Exploring Deep Learning-based Approaches for Predicting Concept Names in SNOMED CT

Fengbo Zheng*, Licong Cui*†
*Department of Computer Science
†Institute for Biomedical Informatics
University of Kentucky, Lexington, Kentucky, USA

***Abstract*—Ontologies or terminologies have been widely used as formal representation of biomedical knowledge. New concepts are constantly added to biomedical ontologies due to the evolving nature of biomedical knowledge. Much progress has been made to identify new concepts in SNOMED CT, the largest clinical healthcare terminology. However, proper naming of new concepts remains challenging and relies on the ontology curators' manual effort. In this paper, we explore three deep learning-based approaches, given bags of words, to automatically predict concept names that comply with the naming convention of SNOMED CT. These deep learning models are simple neural network, Long Short-Term Memory (LSTM), and Convolutional Neural Network (CNN) combined with LSTM. Our experiments showed that LSTM-based approach achieved the best performance: a precision of 65.98%, a recall of 61.04%, and an F1 score of 63.41% for predicting concept names for newly added concepts in the March 2018 Edition of SNOMED CT. It also achieved a precision of 74.58%, a recall of 73.33%, and an F1 score of 73.95% for naming missing concepts identified by our previous work. Further examination of results revealed inconsistencies within SNOMED CT which may be leveraged for quality assurance purpose.**

***Index Terms*—Biomedical Ontologies, Naming Prediction, Neural Networks, Ontology Completeness**

I. Introduction

Ontologies provide formalized representation of concepts and relations among concepts for a specific domain. Biomedical ontologies or terminologies provide domain knowledge to support applications such as semantic annotation of biomedical data, knowledge discovery and exchange, data integration, natural language processing and decision support [1], [2].

To keep up with the rapid development of knowledge in biomedicine, biomedical ontologies are often constantly evolving [3] and new concepts are regularly added to newer versions of ontologies. For example, SNOMED CT, the most comprehensive clinical healthcare terminology product, is released regularly in every six months [4].

Therefore, identifying missing or new concepts in biomedical ontologies (i.e., ontology enrichment) has been an active research area [5]–[10]. He et al. developed a topological-pattern-based method by leveraging mappings between the Unified Medical Language System (UMLS) and its source terminologies to identify potentially missing concepts in SNOMED CT [5] and the National Cancer Institute Thesaurus (NCIt) [6]. Jiang and Chute proposed a Formal Concept Analysis (FCA)-based method to construct the formal context of SNOMED CT and then searched for possible missing concepts [7]. Zhu et al. improved Jiang and Chute's work and developed a scalable multistage algorithm called Spark-MCA [8] which enabled an exhaustive FCA evaluation on all the SNOMED CT concepts within a reasonable amount of time. Cui et al. introduced a structural-lexical method by mining lexical patterns in non-lattice subgraphs, where one of the patterns can identify missing concepts in SNOMED CT and NCIt [9], [10].

Although much progress has been made to identify missing or new concepts, proper naming of those new concepts remains challenging and relies on the curators of biomedical ontologies. However, it is hard for different curators to maintain the same standard and keep it consistent while naming thousands of concepts. Also, according to the experiment results from Zhu et al's work [8], even in well-constructed and mature ontologies such as SNOMED CT, there still exists a large amount of missing concepts. It is labor-intensive and time-consuming for curators to manually find appropriate and unambiguous names for a large number of concepts. Therefore, automated methods are highly desirable to provide suggestions on concept names to reduce curators' manual burden and accelerate the ontology maintenance process.

Deep learning approaches have been widely used in text related tasks such as language modeling, textual analysis, information retrieval and sequence generation, and showed better performance than traditional machine learning methods in these tasks [11]. In this paper, we explore deep learning-based approaches to provide suggestions on potential names for new concepts in SNOMED CT.

Many existing works on ontology enrichment are capable of generating necessary words to construct a concept name. However, the words may be unordered, or the order of the words may not be consistent with the naming convention of the given ontology. For instance, one of the structural-lexical patterns in [9] can suggest a bag of words for each potentially missing concept, such as {*of, neoplasm, malignant, upper, lobe, right, of, lung*} whose proper name should be "*malignant neoplasm of right upper lobe of lung*". Thus, our focus in this paper is generating proper sequence order for a given bag of words. We conquer this task by dividing it into two subtasks.

