Validating Auto-Suggested Changes for SNOMED CT in Non-lattice Subgraphs Using Relational Machine Learning

Qi Sun[a,b], Guo-Qiang Zhang[b,d,e], Wei Zhu[b,c], Licong Cui[d]

[a] *Department of Computer Science, University of Kentucky, Lexington, KY, USA*
[b] *Institute for Biomedical Informatics, University of Kentucky, Lexington, KY, USA*
[c] *Department of Electrical Engineering and Computer Science, Case Western Reserve University, Cleveland, OH, USA*
[d] *School of Biomedical Informatics, University of Texas Health Science Center at Houston, Houston, TX, USA*
[e] *McGovern School of Medicine, University of Texas Health Science Center at Houston, Houston, TX, USA*

Abstract

An attractive feature of non-lattice-based ontology auditing methods is its ability to not only identify potential quality issues, but also automatically generate the corresponding fixes. However, exhaustive manual evaluation of the validity of suggested changes remains a challenge shared with virtually all auditing methods. To address this challenge, we explore machine learning techniques as an aid to systematically evaluate the strength of auto-suggested relational changes in the context of existing knowledge embedded in an ontology. We introduce a hybrid convolutional neural network and multilayer perception (CNN-MLP) classfier using a combination of graph, concept features and concept embeddings. We use lattice subgraphs to generate a curated, loosely-coupled training set of positive and negative instances to train the classifier. Our result shows that machine learning techniques have the potential to alleviate the manual effort required for validating and confirming changes generated by non-lattice-based auditing methods for SNOMED CT.

Keywords:

Non-lattice-based auditing, SNOMED CT, neural networks

Introduction

Non-lattice-based ontology auditing methods are based on the principle of Formal Concept Analysis. They have shown effectiveness in suggesting missing hierarchical relations (or IS-A relations) and concepts in SNOMED CT (SNCT) [1-3] as well as other teminology systems. A non-lattice-based approach consists of the following steps: extracting non-lattice subgraphs with concept pairs voilating the lattice property; detecting defects in the extracted non-lattice subgraphs; suggesting relational changes automatically; and reviewing and validating suggested changes by an ontology curator or a domain expert. Figure 1 shows a non-lattice subgraph in the September 2017 release of SNOMED CT (US edition) [3]. Examination of this subgraph reveals a missing hierarchical relation between nodes 4 and 5: "*Transient neonatal hyperglycemia*" IS-A "*Acute hyperglycemia.*"

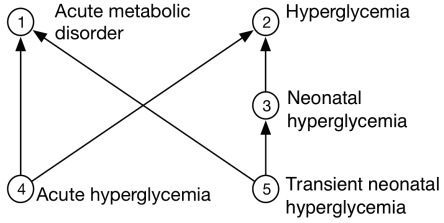

Figure 1 - An example of non-lattice subgraph [3].

However, human review of change remediations requires significant manual effort. The main motivation of this paper is to explore machine learning (ML) techniques as an aid to systematically evaluate the strength of auto-suggested relational changes using existing knowledge embedded in an ontology. Specifically, change predictions resulting from the IS-A relation prediction classifier can be compared with change suggestions using non-lattice subgraphs by assessing their agreement. When such an approach is used to audit the IS-A relations, it can bring benefits in two ways: the agreement between the ML prediction and non-lattice-based prediction can provide independent confirmation of the suggested IS-A changes, while disagreements may point to areas for improvements for both types of approaches. Furthermore, change suggestions generated by ML techniques may provide ranked list of changes based on numeric "strengths." If an ML technique can reach a high level of performance (using e.g. precision and recall measures), it may serve as an independent auditing approach by itself independent of non-lattices.

