Proceedings of the Second International Workshop on Contextualized Knowledge Graphs (CKG 2019)
https://wiki.foodmedy.com/index.php?title=CKG2019

Construction of UMLS Metathesaurus with Knowledge-Infused Deep Learning

Hong Yung Yip[1], Vinh Nguyen[2], Olivier Bodenreider[2]

[1] Artificial Intelligence Institute, University of South Carolina, Columbia, SC, USA
[2] National Library of Medicine, National Institute of Health, Bethesda, MD, USA
hyip@email.sc.edu[1], vinh.nguyen@nih.gov[2], obodenreider@mail.nih.gov[2]

Abstract. The Unified Medical Language System (UMLS) is a Metathesaurus of biomedical vocabularies developed to integrate a variety of ways the same concepts are expressed by different terminologies and to provide cross-walk among them. However, the current process of constructing and inserting new resources to the existing Metathesaurus relies heavily on lexical knowledge, semantic pre-processing, and manual audits by human editors. This project explores the use of supervised Deep Learning approach to identify synonymy and non-synonymy among English UMLS concepts at the atom level. We use a Siamese network with Long Short-Term Memory and Convolutional Neural Network models to learn the similarities and dissimilarities between pairs of atoms from the active subset of 2019AA UMLS. To disambiguate concepts with lexically identical atoms, we contextualize the pairs with various enrichment strategies that reflect the information available to the UMLS editors including the source synonymy, hierarchical context, and source semantic group. Learning from base lexical features of the atoms yields an overall F1-score of 75.97%. Infusing source synonymy to the base yields a higher precision and overall F-1 score of 86.54% and 87.63% respectively. Whereas, infusing hierarchical context trades precision for higher recall of 90.38%. Infusing source synonymy, hierarchical context, and semantic group provides an overall increase in accuracy to 95.20%. However, infusing source synonymy of hierarchical context does not yield any noticeable improvement. A knowledge-infused learning approach provides a good performance indicating promising potential for emulating the current building process. Future works include evaluation with rule-based normalization approach of constructing the Metathesaurus and investigation of the applicability, maintenance, and scalability of these models.

Keywords: Unified Medical Language System · Semantic Similarity · Deep Learning · Contextualized Knowledge Graph

1 Introduction

The Unified Medical Language System (UMLS) is a rich repository of biomedical vocabularies developed by the US National Library of Medicine. It is an effort to overcome challenges to effective retrieval of machine-readable information. One of which is the variety of ways the same concepts are expressed by different terminologies [1]. For example, the concept of "Addison's Disease" is expressed as "Primary hypoadrenalism" in the *Medical Dictionary for Regulatory Activities* (MedDRA) and as "Primary adrenocortical insufficiency" in the *10th revision of the International Statistical Classification of Diseases and Related Health Problems* (ICD-10). The lack of integration between these synonymous terms often leads to poor interoperability between information systems (i.e. how does one map a concept from one terminology to another) and confusion among health professionals. Hence, the UMLS aims to integrate and provide cross-walk among various terminologies as well as facilitate the creation of more effective and interoperable biomedical information systems and services, including electronic health records [3]. Till date, it is increasingly being used in areas such as patient care coordination, clinical coding, information retrieval, and data mining. There are three components to the UMLS Knowledge Sources: the Metathesaurus, the Semantic Network, and the SPECIALIST Lexicon and Lexical Tools.

The Metathesaurus is a vocabulary database organized by concept or meaning. It is built from the electronic versions of various thesauri, code sets, classifications, and lists of controlled terms used in biomedical, clinical, and health services, known as "terminologies" or interchangeably as "source vocabularies". It connects alternative names (i.e. name variants) that are considered to be synonymous under the same concept and identifies useful relationships between various concepts [1]. Concepts are assigned at least one Semantic Type from the Semantic Network to provide semantic categorization. The Lexical Tools provide lexical information for language processing such as identifying string variants and providing normalization as normalized string indexes to the Metathesaurus. As of May 6, 2019, the 2019AA release of the UMLS Metathesaurus contains approximately 3.85 million biomedical and health-related concepts and 14.6 million concept names from 210 source vocabularies including the *National Center for Biotechnology Information* (NCBI) taxonomy, *Systematized Nomenclature of Medicine - Clinical Terms* (SNOMED CT), Gene Ontology, the *Medical Subject Headings* (MeSH), and OMIM[4].