This work was supported by the National Science Foundation under grants 1816805 and 1657306, and by the National Institutes of Health under grant R21CA231904. Correspondence: licong.cui@uky.edu

The first subtask is a binary classification problem to determine if a given sequence of words is a correct concept name or not, for which we explore three types of neural networks: a simple neural network, a Long Short-Term Memory (LSTM) neural network, and a combination of convolutional neural network (CNN) and LSTM. The second subtask is, given a bag of words, to predict correct sequence of words using the trained models in the first subtask. To the best of our knowledge, this is the first work that automatically predict names for new or missing concepts in biomedical ontologies.

The remainder of this paper is organized as follows: Section II specifies the problem and the deep learning models we use. Section III introduces the experiments we conducted and discusses the experiment result. Section IV discusses the limitations of our approach and future work. Section V concludes this paper.

II. Method

In this paper, we focus on addressing the problem of predicting the word sequence given a bag of words for naming a concept. To achieve this, we divide this task into the following two subtasks. Firstly, given a word sequence, we train the models to determine whether the given sequence or ordering is correct (meaningful and satisfying the naming convention of SNOMED CT) or not (binary classification). Secondly, given a bag of words, we utilize the trained models to predict its correct word sequence. Since the trained models return different confidence levels of the correctness judgment for different sequences, we test all the possible sequences of the given words and choose the one(s) with the highest confidence as the predicted concept name(s). Further, we implement a two-step filter to eliminate those potential incorrect candidate concept name(s).

A. Word Embedding & Data Preprocessing

We first use Word2Vec [12] provided by Gensim [13] to learn vector representation of words in SNOMED CT concept names. Each word is mapped to a 125-length vector (word embedding) based on the word(s) surrounding it, and thus mapped to a higher dimensional space. The mappings between words and their word embeddings are stored in a matrix which later will be reused in training the neural network models.

To train the models, we need both positive and negative training data. The original sequences of words (concept names) are labeled as "1" or correct. Then, for each correct sequence, we generate n incorrect sequences ($n = 5$ in this paper) by randomly disordering the sequence and the generated sequences are labeled as "0" or wrong. Since concept names in SNOMED CT are in different lengths (ranging from 1 to 39), all sequences are padded to the same length of 45.

B. Neural Networks for Classifying Word Sequences

We use Keras [14], a high-level neural networks API, to implement three types of deep learning models as shown in Fig. 1. For all three models, there is an embedding layer right after the input layer. It is responsible for mapping each word in the input sequence to its word embedding. The pre-computed word embedding matrix will serve as the trained weight for this layer.

The first model (Fig. 1(a)) is a simple neural network for binary classification. Since word embedding increases the input dimension, we need to flatten the input. Then we use a dense layer which has a *sigmoid* activation function to generate the prediction result. For the classification task, we use *binary_crossentropy* in Keras, which is often used for binary classification problems, in all three models while training.

The concept names in SNOMED CT are in various lengths, and the position of a word may not only depend on the words next to it. This requires the model to be able to learn long-term context and dependencies between words in the input sequence. LSTM is proven to be good at learning such dependencies [15]. Thus we also adopt LSTM (Fig. 1(b)) for binary classification that contains a single LSTM layer with 100 LSTM memory unit in the middle.

When it comes to determining the correctness of a sequence, there may exist featured patterns which can indicate whether a sequence order is correct or not. Thus, to capture those potential featured patterns, we employed a model in which convolutional neural network (CNN) is adopted and combined with LSTM (Fig. 1(c)). CNN is commonly applied to analyzing visual imagery and it can identify lower level features from the minimum unit of the input which eventually may improve the classification process [16]. CNN also benefits sentence classification, such as sentiment analysis and question classification [17]. Regarding our work, the input is sequence of words (concept name) which can be considered as one dimension spatially. By using one-dimension CNN (Conv1D), certain word combinations or patterns will be selected as lower level feature and these learned spatial features will then be learned by an LSTM layer. In this model, the number of output filters in the Conv1D layer is 32 and the window size is set to three. The *pool_size* [18] for the *MaxPooling1D* layer is two.

As mentioned in the data preprocessing step, the number of input data labeled as "1" is less than those labeled as "0", with a proportion of $1 : 5$. To lessen the impact of unbalanced data, we assign different weights to different classes so that during training, the model will weight class "1" more when adjusting the weight.

