Automatic Construction of UMLS Metathesaurus with Deep Learning

Hong Yung (Joey) Yip
Lister Hill National Center for Biomedical Communications,
U.S. National Library of Medicine, National Institute of Health, Bethesda, Maryland
Mentor: Dr. Olivier Bodenreider

Abstract. The Unified Medical Language System (UMLS) is a repository of biomedical vocabularies developed by the US National Library of Medicine to integrate a variety of ways the same concepts are expressed by different terminologies and provide cross-walk among them. However, the current approach of constructing and inserting new resources to the existing Metathesaurus relies heavily on lexical knowledge, semantic pre-processing, and manual audits by human editors. Given the recent successes of supervised machine learning approach in their applications to the medical and healthcare domains, this project explores the use of Deep Learning to identify synonymy and non-synonymy among English UMLS concepts at the atom level. We use the Siamese network with LSTM and CNN models to learn the similarities and dissimilarities between pairs of atoms from the active subset of 2019AA UMLS. We generate about 15 million synonym pairs and for non-synonyms, interesting pairs that are lexically similar but differ in semantics are generated using a heuristic approach with Jaccard index. 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. Using the base lexical features of the atoms yields an overall F1-score of 75.97%. Adding source synonymy to the base yields a higher precision and overall F-1 score of 86.54% and 87.63% respectively. Whereas, adding hierarchical context trades precision for higher recall of 90.38%. Adding source synonymy, hierarchical context, and the semantic group provides an overall increase in accuracy to 95.20%. However, adding source synonymy of hierarchical context does not yield any noticeable improvement. The Deep Learning approach provides relatively good performance in identifying synonymy and non-synonymy among atoms indicating a promising potential for emulating the current building process. Future works include evaluations with the manual rule-based normalization process of constructing the Metathesaurus and investigate the scalability, maintenance, and applicability aspects of these models.

Keywords: Unified Medical Language System, UMLS, Metathesaurus, Semantic Similarity, Deep Learning

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 and by different people [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[1]. Till date, it is increasingly being used in areas such as patient care coordination, clinical coding, information retrieval, and data mining. There are three UMLS Knowledge Sources: the Metathesaurus, the Semantic Network, and the SPECIALIST Lexicon and Lexical Tools.

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

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

The Metathesaurus is a multi-purpose vocabulary database organized by concept or meaning. It is constructed from the electronic versions of different 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 broad and consistent 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 NCBI taxonomy, *Systematized Nomenclature of Medicine - Clinical Terms* (SNOMED CT), Gene Ontology, the *Medical Subject Headings* (MeSH), and OMIM[2].

1.1 Construction of the UMLS Metathesaurus

The current approach in creating the Metathesaurus uses the lexical knowledge, semantic pre-processing, and UMLS human editors. The main idea is that synonymous terms originating from various 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 (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[1]. 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, and metadata information for semantic pre-processing. The UMLS human editors are involved to associate concepts and perform manual reviews. It is important to note that, when combining these source vocabularies, the Metathesaurus preserves the meanings, concept names, and relationships [1]. Nonetheless, these processes of constructing, combining, 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 especially at the current state of Metathesaurus comprising of over 3.85 million concepts. Given the recent advent of supervised machine learning approaches in their applications to the medical and healthcare domains [2], can they be trained to "fit" a new resource to the current "universe" of Metathesaurus?

1.2 Supervised Machine Learning

Supervised machine learning is a learning function that maps an input to an output based on examples of input-output pairs [3]. The Metathesaurus comprises of approximately 10 million English atoms, each with its CUI assignment, one can train a supervised classifier to predict which CUI should be assigned to an "unseen" or "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 approximately 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) two atoms that are lexically similar in nature but are not synonymous, e.g., "Lung disease and disorder" versus "Head disease and disorder" and vice versa, (2) atoms that are lexically dissimilar but are synonymous, e.g., "Addison's disease" versus "Primary adrenal deficiency".