1.1 Construction of the UMLS Metathesaurus

The current approach of building the Metathesaurus relies on the use of lexical knowledge, semantic pre-processing, and UMLS human editors. The core idea is that synonymous terms originating from different source vocabularies are clustered into a concept with a preferred term and a Concept Unique Identifier (CUI). The basic building block of the Metathesaurus, also known as an "atom", is a concept string from each of the source vocabularies. Simply put, each occurrence of a string in each source vocabulary is assigned a unique atom identifier

[3] https://www.nlm.nih.gov/research/umls/index.html

[4] https://www.nlm.nih.gov/research/umls/knowledge_sources/metathesaurus/release/notes.html

(AUI). When a lexically identical string appears in multiple source vocabularies for example "Headache" appearing in both MeSH and ICD-10, they are assigned different AUIs. These AUIs are then linked to a single string identifier (SUI) to represent occurrences of the same string. Each SUI is linked to all of its English lexical variants (detected using the Lexical Variant Generator tool) by a common term identifier (LUI). These LUIs may subsequently be linked to more than one CUI due to strings that are lexical variants of each other have different meanings. Table 1 illustrates how synonymous terms are clustered into a CUI.

Table 1. Metathesaurus AUI, SUI, LUI, and CUI

String (Source)	**AUI**	**SUI**	**LUI**	**CUI**
Headache (MeSH)	A0066000	S0046854	L0018681	C0018681
Headache (ICD-10)	A0065992			
Headaches (MedDRA)	A0066007	S0046855		
Headaches (OMIM)	A12003304			
Cephalodynia (MeSH)	A0540936	S0475647	L0380797	

In addition, some source vocabularies provide source synonyms, hierarchical and non-hierarchical relationships as well as metadata information for semantic pre-processing. The UMLS human editors are involved to associate concepts and perform manual reviews [1]. These processes of constructing and inserting new resources to the existing Metathesaurus from identifying lexical variants to manual audits by domain experts can be both arduous and time-consuming given the current size of Metathesaurus comprises of over 3.85 million concepts. Given the recent successes of supervised Deep Learning (DL) approaches in their applications to the medical and healthcare domains [2], we hypothesize that these DL models can be trained to emulate the current building process.

1.2 Supervised Deep Learning

Supervised DL is a learning function that maps an input to an output based on examples of input-output pairs through layers of dense networks [3]. The Metathesaurus comprises of approximately 10 million English atoms with each assigned a CUI. One can simply train a supervised classifier to predict which CUI should be labeled to a "new" atom (since atoms having the same CUI are synonymous) as an approach to insert new resources to the current Metathesaurus. However, this approach is considered as an extreme classification task [4] due to the huge prediction space of 3.85 million CUIs. Nonetheless, the CUI is merely a "mechanism" to cluster synonymous terms under the same "bucket". We are primarily interested in whether two atoms are synonymous and hence be labeled with the same CUI regardless of whether this CUI has already existed in the Metathesaurus. Hence, this project is modeled as a similarity task where we want to assess similarity based not only on the lexical features of an atom but also based on its context (represented by the lexical features of neighboring concepts in this source vocabulary). Concretely, a fully-trained model should identify and learn scenarios where

1. Atoms that are **lexically similar** in nature but are **not synonymous**, e.g., "Lung disease and disorder" versus "Head disease and disorder"
2. Atoms that are **lexically dissimilar** but are **synonymous**, e.g., "Addison's disease" versus "Primary adrenal deficiency"

Similarity assessment between words and sentences, also known as Semantic Text Similarity (STS) task is an active research area in Natural Language Processing (NLP) due to its crucial role in various downstream tasks such as information retrieval, machine translation, and in our case, synonyms clustering. The STS task can be expressed as follows: given two sentences, a system returns a probability score of 0 to 1 indicating the degree of similarity. STS is a challenging task due to the inherent complexity in language expressions, word ambiguities, and variable sentence lengths. Traditional approach relies on hand-engineering lexical features (e.g. word overlap and subwords [5], syntactic relationship [6], structural representations [7]), linguistic resources (e.g. corpora), bag-of-words and term frequency–inverse document frequency (TF-IDF) models that incorporate a variety of similarity measures [8] for example string-based [9] and term-based [10]. However, most are syntactically and semantically constrained. Recent successes in STS [11] in predicting sentence similarity and relatedness have been obtained by using corpus-based [12] and knowledge-based similarity, e.g. word embedding for feature representation [13] with supervised DL approaches, e.g. Siamese Network with Recurrent Neural Network (RNN) [14] and Convolutional Neural Networks (CNN) [15] to perform deep analysis of words and sentences to learn the necessary semantics and structure.

1.3 Siamese Recurrent Architecture

Contrary to the traditional neural network which takes in one input at a time, the Siamese network is an architecture that takes in a pair of inputs and learns representations based on the explicit similarity and dissimilarity information (i.e. the pair of similar and dissimilar inputs) [16]. It was originally used for signature verification [16] and has since been applied to various applications such as face verification [17], unsupervised acoustic modeling [18], and learning semantic entailment [14] as well as text similarity [19]. A series of DL models can be incorporated within the Siamese architecture. RNN is a type of DL model that excels at processing sequential information due to the presence of memory cell to store and "remember" data read over time [20]. Another variant of RNN is the Long Short-Term Memory (LSTM). It enhances the standard RNN to handle long-term dependencies and to minimize the inherent vanishing gradient problem of RNN with the introduction of "gates" (input, output and forget gates) to control the flow of and retain information better through time. It is more accurate in handling long sequences, however, it comes at the cost of higher memory consumption and slower training times compared to standard RNN which is faster but less accurate. Nonetheless, a combination of Siamese network with RNN and LSTM have been applied to various NLP tasks including similarity assessment with great success [14,21,22]. On the other hand, CNN (another type of DL model) has also performed well in NLP due to its ability to extract distinctive features at a higher granularity [23]. A Siamese CNN model learns sentence embedding and predicts sentence similarity with features from various convolution and pooling operations [24].