C. Predicting Concept Names Given Bags of Words

To suggest candidate concept name for a given bag of words, we first generate all its permutations (i.e., all possible sequences). Then we use the trained models to classify those generated sequences to check which one is valid. While performing classification, the neural network models could return a confidence score (probability) for a sequence to be valid. Thus, we select the sequence(s) with the highest confidence score to be valid as the potential concept name(s) for a given bag of words. Because the number of permutations of n distinct objects is n factorial. When n becomes relatively large, the computation time increases dramatically. Thus, in

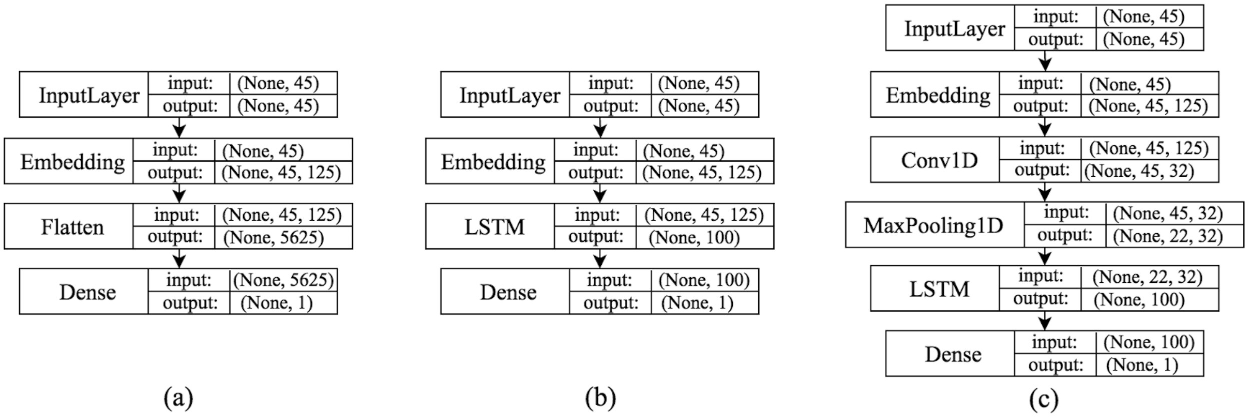
InputLayer | input: (None, 45) | output: (None, 45)
Embedding | input: (None, 45) | output: (None, 45, 125)
Flatten | input: (None, 45, 125) | output: (None, 5625)
Dense | input: (None, 5625) | output: (None, 1)
(a)
InputLayer | input: (None, 45) | output: (None, 45)
Embedding | input: (None, 45) | output: (None, 45, 125)
LSTM | input: (None, 45, 125) | output: (None, 100)
Dense | input: (None, 100) | output: (None, 1)
(b)
InputLayer | input: (None, 45) | output: (None, 45)
Embedding | input: (None, 45) | output: (None, 45, 125)
Conv1D | input: (None, 45, 125) | output: (None, 45, 32)
MaxPooling1D | input: (None, 45, 32) | output: (None, 22, 32)
LSTM | input: (None, 22, 32) | output: (None, 100)
Dense | input: (None, 100) | output: (None, 1)
(c)

Fig. 1. Three neural network models used for classification task. (a) is a simple neural network for binary classification; (b) is an LSTM neural network; (c) is the combination of convolutional neural network and LSTM. "None" means that the dimension is variable.

this paper, we only provide prediction for concept names whose length is less than or equal to 9.

Since there may exist multiple sequences with the highest confidence score for a given bag of words, we implement a two-step filtering process to further select the "best" candidate(s). In the first step, we leverage the idea of Viterbi algorithm [19] which returns the most likely sequence of hidden states, to removed those "invalid" ones. It is based on the assumption that if a word A has never been placed before (or after) another word B in the training data set, then A is not likely to appear before (or after) B in the candidate concept names. In the second step, we leverage similar concept names to further reduce the candidate list. For a bag of words, we first find the most similar concept name (in terms of the bag of words) in the training data, where similarity is calculated by dividing the number of words that appear in both bags of words by the total number of distinct words in two bags. Then we utilize "Levenshtein distance" [20] to compute the distances between each candidate name and the most similar concept name, and the one with the least distance will be selected as the "best" candidate concept names.