ML techniques have been widely used in knowledge graphs [4]. To arrive at an optimal subset of features, CNN has been used to automatically learn features [5]. Also pretrained word embeddings [6] have proven to be useful in neural network models for relation extraction. Recent study further improved performance by incorperating additional hand-crafted features [7]. Inspired by this, we propose a hybrid CNN-MLP classifier for IS-A relation prediction by exploring various knowledge in SNCT. Besides of concept embeddings, we analyze lexical and graph structural information in the entire directed acyclic graph (DAG) of SNCT, such as semantic meaning inheriting concept ancestors. Pesquita et al. [8] summerized two main semantic similarity approaches in an ontology graph for comparing concepts: node-based and edge-based. In this work, we aggregate node-based and edge-based similarity approaches as graph level features. Despite concept embeddings can present semantic meaning of concepts, we still handcraft concept level features as task specific features. Non-lattice-based auditing methods suggest IS-A relational changes to correct defects in non-lattice subgraphs and form lattice subgraphs [1-3]. Accordingly, constructing training set from lattice subgraphs equips us with the same objective of non-lattice-based methods.

In this paper, we introduce an ML approach to validate auto-suggested insertion changes from non-lattice-based auditing methods. Combining various knowledge in lattice subgraphs and DAG of SNCT, we implement IS-A relation prediction and auto-insertion change suggestion as two subtasks in the workflow of validation. We apply CNN to explore the concept pair features in concept embeddings and combine the output of CNN with handcrafted graph, concept features via MLP to

predict IS-A relations of concept pairs. Also evaluation metrics for subtasks are provided. Evaluation of our validation method is implemented on a reference set.

Methods

Our method consists of two main steps. In the first step, learning from various knowledge embedded in lattice subgraphs of SNCT, a hybrid CNN-MLP classifier is designed to predict IS-A relations of concept pairs. In the second step, based on the idea of majority voting [10], we validate the given insertion changes with predicted insertion changes generated by the classifier.

Relation Prediction

Figure 2 is an overview of our relation prediction workflow. First, we generate a training set using lattice subgraphs. Second, we pretrain concept embedding and develop graph and concept level features. Finally, we predict IS-A relations of concept pairs with a hybrid CNN-MLP classifier.

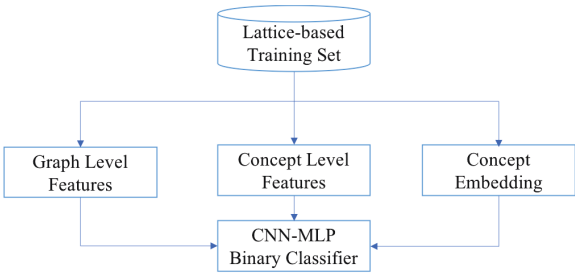

Figure 2-An overview of the IS-A relation prediction workflow

Lattice-based Training Set

A desirable property of the IS-A relationship is that it conforms to the lattice property [9]. This is also a by-product of non-lattice-based auditing methods: non-lattice subgraphs often become lattice-conforming by correcting hierarchical defects (e.g., inserting IS-A relations). Using this as working principle, we construct the training set for our CNN-MLP classifier using relations embedded in lattice subgraphs. Lattice subgraphs also allow us to naturally select a balanced set of positive samples and negative samples. Extracting all lattice subgraphs in SNCT, we construct a lattice-based training set with the following steps.

1. Select a subset of concepts with label length, which is the number of words in a label, no longer than α. We denote this set of concepts as $Con(\alpha)$.
2. Among all lattice subgraphs, choose those with size, which is the number of concepts in a subgraph, no more than β, denoted as $Lattice(\beta)$, and find those that contain any concept in $Con(\alpha)$.
3. Form a positive training set by extracting IS-A edges and their transitive closure from step 2. Form a negative training set by extracting non-edge node pairs in step 2.

Concept Embedding

Learning from current ontologies, the embedding techniques [6] allow us to represent concepts and capture latent semantic properties of concepts. In SNCT, each concept contains an associated human readable label, and concepts occuring in the same relation could describe each other. We take relations represented with labels as sentences to learn word embeddings. After perfoming a few basic text preprocessing steps, such as removing punctuations and digits, converting to lowercase and stemming, we use a skip gram word2vec model [6] to produce word embeddings with 200 dimensions. To generate concept embeddings, a sequence of words in concept labels are transformed to a set of vectors by looking up pretrained word embeddings. The out-of-vocabulory (OOV) word is assigned a vector with zeros. Then the word embeddings are summed into a single embedding as a concept embedding.