In this paper, we explore the use of DL, specifically the Siamese recurrent architecture with a combination of LSTM and CNN for the following contributions:

1. Identify synonymy and non-synonymy among English UMLS concepts at the atom level (i.e. given two English atoms, are they synonymous and thus belong to the same CUI?)
2. Investigate whether the DL approach could emulate the current Metathesaurus building process

2 Methodology

The scope of this project can be divided into four components: (i) retrieving and parsing the UMLS dataset, (ii) generating features for learning, (iii) designing the Siamese architecture, and (iv) evaluating the Siamese network with different data enrichment strategies (i.e., **infusing various knowledge** provided by the source vocabularies).

2.1 Dataset

We use the active subset of the 2019AA UMLS and remove the derivative, duplicative, and spelling variants sources. The final dataset consists of **9,533,853 atoms** grouped into **3,793,516 CUIs**. Table 2 shows the sources removed.

Table 2. Sources Removed

Sources Removed	**Sources**
Derivative and Duplicative	NCI_BRIDG, NCI_BioC, NCI_CDC, NCI_CDISC, NCI_CDISC-GLOSS, NCI_CPTAC, NCI_CRCH, NCI_CTCAE, NCI_CTCAE_3, NCI_CTCAE_5, NCI_CTEP-SDC, NCI_CTRP, NCI_CareLex, NCI_DCP, NCI_DICOM, NCI_DTP, NCI_EDQM-HC, NCI_FDA, NCI_GAIA, NCI_GENC, NCI_ICH, NCI_INC, NCI_JAX, NCI_KEGG, NCI_NCI-GLOSS, NCI_NCI-HGNC, NCI_NCI-HL7, NCI_NCPDP, NCI_NICHD, NCI_PI-RADS, NCI_PID, NCI_RENI, NCI_UCUM, NCI_ZFin, HCDT, HCPT, ICPC2P, LCH_NW
Spelling Variants	ICD10AE, ICD10AMAE, MTHICPC2EAE, MTHICPC2ICD10AE

2.2 Feature Engineering

The goal is to learn the similarities between pairs of atoms within a CUI and dissimilarities between pairs of atoms from different CUIs. Prior to generating the positive and negative pairs, we preprocess the lexical features of the atoms similar to how [25] preprocess their dataset (remove all punctuation except hyphen, lowercase, and tokenize by space) to ensure conformity as we leverage their pre-trained BioWordVec embedding in our downstream network (Section 2.4).

Synonyms. We generate positive pairs based on CUI-asserted synonymy between atoms. Table 3 shows examples of positive pairs generated from one CUI.

Table 3. Positive Pairs from a Single CUI

CUI	**Atom**
C0001403	Addison disease Primary hypoadrenalism Primary adrenocortical insufficiency Addison's disease (disorder)
Positive Pairs	
Addison disease	Primary hypoadrenalism
Addison disease	Primary adrenocortical insufficiency
Addison disease	Addison's disease (disorder)
Primary hypoadrenalism	Primary adrenocortical insufficiency
Primary hypoadrenalism	Addison's disease (disorder)
Primary adrenocortical insufficiency	Addison's disease (disorder)

Non-Synonyms. On the contrary, it is computationally infeasible, time and space complexities wise, to generate all the negative pairs, which is approximately 9.5 million atoms squared since it is one atom against all other atoms from non-related CUIs. In addition, the class imbalance between positive and negative will induce learning bias in which the model will suffer from lower precision in detecting synonyms due to a higher preference towards non-synonyms. Intuitively, we want the DL model to learn interesting negative pairs that are lexically similar but differ in semantics. Hence, we adopt a heuristic approach to reduce the sample space where we compute Jaccard index between atoms to include only negative pairs with high Jaccard similarity from different CUIs (with a cut-off threshold of 0.6 Jaccard index) (Table 4). The pairs are then sorted from the highest to lowest Jaccard index and the number of inclusion pairs is shown in Table 5. The final dataset consists of pairs of strings sampled in a 1:1, 3:1, 4:1, 6:1, and 10:1 ratio of between-CUI (negative) pairs to within-CUI (positive) pairs. These ratios are adopted from [18,19] for Siamese networks.