III. Experiment & Result

To validate the effectiveness of our models, we focused on two research questions:

1) Are the deep learning models able to determine if a sequence of words is a valid concept name in SNOMED CT? (binary classification)
2) Given a bag of words, can our method generate the correct sequence using those words and thus provide suggestions on how to name a concept in SNOMED CT? (sequence prediction)

A. Experiment Setup

To explore the first question regarding binary classification, we performed two experiments. In the first experiment, we randomly separated the concept names in the March 2018 US Edition of SNOMED CT into two groups: training and test datasets. There are 1,753,513 labeled sequences in the training data set and 1,784,744 labeled sequences in test data set. In the second experiment, we trained our models by all the concept names in September 2017 US Edition of SNOMED CT (along with randomly disordered ones). For testing, we extracted all the new concept names that were added into the March 2018 US Edition of SNOMED CT and generated disordered ones. For each of these two experiments, after training our models with the training dataset, we evaluated our models using the test dataset.

For the second question regarding sequence prediction, we evaluated our method in two ways. In the first way, we considered the sequence orders of the newly added concept names in the March 2018 US Edition of SNOMED CT as the ground truth for sequence predication. For each concept name, we regarded it as a bag of words and used our method to generate candidate concept names. Then we compared these suggestions with the ground truth. In the second way, we identified a collection of missing concepts (in the September 2017 US Edition of SNOMED CT) and the corresponding bags of words that are necessary to construct their names using Cui et al.'s method [9]. A total of 60 concepts in the form of bags of words were obtained for testing the performance of word sequence prediction with the help of a human annotator for validation.

B. Result for Binary Classification

In the first experiment for binary classification, we first tested different thresholds of confidence (for a sequence to be labeled as positive) for three models to achieve the best F1 score. The results are 0.5 for simple neural network, 0.8 for LSTM, and 0.7 for CNN and LSTM, respectively. The results of binary classification for three models using these thresholds are shown in Table I, where it can seen that the LSTM model outperformed the other two models, and achieved the best performance: an accuracy of 94.72%, a precision of 84.59%, a recall of 83.51%, an F1 score of 84.05%, and an FP-rate of 3.04%. The simple neural network performed the worst, and generated more false positives than the other two models. This is not surprising because while determining if a sequence is correct or not, the order of words matters and the simple neural network may not be able to learn long-term dependencies. In addition, it is shown that the combination of CNN and LSTM did not improve the performance of binary classification.

TABLE I
RESULT OF BINARY CLASSIFICATION FOR EXPERIMENT I.

	Simple Neural Network	CNN and LSTM	LSTM
Accuracy	73.58%	92.15%	94.72%
Precision	35.42%	74.23%	84.59%
Recall	71.08%	81.03%	83.51%
F1 Score	47.28%	77.48%	84.05%
FP-rate	25.91%	5.63%	3.04%

The result of the second experiment for binary classification is shown in Table II. It can be seen that the LSTM model still achieved the best performance. The main difference between the two experiments is that the first one's evaluation is within the same version of SNOMED CT, however, the second one is using concept names from a newer version of SNOMED CT to test the model which was trained by the older version. Although the training data in the second experiment is much more than the testing data, it exhibited a similar performance as the first experiment.

TABLE II
RESULT OF BINARY CLASSIFICATION FOR EXPERIMENT II.

	Simple Neural Network	CNN and LSTM	LSTM
Accuracy	71.75%	93.35%	95.05%
Precision	31.67%	81.63%	87.56%
Recall	60.05%	77.56%	81.97%
F1 Score	41.47%	79.54%	84.67%
FP-rate	25.90%	3.49%	2.33%

Since the LSTM model achieved the best performance in the binary classification subtask, we utilized it to perform sequence prediction for given bags of words in the next step.

C. Result for Sequence Prediction

For the sequence prediction subtask, we first used the trained LSTM model to predict correct sequence orders for the newly added concept names in the March 2018 US Edition of SNOMED CT. Given the computational challenge for testing all possible sequences and the fact that less than 5% of concept names in SNOMED CT are in the length of more than or equal to ten (see Fig. 2 for the distribution of concept names in terms of their lengths), we performed the sequence prediction for concepts whose lengths are less than ten.

For a bag of words, our method will provide a set of candidate concept names. If a candidate name is in a correct order (i.e., a valid name), then it is considered as a true positive case; otherwise, it is considered as a false positive case. If the correct sequence is not included in the set of candidate concept names, we have one false negative case.