Measuring the similarity 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, text summarization, and in our case, synonyms clustering. The STS task can be expressed as follows: given two sentences, a system returns a continuous score on a scale from 1 to 10 indicating the degree of similarity. STS is a challenging task due to the inherent complexity in language expressions, word ambiguities, and variable sentence lengths. Although there are existing models such as bag-of-words or TF-IDF models that incorporate a variety of similarity measures [5] for example string-based [6], term-based [7], most are syntactically and semantically constrained. Recent successes in sentence similarity have been obtained from combining corpus-based [8] and knowledge-based similarity, e.g. word embeddings [9] with supervised machine learning approach, e.g. Deep Neural Networks [10] and Convolutional Neural Networks (CNN) [11] to perform deep analysis of words and sentences to learn the semantics and structure necessary to predict the sentence similarity. Hence in this paper, we aim to explore the realm of Deep Learning 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 Deep Learning approach could emulate the current Metathesaurus building process

The rest of the report is organized as follows. Section 2 presents some related work in the area of STS. Section 3 describes our methodological approach and Section 4 shows the results and evaluations. Section 5 discusses the outcomes and limitations of this project and Section 6 concludes the project with future work.

2 Related Work

Prior work on STS task centered heavily on hand-engineered lexical features (e.g. word overlap and subwords) and linguistic resources (e.g. corpora). Lai et. al extracted word relations and features based on co-occurrences and similarities between image captions and applied regression functions to estimate similarity scores [12]. Zhao et. al leveraged on the syntactic relationship, distinctive content similitudes, length and string features [13]. Severyn et al. learned textual similarity by integrating relational syntactic and structural representations with Support Vector Regression [14]. Only in recent years, the use of Deep Neural Networks for

STS has gained much attention. In particular, the use of word embeddings for features representation that is trained on huge corpora has shown to improve the results over conventional lexical feature engineering approach [15]. In addition, the advent of various Deep Learning architectures such as Siamese, Recurrent Neural Network (RNN), and CNN further improve the prediction of sentence similarity and relatedness [10].

Contrary to the traditional neural network which takes in one input at a time, the Siamese network or model 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) [22]. It was originally used for signature verification [22] and has since been applied to various applications such as face verification [23], unsupervised acoustic modeling [24], and learning semantic entailment [10] as well as text similarity [25]. On the other hand, RNN excels at processing sequential information due to the presence of memory cell to store and "remember" data read over time [16]. Another variant of RNN is the Long Short-Term Memory (LSTM). It enhances the standard RNN to handle long-term dependencies with the introduction of "gates" (input, output and forget gates) to control the flow of and retain information better through time. LSTM is more accurate in handling long sequences, but 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 [10, 17, 18]. On the other hand, CNN has also performed well in NLP due to its ability to extract distinctive features at a higher granularity [20]. He et al. used Siamese CNN architecture to learn sentence embedding and predict sentence similarity with features from various convolution and pooling operations [21]. In this project, we use a combination of LSTM and CNN to achieve a more accurate similarity prediction. The next section describes our methodology with respect to the characteristics of the UMLS dataset.

3 Methodology

The scope of this project can be streamlined into four different components: (i) retrieving and parsing the UMLS dataset, (ii) generating features for learning, (iii) creating the Siamese Neural Networks, and (iv) evaluating different Siamese networks with different data enrichment strategies. As for the dataset, we use the active subset from the 2019AA UMLS and remove the derivative, duplicative, and spelling variants sources. Table 2 shows the sources removed and the final characteristics of the dataset.

Table 2: Sources Removed and Final Dataset Characteristics

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
Final UMLS Size	
Number of atoms	9,533,853
Number of CUIs	3,793,516

3.1 Feature Engineering

The primary goal is to learn the similarities between atoms within a CUI and dissimilarities between atoms from different CUIs. Prior to generating the positive and negative pairs, we preprocess the lexical features of the atoms similar to how [27] preprocess their dataset (remove all punctuation except hyphen, lowercase, and tokenize by space) as we use their pre-trained BioWordVec embedding in our downstream models.