$$JaccardIndex(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|} \tag{1}$$

Table 4. Jaccard Computation on a Pair of Atom from Different CUIs

C0000473	**C0038784**
Product containing ***para-aminobenzoic*** acid	Product containing ***sulfuric*** acid
Jaccard Index = Intersection (3)/ Union (5) = 0.6	

Table 5. Final Dataset Size

Feature	**Number of Pairs**
Synonyms	15,647,133
Ratio of between-CUI non-synonym pairs to within-CUI synonym pairs	
1:1	15,647,133
3:1	46,941,399
4:1	62,588,532
6:1	93,882,798
10:1	156,471,330

2.3 Experiments

The entry point of our experiment is the lexical features of an atom. However, in order to disambiguate concepts with lexically identical atoms, e.g. the concept **"nail"** with CUI "C0222001" and "C0021885" shown in Figure 1, there is a need to contextualize the two different "nail" concepts (denoted by two distinct CUIs) with additional features/ knowledge that indicate different meanings. Hence, we compose the experiments (Table 6) with different data enrichment strategies i.e. **infusing various knowledge** that reflect the information available to the UMLS editors during manual construction of the Metathesaurus including the source synonymy, hierarchical context, and source semantic group.

Table 6. Five Experimental Setup

Experiment	**Features**
1	Base Atom Lexical Features
2	Base Atom Lexical Features + Source Synonymy
3	Base Atom Lexical Features + Hierarchical Context + Semantic Group
4	Base Atom Lexical Features + Source Synonymy + Hierarchical Context + Semantic Group
5	Base Atom Lexical Features + Source Synonymy + Hierarchical Context + Hierarchical Source Synonymy + Semantic Group

Base. The base consists of only the lexical features of an atom for all synonym (positive) and non-synonym (negative) pairs.

Source synonymy. Some source vocabularies provide synonyms to the atoms which enrich the original atom with additional lexical features that are synonymous. We generate these source synonyms based on the Source Concept Unique Identifier (SCUI) of each atom.

Hierarchical context. Some source vocabularies provide hierarchical relationships (ancestor-descendant or parent-child or broader-narrow relations) which extend the original atom with surrounding contexts. We generate the hierarchical context using the unique lexical features of immediate (1-level) parents and children based on the source relations.

Semantic group. The semantic group provides an additional layer of high-level semantic categorization to an atom. Figure 1 shows the two concepts "nail" are syntactically similar but they differ in semantics in which one refers to "anatomy" and another refers to the "devices". We assign semantic group based on the second-level concept from the root node of the original atom as a proxy to semantic categorization. For source vocabularies that do not provide hierarchical relationships, we assign a semantic group to the best knowledge of the human editors to the source of these atoms.

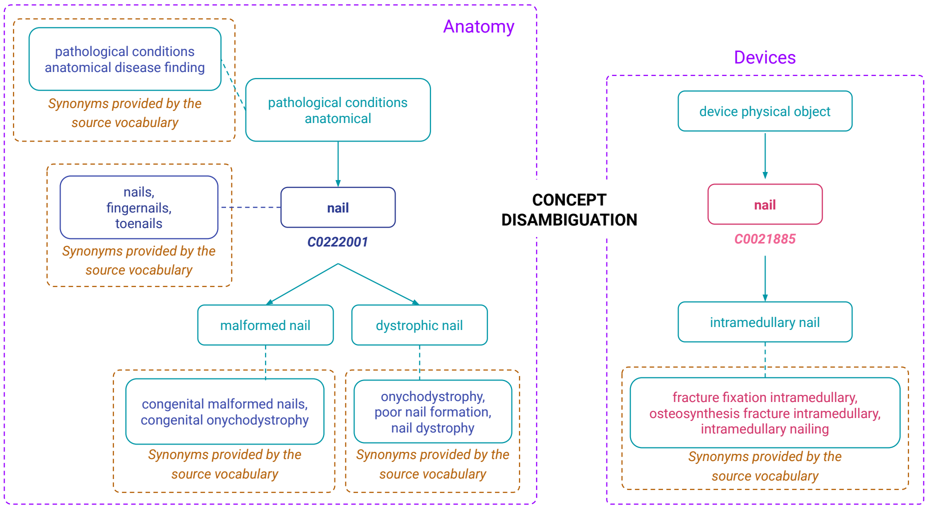
Anatomy
pathological conditions anatomical disease finding
Synonyms provided by the source vocabulary
pathological conditions anatomical
nails, fingernails, toenails
Synonyms provided by the source vocabulary
nail
C0222001
malformed nail
dystrophic nail
congenital malformed nails, congenital onychodystrophy
Synonyms provided by the source vocabulary
onychodystrophy, poor nail formation, nail dystrophy
Synonyms provided by the source vocabulary
CONCEPT DISAMBIGUATION
Devices
device physical object
nail
C0021885
intramedullary nail
fracture fixation intramedullary, osteosynthesis fracture intramedullary, intramedullary nailing
Synonyms provided by the source vocabulary

Fig. 1. Concepts Disambiguation. The dotted brown boxes indicate source synonymy and the green boxes indicate hierarchical contexts. The dotted purple boxes indicate source semantic group.