Table III shows the performance of our LSTM-based sequence prediction approach for predicting concept names with an overall F1 score of 63.41%. It can be seen that concepts whose names are in the length of two and three received an F1 score of above 80%, concepts in length of four and five received an F1 score of above 60%, concepts in length of six and seven received an F1 score of above 50%, concepts in length of eight received an F1 score of 49.1% , and concepts in length of nine received an F1 score of 39.91%. This indicates that as the length of concept names grows, more sequences (false positive cases) might be included in the candidate set which leads to the decreasing performance. However, even for concept names with length of six, seven or eight, the performance are still acceptable. Overall, the F1 score of our model is 63.41%.

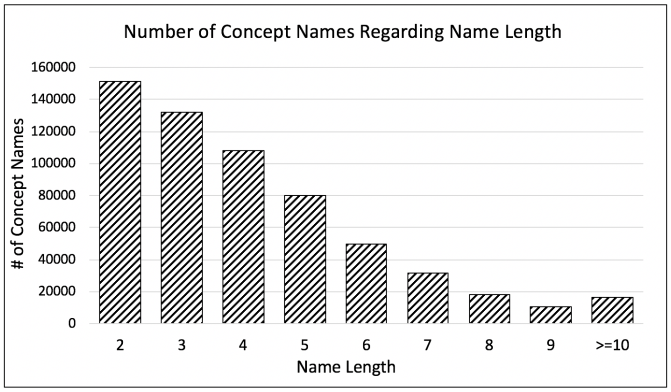

Fig. 2. Number of concept names in terms of the name length for all concepts in the March 2018 US Edition of SNOMED CT.

We also performed another way to evaluate the performance of our LSTM-based sequence prediction approach, by utilizing Cui et al.'s method in [9] to generate bags of words which are necessary to construct the names of new concepts. We obtained 60 bags of words and performed name prediction for them. For each predicted name, a human annotator manually examined whether the generated sequence order is correct and conforming to the name convention of SNOMED CT. Among 60 cases, 44 out of them were considered as correct by the human annotator. Table IV shows our LSTM-based approach achieved an F1 score of 73.95%. The positive examples include "*malignant neoplasm of blood vessel of thorax*", "*structure of layer of tunica vaginalis*" and "*open wound of limb without complication*". This indicates that the LSTM-based method can be applied for naming a new concept based on the bags of words featuring a concept.

D. Potential Factors Affecting the Prediction Performance

There are mainly two factors which potentially affect the performance of our prediction model – the size of bags of words and the words in the bag.

For the first factor, as the length of concept names increases, it becomes more difficult for the model to predict the correct sequence. This is because when the size of a bag of words increases, the number of sequences that need to be classified by the model increases dramatically. Since we assume that there is only one correct concept name for a bag of words, the

TABLE III

Result of LSTM-based sequence prediction in terms of the length of concept names. Training data is from September 2017 US Edition of SNOMED CT and test data is the newly added concepts in the March 2018 Edition.

Length of Concept Name	2	3	4	5	6	7	8	9	All
Number of True Positives	735	1274	909	918	565	446	316	184	5347
Number of False Positives	154	310	556	465	375	278	287	332	2757
Number of False Negatives	155	321	590	665	627	465	368	222	3413
Precision	82.68%	80.43%	62.05%	66.38%	60.11%	61.60%	52.40%	35.66%	65.98%
Recall	82.58%	79.87%	60.64%	57.99%	47.40%	48.96%	46.20%	45.32%	61.04%
F1 Score	82.63%	80.15%	61.34%	61.90%	53.00%	54.56%	49.10%	39.91%	63.41%

TABLE IV

Result of LSTM-based sequence prediction for names of missing concepts identified by Cui et al.'s method in [9].

Number of Concepts Names	60
Number of True Positives	44
Number of False Positives	15
Number of False Negatives	16
Precision	74.58%
Recall	73.33%
F1 Score	73.95%

TABLE V

Result of LSTM-based sequence prediction in terms of whether concept names contain duplicate words or not.

	Without Duplicates	With Duplicates
Number of True Positives	4789	558
Number of False Positives	2224	533
Number of False Negatives	3101	312
Precision	68.29%	51.15%
Recall	60.70%	64.14%
F1 Score	64.27%	56.91%

false positive cases may greatly lower the model's precision. For instance, for a bag of words of size five without duplicate words, it has 120 permutations and only one of them is correct. If we only have two false positive cases, the accuracy is about 98%, however, the actual precision is only 33% and F1 measure is 49%. Thus, length of concept name is an important factor affecting the performance of model. In our experiments, 50% of the testing data are of length that is larger than or equal to five.