Feature Development

The IS-A relation comes with a source concept C_1 and a destination concept C_2. We develop features based on two types of observations. One is that from global graph level of entire IS-A DAG in SNCT, a concept at a lower level inherits semantics from its ancestors and is more specific in terms of the biomedical meaning [11]. The other is that measured from local level of concept pairs, if there are overlaps in noun phrases of C_1 and C_2, the noun phrase in C_1 usually contains more words than C_2 as C_1 becomes more specific. The more overlaping words the two concepts have, the more likely they should form an IS-A relation.

Graph level features. The work of Zhang et al. [11] and Cui et al. [3] provides several heuristics to generate graph level features. Considering concept semantics inherited from its ancestors and path information, we design three graph level features for situations of a concept pair to be in an IS-A relation. All graph level features are explored from the entire active IS-A graph of SNCT.

1. Graph level dissimilarity. The semantic of a concept could be specified through a path to its descendants, that is, a concept inherits all semantics from its ancestors. A pair of concepts, inheriting semantics from different paths, may have some common ancestors. Non common ancestors of C_1 distinguished C_1 from C_2, then the graph-level dissimilarity can be characterized by

   $$GraphDissim(C_1,C_2)=1-CA(C_1,C_2)/A(C_1)$$

   where $CA(C_1,C_2)$ is the common ancestor set of C_1 and C_2, and $A(C_1)$ is the ancestor set of C_1.
2. Graph level similarity. The semantic of a concept could be simplified with a concept label. While a given concept inherits semantics from its ancestors, the concept labels of ancestors could be used to enrich the label of the concept. For an IS-A relation (say C_1 IS-A C_2), because C_1 is more specific than C_2, the enriched label of C_2 maybe a part of enriched label of C_1, so we define graph level similarity as

   $$GraphSim(C_1,C_2)=CEL(C_1,C_2)/EL(C_2)$$

   where $CEL(C_1,C_2)$ is the common enriched label set of $C_1$and C_2, and $EL(C_2)$ is the enriched label set of C_2.
3. Path comparison. If C_1 IS-A C_2, we assume that C_2 has a shorter path to the root R than C_1 does. We define a flag feature indicating whether the shortest path from C_2 to R is shorter than that of C_1 to R.

Concept level features. This category of features concerns about properties of labels in concept pairs.

1. Chunk comparison. We observe that if there are overlaps between noun phrases in C_1 and C_2, the noun phrase in C_1 usually has more words than that in C_2. We design two flag features from all the extracted noun phrases in both C_1 and C_2 to indicate word number comparison of noun phrases in C_1 and C_2 separately.
2. Concept level overlap. This feature considers overlaps of pairwise concepts. We define it as:

   $$ConceptOLP(C_1,C_2)=W(C_1)\cap W(C_2)/W(C_2)$$

   where $W(C_1)\cap W(C_2)$ is the overlapped word set of C_1 and C_2, and $W(C_2)$ is the word set of C_2. High value lesser than one implies that the meaning of C_1 approaches the meaning of C_2.
3. Concept level similarity. This feature concerns about the consine similarity of disjoint words in $C_1$and C_2, and

disjoint words are represented with the summation of word embeddings.

Neural Network Classification

We design a hybrid CNN-MLP binary classifier to incorporate concept embedding and handcrafted features (see Figure 3).

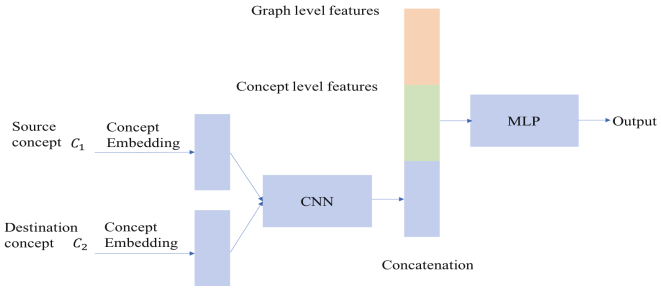

Figure 3 - Architecture of hybrid CNN-MLP binary classifier

We convolute the embeddings of concept pairs with 100 filters of window size 2 to get local features and use ReLU activation function. In order to fomulate a balanced feature vector between embeding features and handcrafted features, we transfer 100 dimension vector into 10 dimension vector by fully connected layer in CNN, which then is concatenated with graph level and concept level features. The concatenated feature vector is fed into a one layer MLP with active function of ReLU for the hidden layer and sigmoid for the output layer. In order to evaluate the performance of features, we feed MLP with different feature combinations, thus the number of neurons of MLP hidden layer is decided by the average of neurons in MLP input and output layers. Time complexity of neural network depends on the architechture, especially the number of neurons, of the network.