2.4 Siamese Models

Two different Siamese Models are designed: the Siamese LSTM and Siamese CNN-LSTM.

Siamese LSTM. This model adopts the Siamese structure from [14] (Figure 2). A pair of atoms are first transformed into their respective numerical word representations, i.e. embedding of word vectors. A word embedding is a language modeling and feature learning techniques in NLP where words are mapped to vectors of real numbers with varying dimensions. These word vectors are positioned in the vector space in a manner where words that share similar contexts in the corpus are situated close to one another in the space [26]. Instead of training the word vectors from scratch, we leverage the pre-trained biomedical word embedding (BioWordVec-intrinsic) with dimension size of 200 per word vector that is trained on PubMed text corpus and MeSH data [25]. The rationale is to "precondition" the Siamese network with prior knowledge of the inherent similarity between words in the UMLS vocabulary. Upon plotting a word length distribution, approximately 97% of atoms in the UMLS have a word length of lesser or equal to 30. Hence, we apply padding or truncation to restrict the word length of each atom to a maximum of length 30 to ensure a uniformity in dimension to speed up the training process. The embedding of the pair of atoms are fed to $LSTM_A$ and $LSTM_B$ which each processes one of the atoms in the given pair and consists of 50 hidden learning units. These units learn the specific semantic and syntactic features based on word orders of each individual atoms through time. The output of the model is a Manhattan distance similarity function, $exp(-||LSTM_A - LSTM_B||_1) \in [0, 1]$, a function that is well-suited for high dimensional space [27]. We apply this model to Experiment 1.

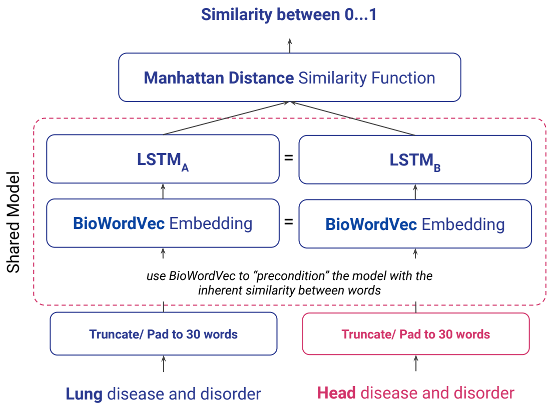
Similarity between 0...1
Manhattan Distance Similarity Function
Shared Model
LSTM_A
=
LSTM_B
BioWordVec Embedding
=
BioWordVec Embedding
use BioWordVec to "precondition" the model with the inherent similarity between words
Truncate/ Pad to 30 words
Truncate/ Pad to 30 words
Lung disease and disorder
Head disease and disorder

Fig. 2. The Siamese LSTM Model. Both left and right branch of the model share the same weights of all the layers.

Siamese CNN-LSTM. We use this model for Experiment 2, 3, 4, and 5 to infuse the additional knowledge and features: source synonymy, hierarchical context, and semantic group information. This model adopts the Siamese structure from [28] (Figure 3). It differs from the first architecture in its hidden learning layers. For this model, instead of having only one embedding from the lexical features of the atoms, we concatenate two extra vectors learned from the embedding that represents the extra context information to the original atom vector. To generate the "context bag", we extract 60 unique lexical features from source synonyms and/or hierarchical context to enrich the base features of an atom and sort them in alphabetical order to minimize word order randomness as the word order is less prioritized prior to transforming them into a context embedding. We apply one layer of CNN with 100 filters and a window size of 5 [28] with batch normalization (to reduce overfitting) to extract and generate an intermediary representation and subsequently apply a layer of LSTM with 50 hidden learning units to learn these features. Similarly, the semantic group information is "infused" by transforming it using BioWordVec embedding and subsequently feeding it to a layer of LSTM with 50 hidden units. The outputs of each LSTM layer (base, context, and semantic group) are averaged over time and these three 50-dimensional vectors are concatenated and used as input to a 2-layer dense Fully Connected (FC) network with learning units of 128 and 50 respectively and Manhattan distance similarity function, $exp(-||FC_A - FC_B||_1) \in [0, 1]$, as the final output layer. The parameters of both models are optimized using the Adam method [29].