The second factor that may affect the model's performance involves the words in the bag. Our model may generate multiple candidate concept names for a given bag of words. This may be because that some words in the bag has the same role in other concept names. For example, for the concept name "*ultrasound guided biopsy of left and right breast*", our model may be confused about the order of words "*left*" and "*right*", because they may appear separately in other concepts but with the same roles or patterns. Another typical case is related to duplicate words in a bag. If a bag contain two (or more) identical words (e.g. multiple "*of*" or multiple "*and*") such as "*MRI of joint of right lower extremity*", it is even harder for the model to decide which word should be attached with the first one, which word should be attached with the second one, and the order of these two parts. Thus in this paper, we also compared the performance of the LSTM-based model applied to the bags of words containing duplicate words and those which do not contain duplicate words. The result is shown in Table V. The model achieved a better F1 score when applied to the bags of words without duplicates.

E. Analysis of False Positives

We manually examined some of the false positive cases in Experiment II of binary classification for potential patterns that our model is not able to deal with. Meanwhile, we compared our sequence predictions for those false positive cases with the ground truth to explore possible causes for mislabeling. Two observed patterns are listed as follows.

The first pattern is related to "*of*". In SNOMED CT, "*B A*" and "*A of B*" are both acceptable names for certain cases. One of them is often defined as the fully specified name (FSN), while the other one is listed as its synonym. For instance, "*cyst of lung*" is an FSN, and "*lung cyst*" is its synonym. However, in some cases, only one of them is included (e.g., concept "*lung mass*" does not have a synonym "*mass of lung*"). Therefore, if a concept name falls into this pattern, our model sometimes cannot predict it correctly.

The second pattern involves two items (e.g., noun or noun phrase) that are connected by a preposition or conjunction, in which case our model sometimes cannot decide which one should be placed first. For example, for the concept "*naproxen sodium and sumatriptan*", our predicted name was "*sumatriptan and naproxen sodium*". Another example is that, for the concept "*fluoroscopy of left and right hip*", our predicted name was "*fluoroscopy of right and left hip*". In such cases, even the predicted names were valid, they were considered as false positives since they differ from the sequence of words provided in the ground truth. In other words, our evaluation was performed in a conservative way.

F. Beyond Naming Purpose

While analyzing false positive cases, we also noticed that this work could identify potential inconsistencies in the naming convention of concepts which can be considered as part of the quality assurance process for biomedical ontologies. For an existing name, we can use our trained model to check if it complies with the naming convention of SNOMED CT. For example, in the March 2018 US Edition of SNOMED CT, a new concept "*Lesion of bone right upper arm*" is added. Our model labeled it as wrong. We found that it is a synonym of "*Lesion of right upper arm bone*". However, when it comes to another similar concept "*Lesion of left lower leg bone*", it does not have a synonym "*Lesion of bone left lower leg*". Instead, it has "*Lesion of bone in left lower leg*" whose pattern does not appear as a synonym of "*Lesion of right upper arm bone*".

Another example is "*Liver of normal size*" that has been added to SNOMED CT in the March 2018 Edition. Our model labeled it as wrong. We found that it is an FSN, and "*Normal sized liver*" is its synonym. However, in other similar concepts such as "*Normal sized tonsils*" and "*Normal sized ear canal*", they are considered as FSNs, but they do not have any synonym that has the pattern "*xx of normal size*". This indicates a name inconsistency. A potential fix is that "*Normal sized liver*" should be the FSN, and the name "*Liver of normal size*" should become inactive. These two examples indicate that our model to some extent can reveal the inconsistency in SNOMED CT names.

IV. Limitations & Future Work

Even though the experiment results showed that our method can provide suggestion in naming concept for biomedical ontologies, there are still some limitations. First, when it comes to training word embedding, even though we have 597,571 concept names which can be used as input for Word2Vec, the context may still not be profound enough (e.g. some words may only appear once or twice) to generate accurate word embedding. Second, our sequence prediction method is based on classifying all possible combinations of words, which is computationally challenging when the length of names is large. Third, the structures of our neural network models are simple, a more complex and well-optimized model to address these limitations is expected.