Validation of Auto-Suggested Insertion Changes

The objective of our work is using an ML approach to validate auto-suggested insertion changes. The validation workflow involves four stages: test set generation, relation prediction, auto-insertion change suggestion, and majority voting based validation. In stage 1, non-lattice-based auditing method [3] provides a set of non-lattice subgraphs containing auto-suggested insertion changes. IS-A relations and their transitive closure (positive instances), and non-edge concept pairs (negative instances) are extracted from the given non-lattice subgraphs to form a test set. In stage 2, the IS-A relations of extracted positive instances and negative instances are predicted by our hybrid CNN-MLP binary classifier. In stage 3, we retrieve top N predictions ranked by IS-A relation probability outputs. The predicted false positives are the set of auto-suggested insertion changes provided by our classifier. In stage 4, inspired by majority voting [10], we confirm the auto-suggested changes with the agreement between our classifier and the given auditing method. In this way, relational ML techniques alleviate the manual validation effort by confirming and validating changes automatically.

In the validation workflow, our classifier is used in two subtasks: predicting IS-A relation of concept pairs in stage 2, and automatically suggesting insertion changes in non-lattice subgraphs in stage 3. To evaluate performance of subtasks and determine configuration of the classifier for the validation task, we specify a test set, a reference set and evaluation metrics next.

Test Set and Reference Set

To evaluate our method, we use relations in a random sample of 200 non-lattice subgraphs from [3] as a test set, which contains 1,545 IS-A and transitive closure of IS-A edges (positive instances), 3,019 non-edge node pairs (negative instances). A total of 223 insertion changes were auto-suggested for the 200 non-lattice subgraphs [3]. Two domain experts confirmed 185 suggested insertions (a precision of 82.96%), which serve as a reference set for evaluating the performance of this work.

Evaluation of Relation Prediction

To thoroughly analyze the relation prediction part along, e.g., comparing training set parameters (α,β) and different categories of features, we predict 1,545 positive instances and 3,019 negative instances in the test set. For evalution, we report the precision-recall curve.

Evaluation of Auto-Suggested Insertion Changes

Motivated by precision and recall metrics, Zhang et al. introduced insertion recall and insertion precision [12] to evaluate an ontology quality assurance (OQA) method against validated changes. They considered an OQA method M as a group of subgraphs. Each subgraph may potentially capture missing IS-A relations as edges. A subgraph s is a graph consisting of a set of IS-A relations. E is a reference set of validated missing IS-A relations. Then:

$$R_{insert} = \frac{|\{r \in E / \exists s \in M, r \in s\}|}{|E|}$$

$$P_{insert} = \frac{|\{s \in M / \exists r \in s, r \in E\}|}{|M|}$$

$$F_{insert} = \sqrt{P_{insert} \times R_{insert}}$$

Our classifier outputs probabilities which can be used to rank instances in the test set from most probable IS-A relation to the least. In order to demonstrate the ability to validate auto-suggested changes, we treat our method as an OQA method and take top N predictions as predicted IS-A relations. False positives predicted by our classifier are auto-suggested insertion changes. The reference set contains 185 validated missing IS-A edges and their transitive closure. We use R_{insert}, P_{insert}, F_{insert} [12] to evaluate auto-suggested insertion changes obtained via our method. The higher the F_{insert}, the greater the agreement between our method and domain expert's validation, that is, more likely our method has the ability to alleviate the manual effort required for validating auto-suggested changes.

Results

Dataset and Implementaion

We used the September 2017 version of SNOMED CT (US edition) in this work. We constructed a lattice-based training set [1] by varying label length α and lattice size β. The statistics of training set are shown in Table 1. While relation prediction performance was evaluated with the test set, the reference set was used to evaluate auto-insertion change suggestion. We implemented the model using Keras [13]. Declaring binary cross entropy as the loss function, we ran 10 epochs for all the training examples. Noun phrase chunk detection and preprocessing for learning embeddings were implemented with NLTK [14].

