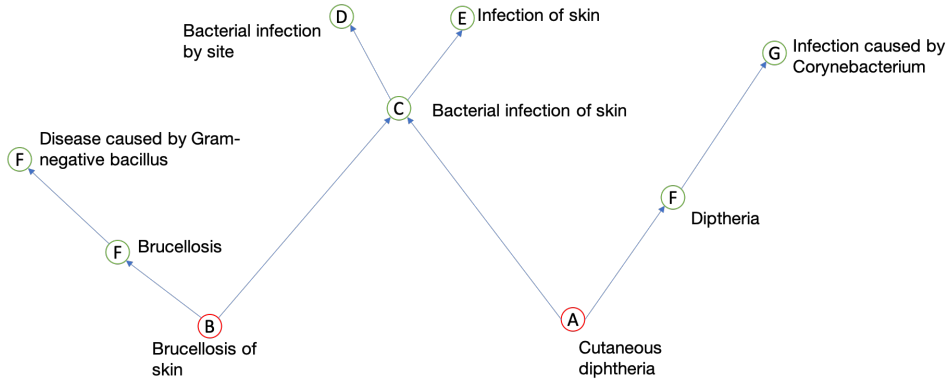

Figure 5: Subgraph formed by negative sample concepts: *Brucellosis of skin* and *Cutaneous diphtheria* together with the concepts in their context.

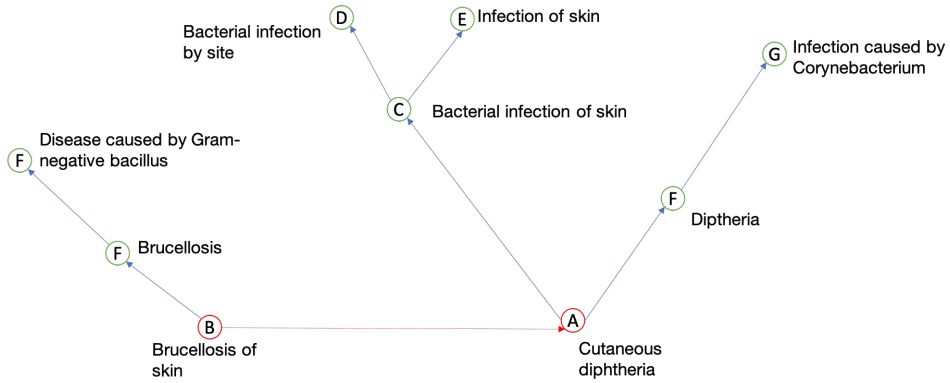

Figure 6: Subgraph in Figure 5 after artificial edge introduction and redundant edge removal.

There exist 117,625 concepts in the Clinical Finding subhierarchy of SNOMED CT. These concepts are connected by 210,349 is-a relations. Like most terminologies, SNOMED CT is a sparse graph, which means the number of relations is much smaller than the number of non-relations. Because of this, our data set is imbalanced. However, most non-relations may be uninteresting since we will not be seeing such non-relations in NLSs. Since we train this model to eventually predict missing relations in NLSs, only non-relations types that could occur in NLSs needs to be used as negative samples. To identify the type of the non-relation, we employ a measure known as "edge-separation". Simply put, for two concepts in a non-relation, the edge-separation is the sum of edges from each concept to a common parent. For example, for siblings, edge-separation is two, and for uncle-nephew pairs, the edge-separation is three. We compute the edge-separation for all non-relations in the upper and lower bounds of NLSs and based on this distribution, we randomly pick 210,349 negatives so that they follow the same distribution in terms of edge-separation. Since, our model needs to identify the direction of the relation as well, we reverse the concept-pairs in the positive samples to generate another set of negatives. i.e. if a is-a b in the positive sample, then we generate b is-a a as a negative. By this we expect the model to learn the correct direction of the relation. Therefore, altogether we have 631,047 samples generated. Then, we randomly pick 90% of the samples (567,942) for cross validation and the rest as a separate holdout testing set (63,105).

3.3 Embeddings for preferred terms

As mentioned above, each node in the graph passed as the input to the first GCN layer, should have a feature embedding to represent it. Each node in our input graph is a SNOMED CT concept and we obtain the feature embeddings for it from its preferred name. Note that the preferred name has also been used with lexical patterns[2]. To do this, we separately train a Doc2Vec model to obtain embeddings for the preferred names of every concept in SNOMED CT. Doc2Vec is an unsupervised framework that learns fixed-length feature representations from variable-length pieces of texts[13]. We set the length of the embeddings as 150, and train the Doc2Vec model for 20 epochs. These embeddings will be assigned to the corresponding node of the graph before feeding into the first GCN layer.