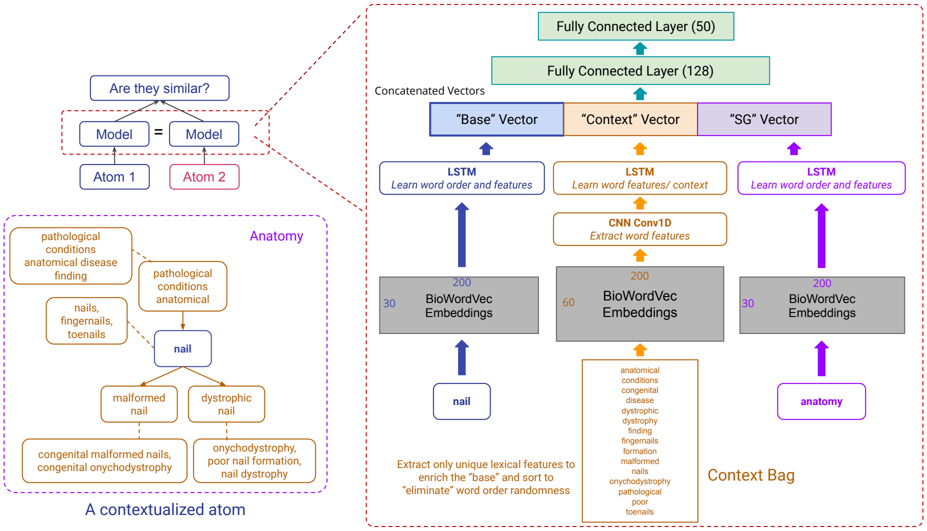
Are they similar?
Model
=
Model
Atom 1
Atom 2
Anatomy
pathological conditions anatomical disease finding
pathological conditions anatomical
nails, fingernails, toenails
nail
malformed nail
dystrophic nail
congenital malformed nails, congenital onychodystrophy
onychodystrophy, poor nail formation, nail dystrophy
A contextualized atom
Fully Connected Layer (50)
Fully Connected Layer (128)
Concatenated Vectors
"Base" Vector
"Context" Vector
"SG" Vector
LSTM
Learn word order and features
LSTM
Learn word features/ context
LSTM
Learn word order and features
CNN Conv1D
Extract word features
200
30
BioWordVec Embeddings
200
60
BioWordVec Embeddings
200
30
BioWordVec Embeddings
nail
anatomy
anatomical
conditions
congenital
disease
dystrophic
dystrophy
finding
fingernails
formation
malformed
nails
onychodystrophy
pathological
poor
toenails
Extract only unique lexical features to enrich the "base" and sort to "eliminate" word order randomness
Context Bag

Fig. 3. The Siamese CNN-LSTM Model. Similarly, both left and right branch of the model share the same weights of all the layers.

Each experiment (Experiment 1, 2, 3, 4, 5) is trained against five various proportions (1:1, 3:1, 4:1, 6:1, and 10:1 ratio) of negative to positve pairs independently for 20 epochs and validated with 5-fold cross-validation with Biowulf Cluster from the National Institute of Health (NIH) High-Performance Computing (HPC) Systems using a mix of Nvidia Tesla P100 and V100 graphical processing unit. A set of experiments are conducted prior on a small data set (training and validation size of 100,000 and 20,000 respectively) to gauge the performance and desired capabilities of the models as well as to fine-tune the hyper-parameters with different incremental range (e.g. learning rate with a range of 0.0005 to 0.001, batch size with a range from 128 to 512). Table 7 summarizes the final set of parameters and hyper-parameters that are used for Siamese LSTM (baseline experiment 1) and Siamese CNN-LSTM (enriched experiment 2, 3, 4, and 5) respectively.

Table 7. The Set of Parameters used for Siamese LSTM and Siamese CNN-LSTM respectively.

Parameters/ Hyperparameters	**Siamese LSTM**	**Siamese CNN-LSTM**
Framework	Keras 2.0 with Tensorflow backend	
Word Vector Size	200	
Maximum Input Length	30	
Maximum Context Input Length	-	60
Embedding	BioWordVec	
LSTM Hidden Units	50	
LSTM Activation	Tanh	
CNN Filters	-	100
CNN Window Size	-	5
CNN Activation	-	ReLU with batch normalization
Fully Connected Layer 1	-	128 units with ReLU activation
Fully Connected Layer 2	-	50 units with ReLU activation
Weights and Biases	Random Initialization	
Optimizer	Adam	
Learning Rate	0.001	
Loss Function	Mean Squared Error (MSE)	
Batch Size	128	
Number of Training Epochs	20	
Validation	5-fold cross-validation	

3 Results and Evaluations

We evaluate the performance of the models in terms of validation accuracy, precision, recall, overall F1-Score, specificity, sensitivity, and false-positive rate. Out of all the various proportions of negative to positive pairs, the 6:1 ratio achieves the best performance in terms of **validation accuracy** in identifying and classifying synonyms and non-synonyms. Table 8 shows the full performance metrics achieved by the 6:1 ratio of negative to positive pairs and Table 9 shows various examples of true positives and true negatives correctly identified, false positives identified, and false negatives not identified by experiment 5.