In future work, we plan to build a more powerful neural network model which utilizes both the word orders within the concept names and the logical definitions of concepts to predict potential concept names. Logical definitions are related to concepts' semantic meanings and thus could potentially be a determinant in how to name concepts. In this case, we no longer need to rely on the bags of words and test all the possible combinations. In addition, as mentioned before, our work may also be used for quality assurance purpose – discovering inconsistencies in concept names. However, in this work, we only manually discovered the inconsistencies. We plan to develop a more systematic way to uncover those inconsistencies.

V. Conclusion

In this paper, we explored three deep learning-based approaches – simple neural network, LSTM, and CNN combined with LSTM, to predict concept names for new concepts in biomedical ontologies. Our experiments showed that the LSTM-based approach achieved the best performance with an F1 score of 63.41% for predicting names for newly added concepts in the March 2018 Edition of SNOMED CT and an F1 score of 73.95% for naming missing concepts identified by Cui et al.'s method in [9]. This indicates that the LSTM-based approach is effective in predicting concept names given bags of words. Further analysis of the false positive cases revealed that this work may also be leveraged for identifying potential inconsistencies within the concept names of SNOMED CT.

References

[1] O. Bodenreider, "Biomedical ontologies in action: role in knowledge management, data integration and decision support," *Yearbook of medical informatics*, p. 67, 2008.

[2] R. Hoehndorf, P. N. Schofield, and G. V. Gkoutos, "The role of ontologies in biological and biomedical research: a functional perspective," *Briefings in bioinformatics*, vol. 16, no. 6, pp. 1069–1080, 2015.

[3] L. Cui, S. Tao, and G.-Q. Zhang, "Biomedical ontology quality assurance using a big data approach," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 10, no. 4, p. 41, 2016.

[4] "Snomed ct international release management home," https://confluence.ihtsdotools.org/display/RMT/, [Online; Accessed May, 2018].

[5] Z. He, J. Geller, and Y. Chen, "A comparative analysis of the density of the snomed ct conceptual content for semantic harmonization," *Artificial intelligence in medicine*, vol. 64, no. 1, pp. 29–40, 2015.

[6] Z. He, Y. Chen, S. de Coronado, K. Piskorski, and J. Geller, "Topological-pattern-based recommendation of umls concepts for national cancer institute thesaurus," in *AMIA Annual Symposium Proceedings*, vol. 2016. American Medical Informatics Association, 2016, p. 618.

[7] G. Jiang and C. G. Chute, "Auditing the semantic completeness of snomed ct using formal concept analysis," *Journal of the American Medical Informatics Association*, vol. 16, no. 1, pp. 89–102, 2009.

[8] W. Zhu, G. Zhang, and L. Cui, "Spark-mca: Large-scale, exhaustive formal concept analysis for evaluating the semantic completeness of snomed ct," in *AMIA Annual Symp Proc*, 2017, pp. 1914–23.

[9] 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," *Journal of the American Medical Informatics Association*, vol. 24, no. 4, pp. 788–798, 2017.

[10] R. Abeysinghe, M. A. Brooks, J. Talbert, and C. Licong, "Quality assurance of nci thesaurus by mining structural-lexical patterns," in *AMIA Annual Symposium Proceedings*, vol. 2017. American Medical Informatics Association, 2017, p. 364.

[11] L. Deng, D. Yu *et al.*, "Deep learning: methods and applications," *Foundations and Trends® in Signal Processing*, vol. 7, no. 3–4, pp. 197–387, 2014.

[12] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[13] R. Rehurek and P. Sojka, "Software framework for topic modelling with large corpora," in *In Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. Citeseer, 2010.

[14] C. D. M. Jeffrey Pennington, Richard Socher, "Keras," https://keras.io, 2015.

[15] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, "Lstm: A search space odyssey," *IEEE transactions on neural networks and learning systems*, vol. 28, no. 10, pp. 2222–2232, 2017.

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[17] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A convolutional neural network for modelling sentences," *arXiv preprint arXiv:1404.2188*, 2014.

[18] D. Ciregan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification," in *Computer vision and pattern recognition (CVPR), 2012 IEEE conference on*. IEEE, 2012, pp. 3642–3649.

[19] G. D. Forney, "The viterbi algorithm," *Proceedings of the IEEE*, vol. 61, no. 3, pp. 268–278, 1973.

[20] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," in *Soviet physics doklady*, vol. 10, no. 8, 1966, pp. 707–710.