3.4 Training the model

We used the python package Deep Graph Library[14] to implement our model. The model was trained using the computation resources of the NIH HPC Biowulf cluster[15]. Particularly, we used an NVIDIA Tesla K20X GPU for training the model. We used Binary Cross Entropy as the loss function and Adam as the optimizer. A learning rate of 0.001 and a batch size of 128 was used. We performed 6-fold cross validation and the training was performed for 100 epochs.

3.5 Applying the trained model to NLSs

We pick small (sizes 3, 4, and 5) NLSs to apply our trained model. Small NLSs were used in earlier lexical-pattern-based works as they are contained in larger NLSs and also easier to be reviewed by domain experts[2]. We obtain all the non-relations that exists between upper and lower bound concepts of these smaller NLSs. Then, we generate subgraphs for these non-relations similar to how we generated subgraphs for negative samples. That is by including all concepts in the contexts of the concept pair, introducing the artificial edge and then removing redundant relations. Then, this subgraph is passed on to the model which will classify the input to either a relation or a non-relation.

4 Results

The performance of our model is given in Table 1. As mentioned earlier, we performed 6-fold cross validation on 567,942 samples and measured the performance on a separate holdout set with 63,105 samples.

Tables 2, 3, 4, and 5 displays true positives, true negatives, false negatives, and false positives respectively obtained by the model from the holdout set. In using them for training, we assume the relations and non-relations in the training sample to be accurate. However, there may be cases in these examples that the SNOMED CT relation/non-relation may be incorrect. Such cases can only be identified through a review by a domain expert.

There exist 7,141 small NLSs in the Clinical Finding subhierarchy of SNOMED CT. These NLSs have 37,404 non-

Table 1: The performance of our model.

Type	Cross validation	Holdout set
Precision	0.8629	0.8439
Recall	0.8475	0.8291
F1 score	0.8552	0.8364

Table 2: True positives: relations identified correctly.

Child	Parent
Duodenal papilla not found	Digestive system finding
Cyproterone adverse reaction	Antineoplastic adverse reaction
Mosaic trisomy 5 syndrome	Anomaly of chromosome pair 5
Acute hepatitis	Inflammatory disease of liver

Table 3: True negatives: non-relations identified correctly.

Child	Parent
Chronic gingivitis	Chronic fibrous gingivitis
Finding of urine drug level	Acetaminophen in urine
Open fracture of proximal phalanx of left thumb	Closed fracture thumb proximal phalanx, head
Dextrotransposition of aorta	Transposition of aorta

Table 4: False negatives: Relations not identified.

Child	Parent
Traumatic arthropathy of metacarpophalangeal joint	Traumatic arthropathy of the hand
Primary basal cell carcinoma of right lower limb	Basal cell carcinoma of lower extremity
Pulmonary aspiration of gastric contents	Pulmonary aspiration of fluid
Lesion of radial nerve	Radial neuropathy

relations in their lower and upper bounds. Applying the trained model on non-relations in upper and lower-bounds of these NLSs, our model identified 11,943 (31.93%) missing is-a relations. We compared the results obtained by our model with the missing is-a detected by the lexical patterns “Containment”, “Union”, and “Intersection” to see how much our model captures the results obtained by the patterns.. The results of this comparison is displayed in Table 6. As it can be seen, in total the model only identifies 53.21% of the missing is-a detected by the lexical patterns.

5 Discussion

Even though our model has good performance with regard to the holdout set, it seems that performance does not transcend to NLSs. Even though we did not perform and evaluation of our results by a domain expert, the fact that

Table 5: False positives: relations identified incorrectly.

Child	Parent
Implanted defibrillator generator infection	Infected pacemaker
Open fracture of right angle of mandible	Open fracture of zygoma
Finding of cochlear function	Cochlear microphonic
Aluminum hydroxide overdose	Kaolin overdose

Table 6: Comparison of our results with lexical patterns in Cui et al[2].

Lexical pattern	Num. of missing is-a obtained by lexical pattern	Num. of common missing is-a obtained by our model
Containment	406	150
Union	110	43
Intersection	542	370

our model identifies 31.92% of non-relations in upper and lower bounds of NLSs as missing is-a suggests that the model provides a lot of false positives leading to a lower precision. The fact that it only identifies 53.21% of the missing is-a relations identified by the lexical patterns suggests that the model has a lower recall. Therefore, this approach based on GNNs does not achieve the kind of performance we expected and cannot be a replacement for lexical pattern-based approaches.