Table 1 - Statistics of Training Set

	# of positives	# of negatives
$\alpha = 5, \beta = 5$	151,553	243,197
$\alpha = 5, \beta = 10$	515,954	1,005,074
$\alpha = 10, \beta = 5$	197,630	319,286
$\alpha = 10, \beta = 10$	693,776	1,351,872

Relation Prediction

Experiments were performed to evaluate IS-A relation prediction by permuting training set construction parameters and various feature categories. The experiments were performed on a MacBook Pro running MacOS Sierra, with 16 GB RAM and 2.7 GHz Intel Core i7 processor.

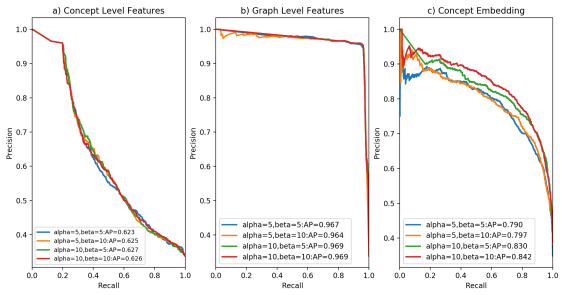

Figure 4 - The performance of relation prediction varying feature categories and training set construction parameters

Comparing Figure 4a), 4b) and 4c), the best prediction performance was produced with graph level features. When recall of IS-A relations was increased, the precision was decreased with concept level features. Though concept embeddings and concept level features both describe sementics of concept pairs, concept embeddings worked better than handcrafted concept level features.

Figure 4a) shows that longer concept label expresses concept level features more effectively, slightly leading to prediction performance improvement. Figure 4b) shows that graph level features are independent with label length α and lattice size $\boldsymbol{\beta}$ which is coincident with development of graph features, especially while increasing recall. As for the feature category of concept embeddings, prediction performance varied with different settings of training set parameters, which may be explained by Table 2 by OOV word rate over test set while looking up pretrained word embeddings. More OOV words in test set resulted in decrease of relation prediction performance.

Table 2 - Out-of-vocabulary word rate over test set

	$\alpha = 5$ $\beta = 5$	$\alpha = 5$ $\beta = 10$	$\alpha = 10$ $\beta = 5$	$\alpha = 10$ $\beta = 10$
OOV rate	4.4%	3.7%	3.0%	2.6%

Figure 5 shows that combining graph level features, concept level features and concept embeddings as the input of MLP had led to performance improvement. The best average precision was 0.972, achieved with $\alpha=5$, $\beta=5$. Overall, our experiments showed that exploring various features was effective for IS-A relation prediction.

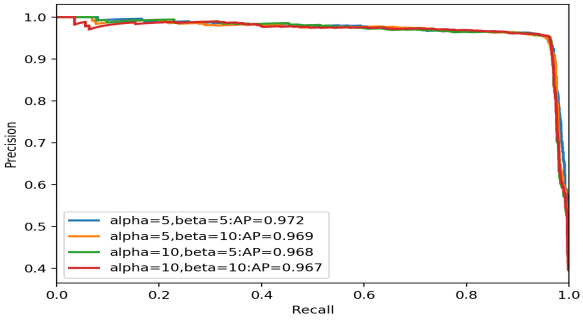

Figure 5-The performance of relation prediction with feature combination varying training set construction parameters

Auto-Suggested Insertion Changes

In this set of experiments, we evaluated our auto-suggested insertion changes for non-lattice subgraphs. A set of 185 validated missing IS-A relations was taken as the reference set. Among top N IS-A relation predictions, we evaluated auto-suggested insertion changes by R_{insert} , P_{insert} , F_{insert} values as defined in the subsection of auto-suggested insertion changes evaluation. Informed by the previous experiments, we configurated this set of experiments with feature combination and setting $\alpha=5$, $\beta=5$.