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 in terms of time and space complexities to generate approximately 9.5 million atoms squared of negative pairs since it is one atom against all other atoms from non-related CUIs. In addition, the class imbalance between positives and negatives will induce learning bias in which the model will suffer from lower precision in detecting synonyms due to a preference towards non-synonyms. In spite of this, the intuition is that we want the Deep Learning model to learn interesting negative pairs that are lexically similar but differ in semantics. Hence, we adopt a heuristic approach to reduce the space of negative pairs where we compute Jaccard index to include 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.

$$Jaccard\ Index\ (A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$$

Table 4: Jaccard Computation on Pairs of Atom from Different CUIs

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

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 [24, 25] for Siamese networks.

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

3.2 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 these concepts with additional features that indicate different meanings. Hence, we compose the experiments with various data enrichment strategies (Figure 2) 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.

Base. The base experiment consists of just 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 the body part and another refers to the medical device. Since not all source vocabularies provide hierarchical relationships, we assign a semantic group to the best knowledge of the human editors to the source of these concepts. Whereas for sources that provide hierarchical relationships, we assign semantic group based on the second-level concept from the root node of the original atom as a proxy to semantic categorization.

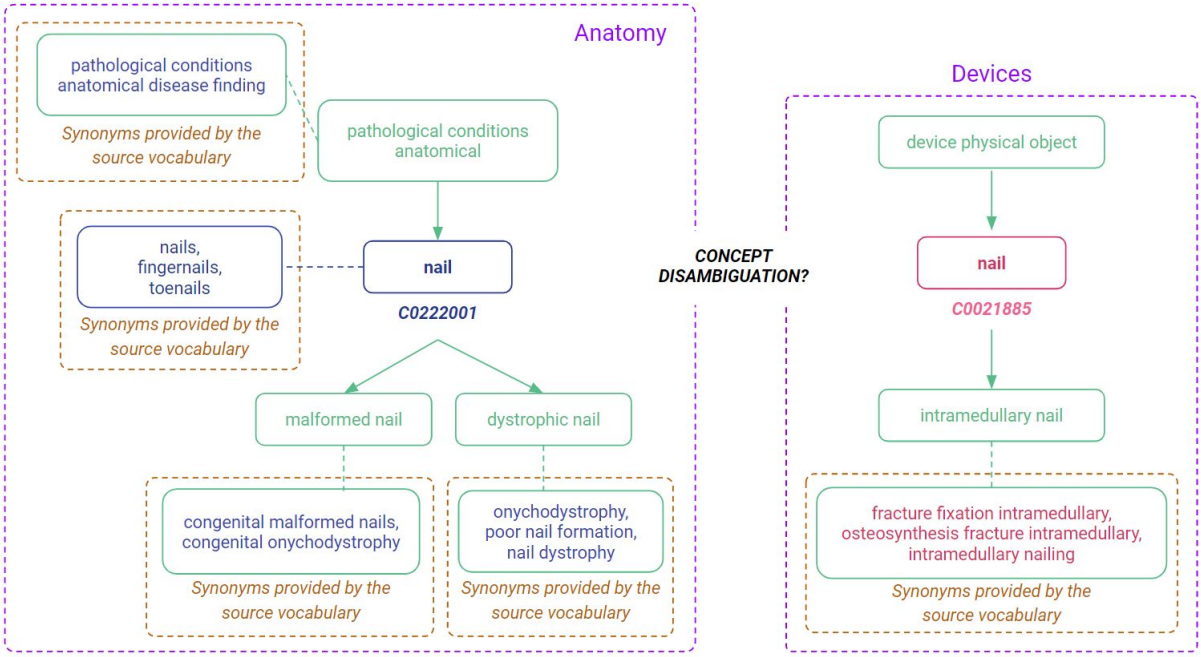

Figure 1: Concepts Disambiguation. The dotted brown boxes indicate source synonymy and the green boxes indicate hierarchical contexts.