Table 8. Performance of the 6:1 Ratio of Negative to Positive Pairs

Model/ Performance Metrics	Siamese LSTM	Siamese CNN-LSTM			
	Exp. 1	Exp. 2	Exp. 3	Exp. 4	Exp. 5
	Base	Base + SS	Base + HC + SG	Base + SS + HC + SG	Base + SS + HC + HSS + SG
Accuracy	0.93333	0.8720	0.9486	0.9520	**0.9541**
Precision	0.7828	**0.8654**	0.7643	0.8296	0.8009
Recall	0.7379	0.8874	0.8381	**0.9038**	0.8978
F1-Score	0.7597	**0.8763**	0.7995	0.8428	0.8466
Specificity	0.9659	0.8560	**0.9640**	0.9601	0.9633
Sensitivity	0.7379	0.8874	0.8381	**0.9038**	0.8978
False Positive Rate	**0.0341**	0.1440	0.0360	0.0399	0.0367

Exp.: Experiment, **SS**: Source Synonymy, **HC**: Hierarchical Context, **SG**: Semantic Group, **HSS**: Hierarchical Source Synonymy

4 Discussion

Based on Table 8, we observe that using only the lexical features of atom yields an overall F1-score of 75.97%. Infusing source synonymy to the base yields a higher precision and overall F-1 score of 86.54% and 87.63% respectively. Whereas, infusing hierarchical context trades precision for higher recall of 90.38%. Infusing source synonymy, hierarchical context, and the semantic group gives an overall boost to the accuracy of 95.20%. However, infusing source synonymy of hierarchical context does not yield any noticeable improvement. Some of the plausible explanations are synonyms provided by the source are closely related and they are alternative variants to the base atom, hence the higher precision. Whereas, hierarchical contexts or parents and children relationships represent broader and narrower relations that encompass a wider variety of lexical features to the base

Table 9. Examples of True Positives, True Negatives, False Positives, and False Negatives from Experiment 5

True Positives (Synonyms) Correctly Identified	
nail clipper	cutters nail
injury of salivary gland	salivary gland injury
avulsion	fracture sprain
True Negatives (Non-synonyms) Correctly Identified	
fingernail	infection of fingernail
product containing only **iron** medicinal product	product containing only **levorphanol** medicinal product
medical and surgical gastrointestinal system insertion **ileum** via natural or artificial opening endoscopic **infusion** device	medical and surgical gastrointestinal system revision **stomach** via natural or artificial opening endoscopic **other** device
False Positives (Non-synonyms) Identified	
finding of **wrist** joint	finding of **knee** joint
malignant neoplasm of upper limb	malignant neoplasm of **muscle of** upper limb
skin **wound** of axillary fold	skin **cyst** of axillary fold
False Negatives (Synonyms) Not Identified	
hla antigen	human leukocyte antigen
pyelotomy	incision of renal pelvis treatment
routine cervical smear	screening for malignant neoplasm of cervix

atom, hence the higher recall. However, extending the hierarchical context to include the source synonymy of the parents and children atoms may be overstretched from the original semantics of the base atom and the model may perceive them as noise.

Based on Table 9, we observe the performance of the trained Siamese model from Experiment 5 on real-scenario examples. With the incorporation of LSTM, the model is able to handle both short and long sequences as well as learn the positional variants of the atoms, e.g. "injury of salivary gland" versus "salivary gland injury". Combining with CNN, the model is able to extract and learn pairs that are lexically similar in nature but are not synonymous, e.g., "product containing only iron medicinal product" versus "product containing only levorphanol medicinal product" and vice versa, atoms that are lexically dissimilar but are synonymous, e.g., "avulsion" versus "fracture sprain". Nonetheless, for words that are closely related to each other semantically such as "wrist" and "knee", and "wound" and "cyst", the model fails to recognize them as non-synonyms. In addition, the model fails to identify synonyms with lexical features that are rare such as "pyelotomy" which indicates that there is still room for fine-tuning the model e.g. expanding the capability of the current architecture to learn from more examples.

5 Conclusion

In conclusion, this study demonstrates the feasibility of using DL to identify synonymy and non-synonymy among atoms with relatively good performance indicating a promising potential for emulating the current Metathesaurus building process. In addition, a **knowledge-infused DL approach** leveraging multiple streams of knowledge provides the necessary contextualization to disambiguate lexically identical features and achieves an overall higher performance compared to vanilla DL approach. Future works include (a) evaluations with the manual rule-based normalization process of constructing the Metathesaurus since the current evaluations are done within the scope of DL, i.e. evaluating whether infusing additional knowledge (features) provide better performance, but not between the traditional and automatic building process, and (b) investigation of the scalability, maintenance, and applicability aspects of these models to complement the current lexical processing and the UMLS human editors.

6 Acknowledgment

This work was supported by the Intramural Research Program of the NIH, National Library of Medicine. The authors would like to extend the acknowledgment to Paul Fontelo (NIH) for orchestrating the Summer Internship Training Program. The UMLS dataset used in this study can be retrieved with a UMLS license at `https://www.nlm.nih.gov/research/umls/licensedcontent/umlsknowledgesources.html`.