The test set contained 1,545 positive instances and 3,019 negative instances. Ranked by probability outputs of our classifier, we kept top N predictions as IS-A relations (predicted positive instances). Predicted false positives constituted insertion change suggestions in the non-lattice subgraphs. The top 1,500 predictions contained 129 IS-A relations already identified in the reference set of 185 validated missing IS-A relations. As we retrieve more predicted IS-A relations from the ranked list (i.e., as N increases), we obtain more validated missing IS-A relations. Note that both insertion recall (R_{insert}) and insertion precision (P_{insert}) increase as N increases. We stopped retrieving predictions at N=2,000 in case more irrelevant IS-A relations out of the referece set would be included.

Table 3 - Evaluation of auto-suggested insertion changes at top N predictions

Top N	R_{insert}(%)	P_{insert}(%)	F_{insert}(%)
1500	69.57	59.80	64.50
1600	80.98	69.35	74.94
1700	85.32	72.36	78.57
1800	86.41	72.86	79.35
1900	89.67	75.38	82.22
2000	90.21	75.88	82.74

Compared with the reference set, Table 4 shows examples of auto-suggested insertion changes from top 2,000 predictions in and out of the reference set. For those out of the reference set, they may reveal additional insertation changes needed to make non-lattice subgraphs conforming to the lattice property.

Table 4-Examples of auto-suggested insertion changes compared with the reference set

In the reference set

Source Concept	Destination concept
Thoracic spondylosis	Spondylosis
Sclerema neonatorum	Neonatal dermatosis
Acute empyema of sphenoidal sinus	Sphenoidal sinusitis
Oculocutaneous albinism	Congenital anomaly of eye
Degloving injury of genitalia	Degloving injury of perineum

Out of the reference set

Source concept	Destination concept
Echography scan B-mode for fetal growth rate	Ultrasound scan of fetus
Dilatation of anastomosis of bile duct	Biliary dilatation procedure
Feeding problems in newborn	Feeding problem
Physiological mobilization of the shoulder	Procedure on shoulder region
Carcinoma in situ of lower labial mucosa	Tumor of lower labial mucosa

The evaluation results and examples of auto-suggested insertion changes indicated that our method is effective to predict IS-A relations of concept pairs. Verified by the reference set, our method automatically suggested insertion changes for non-lattice subgraphs.

Validation of Auto-Suggested Insertion Changes

The performances of two subtasks in the workflow of validation, relation prediction and auto-insertion change suggestion, have been demonstrated to be promising by two sets of experiments. Configured with feature combination and $\alpha=5$, $\beta=5$ for the classifier, we evaluated 223 auto-suggested insertion changes in the test set. We compared insertion

changes resulting from our classifier with the given 223 insertion changes, and those changes with agreement were validated. The number of insertion changes validated by our classifier is shown in Table 5. Since 185 insertion changes were validated by domain experts, we also showed the number of expert-validated insertion changes among our classifier's validation. Validated by our method, the auto-suggested insertion change precision improved from 82.96% to 86.46%. We can further alleviate manual effort by narrowing down the number of insertion changes with the improvement of precision.

Table 5 – Validation evaluation at top N predictions compared with the reference set

Top *N*	# of Classifier Validated	# of Expert Validated	Precision (%)
1500	150	129	86.00
1600	174	149	85.63
1700	183	157	85.79
1800	185	159	85.95
1900	191	165	86.39
2000	192	166	86.46

Discussion

In this paper, we introduced a relational ML-based validation approach, CNN-MLP, to alleviate manual effort required for validating change suggestions in ontology auditing work. In addition to validating auto-suggested insertion changes, our experiments indicated that CNN-MLP classifier may be effectively used in tasks of concept pair relation prediction and ontology auditing. As shown in Table 4, five auto-suggested insertion changes were out of the reference set; although they were not used to validate change suggestions in the reference set, they may provide additonal candidates for change redemiations.

Evaluated with the reference set, the performances of auto-suggested insertion change and auto-suggested insertion change validation look promising. However, note we have not addressed the task of auto-deletion change suggestion and deletion change validation due to the lack of the reference set containing validated deletion changes. Labeled data (or reference set) is required in the framework of supervised ML approach. SNCT is updated by domain experts twice a year, and it may be possible to treat insertion and deletion changes in different versions of SNCT as labeled data for ML based auditing or validation.