Figure 2: Five Experimental Setup

3.3 Deep Learning Models

A total of two different Deep Learning Models are designed: Siamese LSTM and Siamese CNN-LSTM.

Siamese LSTM. This model adopts the Siamese structure from [10] (Figure 3) where 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 use 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 [27]. 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 (Figure 3), approximately 97% of atoms in the UMLS have a word length of lesser or equal to 30. We apply padding or truncation to restrict the word length of each atom to a maximum of length 30 to ensure a uniform dimension to speed up the training process. The embeddings of the pair of atoms are fed to the LSTM network which 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 similarity score between the two atoms ranging from 0 to 1 (likelihood probability) measured by Manhattan distance similarity function, a function that is well-suited for high dimensional space [28]. We apply this model to Experiment 1.

Siamese CNN-LSTM. We use this model for Experiment 2, 3, 4, and 5 to account for the additional features (source synonymy and/or hierarchical context) and semantic group information. This model adopts the Siamese structure from [29] (Figure 4), however, it differs from the first model in its hidden learning layers. For this model, instead of having only an embedding from the lexical features of the atoms, we concatenate two extra vectors learned from the embeddings that represent 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 since the word order is less prioritized prior to transforming them into context embedding. We apply CNN with 100 filters and a window size of 5 [29] with batch normalization (to reduce overfitting) to analyze together all the words and generates their representation as a unique structure and then apply a LSTM layer with 50 hidden learning units to learn these features. Similarly, the semantic group information is incorporated by transforming it using BioWordVec embedding and subsequently feeding it to a LSTM layer with 50 hidden units. The outputs of each LSTM layers (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 feedforward network with learning units of 128 and 50 respectively with Manhattan distance similarity function as the final output layer.

The parameters of both models are optimized using the Adam method [30] and each model is trained for 20 epochs and validated with 5-fold cross-validation the 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. Nonetheless, a set of experiments are conducted 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 hyperparameters of the network 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 6 summarizes the final set of parameters and hyperparameters that are used for the baseline experiment 1 and enriched experiment 2, 3, 4, and 5 respectively.

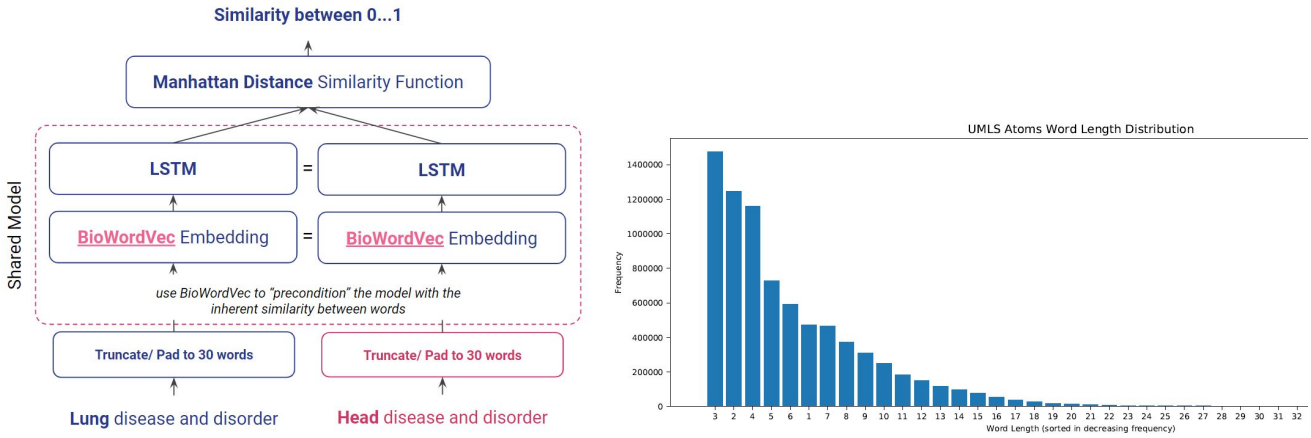

Figure 3: An overview of the Siamese LSTM Model. The weights of all the layers are shared between the left and right branch of the model.