References

1. Bodenreider, O.: The Unified Medical Language System (UMLS): integrating biomedical terminology. Nucleic Acids Research. 32, 267D-270 (2004)
2. Esteva, A., Robicquet, A., Ramsundar, B., Kuleshov, V., DePristo, M., Chou, K., Cui, C., Corrado, G., Thrun, S., Dean, J.: A guide to deep learning in healthcare. Nature Medicine. 25, 24-29 (2019)
3. Russell, S., Norvig, P.: Artificial intelligence: a modern approach. (2009)
4. Bengio, S., Dembczyński, K., Joachims, T., Kloft, M., Varma, M.: Extreme Classification.
5. Lai, A., Hockenmaier, J.: A denotational and distributional approach to semantics. In Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), pp. 329-334. (2014)
6. Zhao, J., Zhu, T., Lan, M.: Ecnu: One stone two birds: Ensemble of heterogeneous measures for semantic relatedness and textual entailment. In Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), pp. 271-277. (2014)
7. Severyn, A., Nicosia, M., Moschitti, A.: Learning semantic textual similarity with structural representations. In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pp. 714-718. (2013)
8. Gomaa, W. H., Fahmy, A. A.: A survey of text similarity approaches. International Journal of Computer Applications. 68(13), 13-18 (2013)

9. Hall, P. A., Dowling, G. R.: Approximate string matching. ACM computing surveys (CSUR). 12(4), 381-402 (1980)
10. Salton, G., Buckley, C.: Term-weighting approaches in automatic text retrieval. Information processing & management. 24(5), 513-523 (1988)
11. Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., Dean, J.: Distributed representations of words and phrases and their compositionality. In Advances in neural information processing systems, pp. 3111-3119. (2013)
12. Landauer, T. K., Dumais, S. T.: A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. Psychological review. 104(2), 211 (1997)
13. Mihalcea, R., Corley, C., Strapparava, C.: Corpus-based and knowledge-based measures of text semantic similarity. In AAAI (Vol. 6, No. 2006), pp. 775-780. (2006)
14. Mueller, J., Thyagarajan, A.: Siamese recurrent architectures for learning sentence similarity. In Thirtieth AAAI Conference on Artificial Intelligence. (2016)
15. He, H., Gimpel, K., Lin, J.: Multi-perspective sentence similarity modeling with convolutional neural networks. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 1576-1586. (2015)
16. Bromley, J., Guyon, I., LeCun, Y., Säckinger, E., Shah, R.: Signature verification using a "siamese" time delay neural network. In Advances in neural information processing systems, pp. 737-744. (1994)
17. Chopra, S., Hadsell, R., LeCun, Y.: Learning a similarity metric discriminatively, with application to face verification. In CVPR (1), pp. 539-546. (2005)
18. Synnaeve, G., Dupoux, E.: A temporal coherence loss function for learning unsupervised acoustic embeddings. Procedia Computer Science. 81, 95-100 (2016)
19. Neculoiu, P., Versteegh, M., Rotaru, M.: Learning text similarity with siamese recurrent networks. In Proceedings of the 1st Workshop on Representation Learning for NLP, pp. 148-157. (2016)
20. Rychalska, B., Pakulska, K., Chodorowska, K., Walczak, W., Andruszkiewicz, P.: Samsung Poland NLP Team at SemEval-2016 Task 1: Necessity for diversity; combining recursive autoencoders, WordNet and ensemble methods to measure semantic similarity. In Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016), pp. 602-608. (2016)
21. Greff, K., Srivastava, R. K., Koutník, J., Steunebrink, B. R., Schmidhuber, J.: LSTM: A search space odyssey. IEEE transactions on neural networks and learning systems. 28(10), 2222-2232 (2016)
22. Tai, K. S., Socher, R., Manning, C. D.: Improved semantic representations from tree-structured long short-term memory networks. arXiv preprint arXiv:1503.00075. (2015)
23. Kim, Y.: Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882. (2014)
24. He, H., Gimpel, K., Lin, J.: Multi-perspective sentence similarity modeling with convolutional neural networks. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 1576-1586. (2015)
25. Zhang, Y., Chen, Q., Yang, Z., Lin, H., Lu, Z.: BioWordVec, improving biomedical word embeddings with subword information and MeSH. Scientific data. 6(1), 52 (2019)
26. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781. (2013)
27. Aggarwal, C. C., Hinneburg, A., Keim, D. A.: On the surprising behavior of distance metrics in high dimensional space. In International conference on database theory, pp. 420-434. Springer, Berlin, Heidelberg (2001)

28. Pontes, E. L., Huet, S., Linhares, A. C., Torres-Moreno, J. M.: Predicting the Semantic Textual Similarity with Siamese CNN and LSTM. arXiv preprint arXiv:1810.10641. (2018)
29. Kingma, D. P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980. (2014)