5.1 Future Work

We believe the problem of our approach is that it is not trained for the task that it will be used for. Training samples are generated generally from the terminology while the model is applied on non-lattice subgraphs. Therefore, moving forward we will focus more on the generation of much suitable training samples. Particularly, we will generate the samples so that they reflect the types of relations or non-relations that the model will see during the application. We expect to generate samples from lattice subgraphs: fragments of the terminology that agrees with the lattice property. To train for upper bound relation/non-relation, we will only use subgraph generated by descendants of a lattice-pair based on a certain context. Figure 7: (1) denotes such a subgraph. Here, from the lattice-pair A and B, we generate the subgraph from all their descendants. Similarly, to train for lower bound relations/non-relations, we will only use subgraph generated by ancestors of a lattice-pair. Figure 7: (2) denotes such a subgraph. Here, the lattice pair I, J derives the subgraph from their ancestors.

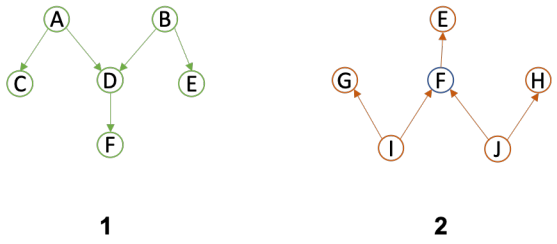

Figure 7: Generating samples from lattice subgraphs: (1) to train for upper bound concepts, (2) to train for lower bound concepts

6 Conclusion

We investigated a deep learning approach-based on graph neural networks to identify missing hierarchical relations in upper and lower bounds of non-lattice subgraphs. To train our model, we generated existing relations in SNOMED CT Clinical Finding subhierarchy as positive samples and existing non-relations as negative samples. After training the model, we applied it to predict missing is-a relations in non-lattice subgraphs. Our model predicted too many incorrect is-a relations and could not capture close to half of the missing is-a identified by lexical patterns, which suggests the precision and the recall of the method is not currently at an acceptable standard.

References

1. Lee D, de Keizer N, Lau F, Cornet R. Literature review of SNOMED CT use. Journal of the American Medical Informatics Association. 2013;21(e1):e11-9.
2. Cui L, Zhu W, Tao S, Case JT, Bodenreider O, Zhang GQ. Mining non-lattice subgraphs for detecting missing hierarchical relations and concepts in SNOMED CT. Journal of the American Medical Informatics Association. 2017;24(4):788-98.
3. Abeysinghe R, Brooks MA, Talbert J, Cui L. Quality assurance of NCI thesaurus by mining structural-lexical patterns. In AMIA Annual Symposium Proceedings. 2017:364-73.
4. Cui L, Bodenreider O, Shi J, Zhang GQ. Auditing SNOMED CT hierarchical relations based on lexical features of concepts in non-lattice subgraphs. Journal of biomedical informatics. 2018;78:177-84.
5. Agrawal A, Perl Y, Ochs C, Elhanan G. Algorithmic detection of inconsistent modeling among SNOMED CT concepts by combining lexical and structural indicators. In 2015 IEEE International Conference on Bioinformatics and Biomedicine (BIBM) 2015 Nov 9 (pp. 476-483). IEEE.
6. 5-Step Briefing [Accessed: 04 August 2019], Available from: `https://www.snomed.org/snomed-ct/ five-step-briefing`
7. Bodenreider O. Identifying Missing Hierarchical Relations in SNOMED CT from Logical Definitions Based on the Lexical Features of Concept Names. ICBO/BioCreative. 2016;2016.
8. SNOMEDCT US (SNOMED Clinical Terms US Edition) - Synopsis [Accessed: 04 August 2019], Available from: `https://www.nlm.nih.gov/research/umls/sourcereleasedocs/current/SNOMEDCT_US/`
9. Zhou J, Cui G, Zhang Z, Yang C, Liu Z, Sun M. Graph neural networks: A review of methods and applications. arXiv preprint arXiv:1812.08434. 2018 Dec 20.
10. Zhang GQ, Bodenreider O. Large-scale, exhaustive lattice-based structural auditing of SNOMED CT. InAMIA Annual Symposium Proceedings 2010 (Vol. 2010, p. 922). American Medical Informatics Association.
11. Wu Z, Pan S, Chen F, Long G, Zhang C, Yu PS. A comprehensive survey on graph neural networks. arXiv preprint arXiv:1901.00596. 2019 Jan 3.
12. Kipf TN, Welling M. Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907. 2016 Sep 9.
13. Le Q, Mikolov T. Distributed representations of sentences and documents. In International conference on machine learning 2014 Jan 27 (pp. 1188-1196).
14. Deep Graph Library [Accessed: 12 August 2019], Available from: `https://www.dgl.ai/`
15. Biowulf [Accessed: 12 August 2019], Available from: `http://hpc.nih.gov`