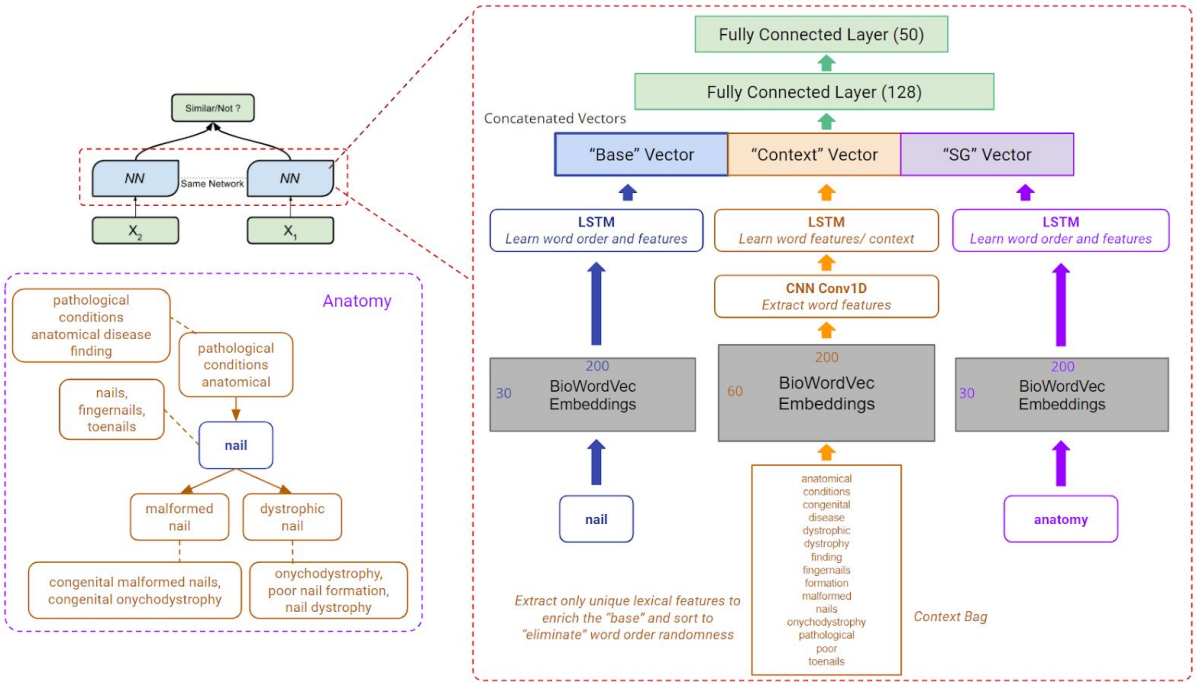

Figure 4: An overview of the Siamese CNN-LSTM Model. Similarly, the weights of all the layers are shared between the left and right branch of the model.

Table 3: 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	Keras 2.0 with Tensorflow backend
Word Vector Size	200	200
Maximum Input Length	30	30
Maximum Context Input Length	-	60
Embedding	Trainable	Trainable
LSTM Hidden Units	50	50
LSTM Activation	Tanh	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	Random Initialization
Optimizer	Adam	Adam
Learning Rate	0.001	0.001

Loss Function	Mean Squared Error (MSE)	Mean Squared Error (MSE)
Batch Size	128	128
Number of Training Epochs	20	20
Validation	5-fold cross-validation	5-fold cross-validation

The rationale behind the configurations of some hyper-parameters as follows:

- **Random initialization of weights and biases** ensure symmetry breaking for faster convergence.
- **Small batch size** of 128 per epoch step is applied to reap the benefits of both stochastic and batch update for computational faster and less memory overhead with a relatively good estimator.
- The number of training epochs is set to 20 to ensure uniform comparison among models.
- **Adam** optimizer is applied to reap the benefits of both RMSprop optimizer and momentum using stochastic gradient descent (SGD) to achieve faster learning [30].
- A standard **small learning rate** of 0.001 was selected to ensure numerical and weight stability.