Conclusions

We have presented an ML approach to validate insertion changes generated by non-lattice-based auditing methods. In validation workflow, we introduced a hybrid CNN-MLP classifier to predict IS-A relations of concept pairs, and automatically suggest insertion changes for non-lattice subgraphs. Experiments on IS-A relation prediction achieves an average precision of 0.972, and auto-suggested insertion changes achieves F_{insert} of 82.43% with top 2,000 predictions, which indicate the potential of our classifier to validate IS-A insertion changes. Validated with insertion changes resulting from our classifier, the precision of a given set of auto-suggested IS-A insertion changes is improved from 82.96% to 86.46%.

By cumulating SNCT deletion changes as a surrogate reference set for ML approaches [12], our future work will focus on exploring features to predict potentially incorrect IS-A relaitons for validating auto-suggested deletion changes in non-lattice subgraphs.

Acknowledgements

This work was supported by the U.S. National Science Foundation under grant 1816805, and by the U.S. National Institutes of Health under grant R21CA231904.

References

[1] G.-Q. Zhang, W. Zhu, M. Sun, S. Tao, O. Bodenreider, and L. Cui, MaPLE: A MapReduce Pipeline for Lattice-based Evaluation and Its Application to SNOMED CT, *Proc IEEE Int Conf Big Data*. **2014** (2014) 754–759.

[2] L. Cui, W. Zhu, S. Tao, J.T. Case, O. Bodenreider, and G.-Q. Zhang, Mining non-lattice subgraphs for detecting missing hierarchical relations and concepts in SNOMED CT, *J. Am. Med. Inform. Assoc.* **24** (2017) 788–798.

[3] L. Cui, O. Bodenreider, J. Shi, and G.-Q. Zhang, Auditing SNOMED CT hierarchical relations based on lexical features of concepts in non-lattice subgraphs, *J. Biomed. Inform.* **78** (2018) 177–184.

[4] M. Nickel, K. Murphy, V. Tresp, and E. Gabrilovich, A Review of Relational Machine Learning for Knowledge Graphs, *Proc. IEEE*. **104** (2016) 11–33.

[5] D. Zeng, K. Liu, S. Lai, G. Zhou, and J. Zhao, Relation Classification via Convolutional Deep Neural Network, *Proc. of COLING* (2014), Technical Papers.

[6] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, and J. Dean, Distributed Representations of Words and Phrases and their Compositionality, *Advances in Neural Information Processing Systems* **26** (2013) 3111–3119.

[7] E. Park, X. Han, T.L. Berg, and A.C. Berg, Combining multiple sources of knowledge in deep CNNs for action recognition, *IEEE Winter Conference on Applications of Computer Vision* (2016) 1–8.

[8] C. Pesquita, D. Faria, A.O. Falcão, P. Lord, and F.M. Couto, Semantic similarity in biomedical ontologies, *PLoS Comput. Biol.* **5** (2009).

[9] P. Zweigenbaum, B. Bachimont, J. Bouaud, J. Charlet, and J.F. Boisvieux, Issues in the structuring and acquisition of an ontology for medical language understanding, *Methods Inf. Med.* **34** (1995) 15–24.

[10] M. van Erp, L. Vuurpijl, and L. Schomaker, An overview and comparison of voting methods for pattern recognition, *Proceedings Eighth International Workshop on Frontiers in Handwriting Recognition* (2002) 195–200.

[11] S.-B. Zhang, and J.-H. Lai, An integrated information-based similarity measurement of gene ontology terms, *Computer Science and Information Systems*. **12** (2015) 1235–1253.

[12] G.-Q. Zhang, Y. Huang, and L. Cui, Can SNOMED CT Changes Be Used as a Surrogate Standard for Evaluating the Performance of Its Auditing Methods? *AMIA Annu. Symp. Proc.* **2017** (2017) 1903–1912.

[13] Keras documentation, (2016). http://keras.io [accessed on March 27, 2019]

[14] Toolkit-NLTK 3.4 documentation, https://www.nltk.org [accessed on March 27, 2019]

Address for correspondence

Licong Cui (Licong.Cui@uth.tmc.edu)