4 Results and Evaluations

We evaluate the performance of the models in terms of validation accuracy, precision, recall, overall F1-Score, Matthew correlation coefficient, specificity, sensitivity, and false-positive rate. The rationale for such extensive measurements is due to the models learning at various proportions of negative to positive pairs and the accuracy metric alone may be "skewed" towards correctly identifying negative pairs and less of positive pairs. Table 4 shows the performance metrics achieved by the 6:1 ratio of negative to positive pairs in the process of classifying synonyms and non-synonyms. Table 5 shows some examples of true positives and true negatives correctly identified, false positives identified, and false negatives not identified.

Table 4: Performance of the 6:1 Ratio of Negative to Positive Pairs

Model/ Performance Metrics	**Siamese LSTM**	**Siamese CNN LSTM**			
	Experiment 1	**Experiment 2**	**Experiment 3**	**Experiment 4**	**Experiment 5**
	Base	**Base + Source Synonymy**	**Base + Hier. Context + Semantic Group**	**Base + Source Synonymy + Hier. Context + Semantic Group**	**Base + Source Synonymy + Hier. Context + Hier. Source Synonymy + Semantic Group**
Accuracy	0.9333	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
Matthew CC	0.7214	0.7441	0.7712	**0.8173**	0.8215
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

Table 5: Examples of True Positives and True Negatives Correctly Identified, False Positives Identified, and False Negatives Not Identified by 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

5 Discussion

Based on Table 4, we observe that using only the lexical features of atom yields an overall F1-score of 75.97%. Adding source synonymy to the base yields a higher precision and overall F-1 score of 86.54% and 87.63% respectively. Whereas, adding hierarchical context trades precision for higher recall of 90.38%. Adding source synonymy, hierarchical context, and the semantic group gives an overall boost to the accuracy of 95.20%. However, adding 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 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 5, we observe the performance of the trained 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 current architecture to learn from more examples.

6 Conclusion

In conclusion, this study demonstrates the feasibility of using Deep Learning to provide relatively good performance in identifying synonymy and non-synonymy among atoms indicating a promising potential for emulating the current Metathesaurus building process. The findings can be summarized as follows: (i) Adding source synonymy provides higher precision, but (ii) adding hierarchical context trades precision for higher recall. However, (iii) adding source synonymy, hierarchical context, and the semantic group gives an overall boost to accuracy, and (iv) adding source synonymy of hierarchical context does not yield any noticeable improvement. Nonetheless, this approach does not address the inter-concept and semantic type categorizations (other components in the Metathesaurus). Future work includes (a) evaluations with the manual rule-based normalization process of constructing the Metathesaurus since the current evaluations are done within the realm of Deep Learning, i.e. evaluating which features provide better performance, and not between the manual and automatic way of constructing the Metathesaurus, as well as (b) the scalability, maintenance, and applicability aspects of these models to complement the current lexical processing and the UMLS human editors.

Acknowledgment

I would like to thank Dr. Olivier Bodenreider for his guidance, time, and patience throughout the summer project. I would also like to extend my acknowledgment to Dr. Paul Fontelo for orchestrating the Summer Internship Training Program and Dr. Vinh Nguyen for her valuable insights on data engineering and optimization. Lastly, I would like to thank my colleagues, Rashmie Abeysinghe and Karan Luthria for their support on this project. The UMLS dataset used in this study can be retrieved with a UMLS license/ account at https://www.nlm.nih.gov/research/umls/licensedcontent/umlsknowledgesources.html.

References

1. Bodenreider, O. (2004). The unified medical language system (UMLS): integrating biomedical terminology. Nucleic acids research, 32(suppl_1), D267-D270.
2. Esteva, A., Robicquet, A., Ramsundar, B., Kuleshov, V., DePristo, M., Chou, K., ... & Dean, J. (2019). A guide to deep learning in healthcare. Nature medicine, 25(1), 24.
3. Russell, S. J., & Norvig, P. (2016). Artificial intelligence: a modern approach. Malaysia; Pearson Education Limited,.
4. Bengio, S., Dembczyński, K., Joachims, T., Kloft, M., & Varma, M. Extreme Classification.
5. Gomaa, W. H., & Fahmy, A. A. (2013). A survey of text similarity approaches. International Journal of Computer Applications, 68(13), 13-18.
6. Hall, P. A., & Dowling, G. R. (1980). Approximate string matching. ACM computing surveys (CSUR), 12(4), 381-402.

7. Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. Information processing & management, 24(5), 513-523.
8. Landauer, T. K., & Dumais, S. T. (1997). A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. Psychological review, 104(2), 211.
9. Mihalcea, R., Corley, C., & Strapparava, C. (2006, July). Corpus-based and knowledge-based measures of text semantic similarity. In Aaai (Vol. 6, No. 2006, pp. 775-780).
10. Mueller, J., & Thyagarajan, A. (2016, March). Siamese recurrent architectures for learning sentence similarity. In Thirtieth AAAI Conference on Artificial Intelligence.
11. He, H., Gimpel, K., & Lin, J. (2015, September). 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).
12. Lai, A., & Hockenmaier, J. (2014, August). Illinois-lh: A denotational and distributional approach to semantics. In Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014) (pp. 329-334).
13. Zhao, J., Zhu, T., & Lan, M. (2014, August). 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).
14. Severyn, A., Nicosia, M., & Moschitti, A. (2013, August). 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).
15. Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In Advances in neural information processing systems (pp. 3111-3119).
16. Rychalska, B., Pakulska, K., Chodorowska, K., Walczak, W., & Andruszkiewicz, P. (2016, June). 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).
17. Greff, K., Srivastava, R. K., Koutník, J., Steunebrink, B. R., & Schmidhuber, J. (2016). LSTM: A search space odyssey. IEEE transactions on neural networks and learning systems, 28(10), 2222-2232.
18. Tai, K. S., Socher, R., & Manning, C. D. (2015). Improved semantic representations from tree-structured long short-term memory networks. arXiv preprint arXiv:1503.00075.
19. Kim, Y. (2014). Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882.
20. Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., & Kuksa, P. (2011). Natural language processing (almost) from scratch. Journal of machine learning research, 12(Aug), 2493-2537.
21. He, H., Gimpel, K., & Lin, J. (2015, September). 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).
22. Bromley, J., Guyon, I., LeCun, Y., Säckinger, E., & Shah, R. (1994). Signature verification using a "siamese" time delay neural network. In Advances in neural information processing systems (pp. 737-744).
23. Chopra, S., Hadsell, R., & LeCun, Y. (2005, June). Learning a similarity metric discriminatively, with application to face verification. In CVPR (1) (pp. 539-546).
24. Synnaeve, G., & Dupoux, E. (2016). A temporal coherence loss function for learning unsupervised acoustic embeddings. Procedia Computer Science, 81, 95-100.
25. Neculoiu, P., Versteegh, M., & Rotaru, M. (2016, August). Learning text similarity with siamese recurrent networks. In Proceedings of the 1st Workshop on Representation Learning for NLP (pp. 148-157).
26. Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.
27. Zhang, Y., Chen, Q., Yang, Z., Lin, H., & Lu, Z. (2019). BioWordVec, improving biomedical word embeddings with subword information and MeSH. Scientific data, 6(1), 52.
28. Aggarwal, C. C., Hinneburg, A., & Keim, D. A. (2001, January). On the surprising behavior of distance metrics in high dimensional space. In International conference on database theory (pp. 420-434). Springer, Berlin, Heidelberg.
29. Pontes, E. L., Huet, S., Linhares, A. C., & Torres-Moreno, J. M. (2018). Predicting the Semantic Textual Similarity with Siamese CNN and LSTM. arXiv preprint arXiv:1810.10641.
30. Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.